Rexroth
Bosch Group

Rexroth IndraWorks 13VRS WinStudio 7.3

R911336881
Ausgabe 01

Referenz



Titel Rexroth IndraWorks 13VRS
WinStudio 7.3

Art der Dokumentation Referenz

Dokumentations-Type DOK-IWORKS-WINSTUD*V13-AP01-DE-P

Interner Ablagevermerk RS-745c358d9b032dba0a6846a501cacaba-1-de-DE-25

Zweck der Dokumentation Dieses "Benutzerhandbuch und Technische Referenz" soll Sie dabei unterstützen, mit Ihrer Software "Rexroth WinStudio"™ optimale Ergebnisse zu erzielen.

Das Dokument bietet technische Informationen und Schritt-für-Schritt-Anleitungen zum Anlegen von webfähigen HMI/SCADA-Programmen.

Änderungsverlauf

Ausgabe	Stand	Bemerkung
Ausgabe 01	10.2013	Erstausgabe 13 VRS

Schutzvermerk © Bosch Rexroth AG 2013

Alle Rechte bei Bosch Rexroth AG, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht, bei uns.

Verbindlichkeit Die angegebenen Daten dienen allein der Produktbeschreibung und sind nicht als zugesicherte Eigenschaften im Rechtssinne zu verstehen. Änderungen im Inhalt der Dokumentation und Liefermöglichkeiten der Produkte sind vorbehalten.

Herausgeber Bosch Rexroth AG

Bgm.-Dr.-Nebel-Str. 2 ■ D-97816 Lohr a. Main

Telefon +49 9352 18 0 ■ Fax +49 9352 18 8400

http://www.boschrexroth.com/

DC-IA/EAY2 TBe (vha)

Hinweis Diese Dokumentation ist auf chlorfrei gebleichtem Papier gedruckt.

Inhaltsverzeichnis

Seite

1 Zu dieser Dokumentation ... 19
1.1 Gültigkeit dieser Dokumentation ... 19
1.2 Gliederung der Dokumentation ... 19
1.3 Gebrauch der Sicherheitshinweise ... 20
1.3.1 Aufbau der Sicherheitshinweise ... 20
1.3.2 Erläuterung der Signalwörter und der Signalgrafik ... 21
1.3.3 Verwendete Symbole ... 21
1.4 In dieser Dokumentation verwendete Konventionen ... 22
1.5 Weiterführende Dokumentationen ... 23

2 Über diese Anwendung ... 25
2.1 Über diese Anwendung, Allgemeines ... 25
2.2 Produktübersicht ... 25
2.3 Produktmerkmale ... 26
2.4 Funktionsweise der Software ... 27
2.4.1 Interne Struktur und Datenfluss ... 27
2.4.2 Ausführen/Wechseln von Modulen ... 30
2.4.3 Ausführen/Wechseln des Hintergrund-Tasks ... 35

3 Installation ... 39
3.1 Installation, Allgemein ... 39
3.2 Systemanforderungen ... 39
3.3 Installation der Software ... 40
3.4 Starten der Engineering Software ... 44

4 Lizenzierung ... 47
4.1 Arten des Lizenzschutzes ... 47
4.2 Lizenzeinstellungen ... 47
4.3 Lizenzarten ... 48
4.4 Produktversionen ... 49
4.5 Aktivieren einer WinStudio-Lizenz ... 50
4.6 Ungültige Lizenzen ... 51

5 Entwicklungsumgebung ... 53
5.1 Die Entwicklungsumgebung, Allgemeines ... 53
5.2 WinStudio-Symbolleisten ... 53
5.2.1 WinStudio-Symbolleisten, Anpassen ... 53
5.2.2 Symbolleiste WinStudio-Ausführungsfunktionen ... 54
5.2.3 Symbolleiste WinStudio-Ausrichtungsfunktionen ... 55
5.2.4 Symbolleiste WinStudio-Tag-Eigenschaften ... 56
5.2.5 Symbolleiste WinStudio-Objekteigenschaften ... 56
5.2.6 Symbolleiste WinStudio-Tabellenfunktionen ... 57
5.2.7 Symbolleiste WinStudio-Debug-Funktionen ... 57

Inhaltsverzeichnis

Seite

5.2.8 Symbolleiste WinStudio-Dynamikfunktionen ... 58
5.3 Projektexplorer ... 59
5.3.1 Projektexplorer, Allgemeines ... 59
5.3.2 Ordner "WinStudio" ... 60
5.3.3 Ordner "Variablen" ... 60
5.3.4 Ordner "Tasks" ... 61
5.3.5 Ordner Kommunikation ... 63
5.4 Anwendung konfigurieren und testen ... 64
5.4.1 Anwendung verifizieren ... 64
5.4.2 Anwendung starten ... 65
5.4.3 Anwendung beenden ... 66
5.4.4 WinStudio-Zielplattform ... 66
5.4.5 Projekteinstellungen ... 67
Registerkarte Informationen ... 67
Registerkarte Optionen ... 68
Registerkarte Optionen, Allgemeines ... 68
Alarm-Historie in einer externen Datenbank speichern ... 69
Konfigurieren einer Standarddatenbank für alle Task-Historien ... 70
Datenbankkonfiguration ... 70
Registerkarte Viewer ... 75
Registerkarte Viewer, Allgemeines ... 75
Tastatur ... 80
Virtuelle Tastatur ... 80
Registerkarte Kommunikation ... 81
Registerkarte Voreinstellungen ... 83
Registerkarte Web ... 86
5.5 Werkzeuge ... 90
5.6 Bild-/Arbeitsblatt-Editor ... 92
5.7 Datenbankspion ... 93
5.8 Ausgabefenster LogWin ... 93
5.9 Querverweisliste ... 96
5.10 Statuszeile ... 96

6 WinStudio-Projekte ... 99
6.1 Anlegen eines neuen Projekts ... 99
6.1.1 Anlegen eines neuen Projekts, Allgemeines ... 99
6.1.2 Zielplattformen und Produkttypen ... 100
6.1.3 Zielsystem eines bestehenden Projekts ändern ... 101
6.2 E-Mail-Einstellungen konfigurieren ... 102
6.3 FTP-Einstellungen konfigurieren ... 104
6.4 Projektausführung unter Windows-Systemdiensten ... 105

7 Arbeiten mit Tags ... 115
7.1 Tags und die Projektdatenbank ... 115
7.1.1 Tags und die Projektdatenbank, Allgemeines ... 115

Inhaltsverzeichnis

Seite

7.1.2 Anwendungsvariablen ... 116
Anwendungsvariablen, Allgemeines ... 116
Erweitern des Datenblatts Anwendungsvariablen ... 118
7.1.3 Ordner Klassen ... 118
7.1.4 Verteilte Variablen ... 120
7.1.5 Interne Variablen ... 121
7.2 Erstellen von Tags ... 121
7.2.1 Namenssyntax für Tags ... 121
7.2.2 Auswahl des Tag-Typs ... 122
7.2.3 Auswahl des Tag-Datentyps ... 124
7.2.4 Verwenden von Array-Tags ... 125
7.2.5 Verwenden von Klassen ... 128
7.2.6 Verwendung von indirekten Tags (Zeiger) ... 131
7.2.7 Auswahl des Tag-Gültigkeitsbereichs ... 132
7.3 Erstellen von Datenbank-Tags ... 132
7.3.1 Erstellen von Tags im Datenblatt ... 132
7.3.2 Erstellen von Tags "On the Fly" ... 134
7.3.3 Bearbeiten von Tags ... 134
7.4 Erstellen von Klassen ... 135
7.5 Festlegen von Tag-Eigenschaften ... 136
7.5.1 Tag-Eigenschaften und -Parameter ... 136
7.5.2 Verwenden von Tag-Eigenschaften: Alarme ... 138
7.5.3 Verwenden von Tag-Eigenschaften: Historie ... 141
7.5.4 Liste der Tag-Eigenschaften ... 142
7.6 Verwenden von Tags in Ihrem Projekt ... 147
7.7 Löschen eines Tags aus der Projektdatenbank ... 148
7.8 Tags umbenennen und entfernen ... 148
7.8.1 Tags global ersetzen ... 148
7.8.2 Tags in einem Dokument oder Bildobjekt ersetzen ... 149
7.8.3 Nicht verwendete Tags aus der Projektdatenbank entfernen ... 150
7.8.4 Tag-Datenbank zurücksetzen ... 152
7.8.5 Verwenden vorhandener Tags ... 152
7.8.6 Ändern von Tag-Eigenschaften ... 153
7.9 Importieren einer externen Datenbank ... 154
7.9.1 Import-Assistent ... 154
7.9.2 Importieren aus ... 157
Importieren aus anderen Projektdatenbanken ... 157
Importieren aus OPC-Server-Datenbanken ... 159
Importieren aus CSV-Datenbanken ... 161
Importieren aus ODBC-Datenbanken ... 162
7.10 Integrieren von Tags ... 163
7.10.1 Integrieren von Tags, Allgemeines ... 163
7.10.2 Tags gemeinsam mit IndraLogic-Geräten verwenden ... 163

8 Bilder und Grafiken ... 169
8.1 Bilder und Grafiken, Allgemeines ... 169

Inhaltsverzeichnis

Seite

8.2 Ordner des WinStudios ... 170
8.2.1 Ordner WinStudio ... 170
Ordner WinStudio, Allgemeines ... 170
Dialog Bildeigenschaften ... 170
Hintergrundfarbe oder Hintergrundbild eines WinStudio-Bildes ... 174
8.2.2 Ordner Bildgruppen ... 175
8.2.3 Ordner Webseiten ... 176
8.3 Verwenden von Bildobjekten und Animationen ... 178
8.3.1 Bearbeiten von Bildobjekten ... 178
Bearbeiten von Bildobjekten, Allgemeines ... 178
Bearbeiten von Objekteigenschaften ... 178
Ersetzen von Projekt-Tags in einem Dokument oder Bildobjekt ... 179
Anordnen von Objekten ... 180
Verschieben von Objekten im Bild-Editor sperren ... 181
Rückgängig ... 181
Rastereinstellungen ... 181
8.3.2 Statische Objekte ... 182
Statische Objekte, Allgemeines ... 182
Objekt Linie ... 182
Objekt Offenes Polygon ... 183
Objekt Geschlossenes Polygon ... 184
Objekt Rechteck ... 185
Objekt Abgerundetes Rechteck ... 187
Objekt Ellipse ... 188
Hinzufügen von Grafiken zu einem WinStudio-Bild ... 189
8.3.3 Aktive Objekte ... 190
Aktive Objekte, Allgemeines ... 190
Objekt Text ... 190
Objekt Schaltfläche ... 191
Objekt Drucktaste ... 193
Objekt Kontrollkästchen ... 198
Objekt Radiobutton ... 201
Betriebsmodi für die Objekte Kontrollkästchen und Optionsfeld ... 204
Objekt Combobox ... 206
Objekt Listenfeld ... 210
Objekt Smart Message ... 214
8.3.4 WinStudio-Bibliotheken ... 219
Symbole ... 219
Symbole – Allgemeines ... 219
Speichern von eigenen Projektsymbolen ... 220
Active X-Control ... 229
.NET-Controls ... 236
Hinzufügen einer Verknüpfung zu einer externen Grafikdatei ... 242
8.3.5 Anwenden von Animationen auf Bildobjekte ... 243
Anwenden von Animationen auf Bildobjekte, Allgemeines ... 243
Befehls-Animation ... 243

Inhaltsverzeichnis

Seite

Hyperlink-Animation ... 247
Balkendiagramm-Animation ... 248
Animation Text-Daten-Verknüpfung ... 249
Farb-Animation ... 251
Positions-Animation ... 253
Animation Größenänderung ... 254
Animation Drehen ... 256
8.4 Formatierung von Bildobjekten ... 257
8.4.1 Formatierung von Bildobjekten, Allgemeines ... 257
8.4.2 "In den Vordergrund" und "In den Hintergrund" ... 258
8.4.3 "Nach Vorne" und "Nach Hinten", eine Ebene ... 259
8.4.4 "Objekte gruppieren" und "Gruppierung aufheben" ... 259
8.4.5 "Ausrichten", "Zentrieren" und "Verteilen" ... 260
8.4.6 "Drehen" und "Spiegeln" ... 262
"Drehen" ... 262
"Vertikal spiegeln" ... 262
"Horizontal spiegeln" ... 263
8.4.7 Position anpassen ... 263
8.4.8 "Gleiche Breite" / "Gleiche Höhe" für ausgewählte Objekte ... 263
8.4.9 Füllfarbe ... 264
8.4.10 Linienfarbe ... 265
8.4.11 Farbschnittstelle ... 265
8.4.12 Schriftarten ... 269

9 Alarme, Ereignisse und Trends ... 273
9.1 Alarme, Ereignisse und Trends, Allgemeines ... 273
9.2 Alarmarbeitsblatt ... 273
9.2.1 Alarmarbeitsblatt, Allgemeines ... 273
9.2.2 Titelbereich Alarmarbeitsblatt ... 275
Titelbereich Alarmarbeitsblatt, Allgemeines ... 275
E-Mail-Einstellungen für Alarmarbeitsblatt ... 276
Weitere Einstellungen für Alarmarbeitsblatt ... 279
9.2.3 Hauptteil Alarmarbeitsblatt ... 280
9.2.4 Speichern der Alarmhistorie in einer externen Datenbank ... 281
9.2.5 Format der Alarmhistorie ... 282
9.3 Ereignisse und Ereignisprotokolle ... 285
9.3.1 Ereignisse und Ereignisprotokolle, Allgemeines ... 285
9.3.2 Aktivierung der Ereignisprotokollierung ... 285
9.3.3 Speichern des Ereignisprotokolls in einer externen Datenbank ... 287
9.3.4 Format des Ereignisprotokolls ... 288
9.4 Alarm-/Ereignissteuerungsobjekt ... 290
9.5 Trendarbeitsblatt ... 301
9.5.1 Trendarbeitsblatt, Allgemeines ... 301
9.5.2 Konvertieren von Trendhistoriendateien von binär in Text ... 303
9.5.3 Konvertieren von Trendhistoriendateien von Text in binär ... 304
9.5.4 Stapelerstellung Historie ... 305

Inhaltsverzeichnis

Seite

9.5.5 Trenddatenbank einstellen ... 308
9.6 Trendanzeige ... 309
9.6.1 Trendanzeige, Allgemeines ... 309
9.6.2 Bedienelemente der Trendanzeige ... 310
9.6.3 Eigenschaften eines Trendobjekts ... 313
Eigenschaften eines Trendobjekt, Allgemeines ... 313
Punkte ... 314
Dialog Punkte, Allgemeines ... 314
Stiftstil ... 315
Optionen ... 317
SPC ... 319
Achsen ... 320
Werkzeugleiste ... 325
Datenquellen ... 326
Legende ... 330
Weitere Einstellungen ... 332
9.6.4 Verwenden einer Textdatei als Datenquelle ... 335
9.6.5 Verwenden einer Datenbank als Datenquelle ... 339
9.7 Tabellenobjekt ... 345
9.7.1 Tabellenobjekt, Allgemeines ... 345
9.7.2 Dialog Spalten ... 347
9.7.3 Dialog Daten ... 349
9.7.4 Dialog Erweiterte Einstellungen ... 351

10 Hintergrundtasks ... 357
10.1 Hintergrundtasks, Allgemeines ... 357
10.2 Alarmarbeitsblatt, Allgemeines ... 357
10.3 Trendarbeitsblatt, Allgemeines ... 358
10.4 Rezeptur-Arbeitsblatt ... 361
10.5 Report-Arbeitsblatt ... 363
10.6 ODBC-Arbeitsblatt ... 364
10.7 Mathematik-Arbeitsblatt ... 367
10.8 Skript-Arbeitsblatt ... 368
10.8.1 Skript-Arbeitsblatt, Allgemeines ... 368
10.8.2 Skript-Arbeitsblatt Startup ... 369
10.9 Scheduler-Arbeitsblatt ... 370
10.10 Datenbank-/ERP-Arbeitsblatt ... 371

11 Kommunikation mit anderen Geräten ... 379
11.1 Kommunikation mit anderen Geräten, Allgemeines ... 379
11.2 Konfiguration der direkten Kommunikation mit einem Remote-Gerät ... 379
11.2.1 Konfiguration der direkten Kommunikation mit einem Remote-Gerät, Allgemeines ... 379
11.2.2 Treiberblätter ... 387
11.3 Konfigurieren einer OPC-Client-Verbindung mit einem OPC-Server ... 390
11.4 Konfigurieren einer OPC-UA-Client-Verbindung mit einem OPC-UA-Server ... 395

Inhaltsverzeichnis

Seite

11.5 Konfigurieren einer TCP/IP-Client-Verbindung zu einem anderen Projekt ... 401
11.6 Konfigurieren einer DDE-Client-Verbindung mit einem DDE-Server ... 403

12 Projektsicherheit ... 407
12.1 Projektsicherheit, Allgemeines ... 407
12.2 Sicherheitsmodi ... 407
12.3 Sicherheitszugriffsebenen ... 408
12.4 Verwenden des Konfigurationsassistenten für das Sicherheitssystem ... 410
12.5 Konfigurieren der Servereinstellungen für Sicherheitsmodi ... 412
12.5.1 Konfigurieren der Servereinstellungen für Sicherheitsmodi, Allgemeines ... 412
12.5.2 Erweitern des LDAP-Schemas zum Speichern von Sicherheitsberechtigungen ... 417
12.6 Erstellen und Konfigurieren von Gruppen ... 423
12.7 Erstellen und Konfigurieren von Benutzern ... 428
12.8 Dialog Sicherheitssystem ... 430
12.9 Datensicherung der Sicherheitssystemkonfiguration ... 431
12.10 An- / Abmelden ... 432
12.11 Sperren / Entsperren eines Benutzers ... 434
12.12 Passwortschutz für Bilder, Symbole und Arbeitsblätter ... 434

13 Übersetzung von Anwendertexten ... 437
13.1 Übersetzung von Anwendertexten, Allgemeines ... 437
13.2 Hinzufügen einer Projektsprache ... 437
13.3 Übersetzen Ihrer Anwendungstexte in weitere Sprachen ... 438
13.4 Aktivierung einer Sprache zur Laufzeit ... 438
13.4.1 Aktivierung einer Sprache zur Laufzeit, Allgemein ... 438
13.4.2 Einstellen der Projektsprache bei Programmstart ... 439
13.4.3 Einstellen der Projektsprache während der Laufzeit ... 439
13.5 Übersetzung ausgewählter Bildobjekte deaktivieren ... 440

14 Test und Fehlerbeseitigung für Projekte ... 443
14.1 Ausgabe (LogWin) ... 443
14.2 Datenbankspion ... 445
14.3 Verwenden des LogWin-Moduls ... 446
14.4 Verwenden von Remote-Werkzeugen ... 447
14.4.1 Remote-Datenbankspion ... 447
14.4.2 Remote LogWin ... 448

15 Bereitstellung als Web-Anwendung ... 449
15.1 Einführung in Thin Clients ... 449
15.2 Erstellen eines einfachen Thin-Client-Programms ... 451
15.3 Zugrunde liegende Technologie ... 454
15.4 ISSymbol Control Layer ... 455
15.5 Beispiele für Client/Server-Architektur ... 458
15.6 Konfigurieren des Datenservers ... 463
15.7 Konfigurieren eines Webservers als Host Ihrer Projektseiten ... 467

Inhaltsverzeichnis

Seite

15.8 Installieren des Web Tunneling Gateway ... 470
15.9 Konfigurieren des Thin Client ... 473
15.10 Sicherheit für webbasierte Anwendungen implementieren ... 477
15.11 Port-Belegung ... 482
15.12 Übung: Betrachten Ihres Projekts im Web ... 483

16 Herunterladen auf ein Remote-Gerät ... 487
16.1 Konfigurieren des Zielsystems ... 487
16.2 Konfigurieren der Entwicklungsstation ... 489
16.3 Automatisches Ausführen eines Projekts ... 490

17 Datenbankschnittstelle ... 493
17.1 Datenbankschnittstelle – Allgemeines ... 493
17.2 Relationale SQL-Datenbanken ... 494
17.3 Verknüpfen der Datenbank über einen Remote-Datenbankanbieter ... 495
17.4 Studio-Datenbank-Gateway ... 496
17.5 Datenbankkonfiguration ... 502
17.6 Konfigurieren einer Standarddatenbank für alle Task-Historien ... 508
17.7 Fehlerbeseitigung Datenbank ... 509
17.8 Anhänge ... 513
17.8.1 Verwendung von ODBC-Datenbanken ... 513
17.8.2 Verwendung von Microsoft SQL Server ... 515
17.8.3 Verwendung von ORACLE-Datenbanken ... 517
17.8.4 Verwendung von Microsoft Access Datenbanken ... 520
17.8.5 Verwendung von SQL Server CE ... 523
Verwendung von SQL Server CE, Allgemeines ... 523
Installieren von SQL Server auf einem Windows-Embedded-Gerät ... 524
17.8.6 Verwendung von Sybase ... 525
17.8.7 Verwendung von Microsoft Excel ... 526
17.8.8 Verwendung von MySQL ... 530

18 Fehlerbeseitigung und häufig gestellte Fragen ... 533
18.1 Allgemeine Fehlerbeseitigung ... 533
18.2 Häufig gestellte Fragen ... 534

19 Anhang: WinStudio-interne Skriptsprache ... 545
19.1 Anhang: Interne Skriptsprache, Allgemeines ... 545
19.2 Liste der verfügbaren Funktionen ... 545
19.3 Aufbau der Funktionsbeschreibungen ... 556
19.4 Protokollmeldungsfunktionen ... 557
19.4.1 Protokollmeldungsfunktionen – Allgemeines ... 557
19.4.2 Trace ... 557
19.5 Arithmetische Funktionen ... 558
19.5.1 Arithmetische Funktionen – Allgemeines ... 558
19.5.2 Arithmetische Operatoren ... 559

Inhaltsverzeichnis

Seite

19.5.3 Abs ... 559
19.5.4 Div ... 560
19.5.5 Format ... 560
19.5.6 GetBit ... 564
19.5.7 Mod ... 564
19.5.8 Pow ... 565
19.5.9 ResetBit ... 565
19.5.10 Round ... 566
19.5.11 SetBit ... 567
19.5.12 Sqrt ... 568
19.5.13 Swap16 ... 568
19.5.14 Swap32 ... 568
19.5.15 Trunc ... 569
19.6 Statistische Funktionen ... 569
19.6.1 Statistische Funktionen – Allgemeines ... 569
19.6.2 Avg ... 570
19.6.3 Max ... 571
19.6.4 Min ... 571
19.6.5 Rand ... 572
19.7 Logarithmische Funktionen ... 573
19.7.1 Logarithmische Funktionen – Allgemeines ... 573
19.7.2 Exp ... 573
19.7.3 Log ... 573
19.7.4 Log10 ... 574
19.8 Logische Funktionen ... 574
19.8.1 Logische Funktionen – Allgemeines ... 574
19.8.2 Logische Operatoren ... 574
19.8.3 False ... 576
19.8.4 If ... 576
19.8.5 Toggle ... 577
19.8.6 True ... 577
19.9 String-Funktionen ... 578
19.9.1 Stringfunktionen – Allgemeines ... 578
19.9.2 Asc2Str ... 579
19.9.3 CharToValue ... 579
19.9.4 CharToValueW ... 580
19.9.5 ClassMembersToStrVector ... 581
19.9.6 NCopy ... 582
19.9.7 Num ... 583
19.9.8 Str ... 583
19.9.9 Str2Asc ... 583
19.9.10 StrCompare ... 584
19.9.11 StrCompareNoCase ... 585
19.9.12 StrFromInt ... 585
19.9.13 StrFromReal ... 586
19.9.14 StrFromTime ... 587

Inhaltsverzeichnis

Seite

19.9.15 StrGetElement ... 588
19.9.16 StrLeft ... 588
19.9.17 StrLen ... 589
19.9.18 StrLower ... 589
19.9.19 StrRChr ... 589
19.9.20 StrRight ... 590
19.9.21 StrSetElement ... 590
19.9.22 StrStr ... 591
19.9.23 StrStrPos ... 591
19.9.24 StrTrim ... 592
19.9.25 StrTrimAll ... 593
19.9.26 StrUpper ... 593
19.9.27 ValueToChar ... 594
19.9.28 ValueWToChar ... 594
19.10 Datums- und Zeitfunktionen ... 595
19.10.1 Datums- und Zeitfunktionen – Allgemeines ... 595
19.10.2 ClockGetDate ... 595
19.10.3 ClockGetDayOfWeek ... 596
19.10.4 ClockGetTime ... 597
19.10.5 DateTime2Clock ... 597
19.10.6 GetClock ... 598
19.10.7 Hour2Clock ... 598
19.10.8 SetSystemDate ... 599
19.10.9 SetSystemTime ... 599
19.11 Trigonometrische Funktionen ... 599
19.11.1 Trigonometrische Funktionen – Allgemeines ... 599
19.11.2 ACos ... 600
19.11.3 ASin ... 600
19.11.4 ATan ... 601
19.11.5 Cos ... 601
19.11.6 Cot ... 601
19.11.7 Pi ... 602
19.11.8 Sin ... 602
19.11.9 Tan ... 602
19.12 Bildfunktionen ... 603
19.12.1 Bildfunktionen – Allgemeines ... 603
19.12.2 Close ... 603
19.12.3 Open ... 604
19.12.4 OpenPrevious ... 606
19.12.5 ShowInplaceInput ... 607
19.12.6 ShowMessageBox ... 608
19.13 Sicherheitsfunktionen ... 610
19.13.1 Sicherheitsfunktionen – Allgemeines ... 610
19.13.2 BlockUser ... 610
19.13.3 CheckESign ... 611
19.13.4 CreateUser ... 611

Inhaltsverzeichnis

Seite

19.13.5 ExportSecuritySystem... 612
19.13.6 GetSecuritySystemStatus... 613
19.13.7 GetUserFullName... 615
19.13.8 GetUserNames... 615
19.13.9 GetUserPwdAging... 616
19.13.10 GetUserState... 617
19.13.11 ImportSecuritySystem... 617
19.13.12 RemoveUser... 619
19.13.13 SetPassword... 619
19.13.14 UnblockUser... 620
19.14 Funktionen zur Modulaktivität... 621
19.14.1 Funktionen zur Modulaktivität – Allgemeines... 621
19.14.2 AppActivate... 622
19.14.3 AppIsRunning... 624
19.14.4 AppPostMessage... 624
19.14.5 AppSendKeys... 625
19.14.6 CleanReadQueue... 626
19.14.7 CloseSplashWindow... 626
19.14.8 DisableMath... 627
19.14.9 EnableMath... 627
19.14.10 EndTask... 627
19.14.11 ExitWindows... 628
19.14.12 IsScreenOpen... 629
19.14.13 IsTaskRunning... 629
19.14.14 IsViewerInFocus... 630
19.14.15 KeyPad... 630
19.14.16 LogOff... 632
19.14.17 LogOn... 632
19.14.18 Math... 633
19.14.19 PostKey... 633
19.14.20 Recipe... 634
19.14.21 Report... 635
19.14.22 RunGlobalProcedureAsync... 636
19.14.23 RunGlobalProcedureAsyncGetStatus... 637
19.14.24 RunGlobalProcedureOnFalse... 639
19.14.25 RunGlobalProcedureOnServer... 640
19.14.26 RunGlobalProcedureOnTrigger... 641
19.14.27 RunGlobalProcedureOnTrue... 642
19.14.28 RunVBScript... 643
19.14.29 SecureViewerReload... 644
19.14.30 SendKeyObject... 644
19.14.31 SetAppPath... 646
19.14.32 SetViewerInFocus... 647
19.14.33 SetViewerPos... 647
19.14.34 ShutDown... 648
19.14.35 StartTask... 648

Inhaltsverzeichnis

Seite

19.14.36 ViewerPostMessage ... 649
19.14.37 Wait ... 650
19.14.38 WinExec ... 651
19.14.39 WinExecIsRunning ... 653
19.15 Dateifunktionen ... 653
19.15.1 Dateifunktionen – Allgemeines ... 653
19.15.2 DeleteOlderFiles ... 654
19.15.3 DirCreate ... 655
19.15.4 DirDelete ... 655
19.15.5 DirLength ... 656
19.15.6 DirRename ... 657
19.15.7 FileCopy ... 657
19.15.8 FileDelete ... 658
19.15.9 FileLength ... 659
19.15.10 FileRename ... 659
19.15.11 FileWrite ... 659
19.15.12 FindFile ... 661
19.15.13 FindPath ... 662
19.15.14 GetFileAttributes ... 662
19.15.15 GetFileTime ... 663
19.15.16 GetHstInfo ... 663
19.15.17 GetLine ... 664
19.15.18 HST2TXT ... 667
19.15.19 HST2TXTIsRunning ... 668
19.15.20 LookupContains ... 669
19.15.21 LookupGet ... 670
19.15.22 LookupLoad ... 670
19.15.23 PDFCreate ... 671
19.15.24 Print ... 672
19.15.25 RDFileN ... 673
19.15.26 WebGetFile ... 673
19.16 Grafische Funktionen ... 674
19.16.1 Grafische Funktionen – Allgemeines ... 674
19.16.2 AutoFormat ... 674
19.16.3 GetScrInfo ... 675
19.16.4 PrintSetup ... 676
19.16.5 PrintWindow ... 677
19.16.6 ResetDecimalPointsTable ... 678
19.16.7 RGBColor ... 679
19.16.8 RGBComponent ... 679
19.16.9 SaveScreenShot ... 680
19.16.10 SetDecimalPoints ... 681
19.16.11 SetDisplayUnit ... 682
19.16.12 SetTagDisplayUnit ... 683
19.17 Übersetzungsfunktionen ... 684
19.17.1 Übersetzungsfunktionen – Allgemeines ... 684

Inhaltsverzeichnis

Seite

19.17.2 Ext ... 684
19.17.3 SetLanguage ... 685
19.17.4 SetTranslationFile ... 685
19.18 Multimediafunktionen ... 686
19.18.1 Multimediafunktionen – Allgemeines ... 686
19.18.2 Play ... 686
19.19 Systeminformationsfunktionen ... 687
19.19.1 Systeminformationsfunktionen – Allgemeines ... 687
19.19.2 DbVersion ... 688
19.19.3 GetAppHorizontalResolution ... 689
19.19.4 GetAppPath ... 689
19.19.5 GetAppVerticalResolution ... 690
19.19.6 GetComputerIP ... 690
19.19.7 GetComputerName ... 690
19.19.8 GetCursorX ... 691
19.19.9 GetCursorY ... 691
19.19.10 GetDisplayHorizontalResolution ... 691
19.19.11 GetDisplayVerticalResolution ... 692
19.19.12 GetFreeMemoryCE ... 692
19.19.13 GetHardKeyModel ... 693
19.19.14 GetHardKeySN ... 693
19.19.15 GetIPAll ... 694
19.19.16 GetMemoryCE ... 694
19.19.17 GetNetMACID ... 695
19.19.18 GetOS ... 696
19.19.19 GetPrivateProfileString ... 696
19.19.20 GetProductPath ... 697
19.19.21 GetRegValue ... 697
19.19.22 GetRegValueType ... 698
19.19.23 GetServerHostName ... 699
19.19.24 GetTickCount ... 699
19.19.25 InfoAppAlrDir ... 700
19.19.26 InfoAppHSTDir ... 700
19.19.27 InfoDiskFree ... 700
19.19.28 InfoResources ... 701
19.19.29 IsActiveXReg ... 701
19.19.30 IsAppChangedOnServer ... 702
19.19.31 NoInputTime ... 703
19.19.32 ProductVersion ... 703
19.19.33 RegSaveCE ... 704
19.19.34 ReloadAppFromServer ... 704
19.19.35 SaveAlarmFile ... 705
19.19.36 SetAppAlarmPath ... 706
19.19.37 SetAppHSTPath ... 706
19.19.38 SetDateFormat ... 707
19.19.39 SetKeyboardLanguage ... 707

Inhaltsverzeichnis

Seite

19.19.40 SetRegValue ... 708
19.19.41 SetWebConfig ... 709
19.19.42 SNMPGet ... 711
19.19.43 SNMPSet ... 712
19.19.44 WritePrivateProfileString ... 714
19.20 Tag-Datenbank-Funktionen ... 714
19.20.1 Tag-Datenbank-Funktionen – Allgemeines ... 714
19.20.2 ExecuteAlarmAck ... 715
19.20.3 ForceTagChange ... 716
19.20.4 GetTagValue ... 716
19.20.5 SetTagValue ... 717
19.21 Schleifenfunktionen ... 718
19.21.1 Schleifenfunktionen – Allgemeines ... 718
19.21.2 For … Next ... 718
19.22 ODBC-Funktionen ... 719
19.22.1 ODBC-Funktionen – Allgemeines ... 719
19.22.2 ODBCBeginTrans ... 720
19.22.3 ODBCBindCol ... 721
19.22.4 ODBCCanAppend ... 722
19.22.5 ODBCCanTransact ... 722
19.22.6 ODBCCanUpdate ... 723
19.22.7 ODBCClose ... 723
19.22.8 ODBCCommitTrans ... 724
19.22.9 ODBCDelete ... 724
19.22.10 ODBCExecuteSQL ... 725
19.22.11 ODBCInsert ... 725
19.22.12 ODBCIsBOF ... 726
19.22.13 ODBCIsDeleted ... 727
19.22.14 ODBCIsEOF ... 727
19.22.15 ODBCIsFieldNULL ... 728
19.22.16 ODBCIsFieldNullable ... 728
19.22.17 ODBCMove ... 729
19.22.18 ODBCMoveFirst ... 729
19.22.19 ODBCMoveLast ... 730
19.22.20 ODBCMoveNext ... 730
19.22.21 ODBCMovePrev ... 731
19.22.22 ODBCOpen ... 732
19.22.23 ODBCQuery ... 733
19.22.24 ODBCRollBack ... 733
19.22.25 ODBCSetFieldNULL ... 734
19.22.26 ODBCSetFilter ... 734
19.22.27 ODBCSetSort ... 735
19.22.28 ODBCUnbindCol ... 736
19.22.29 ODBCUpdate ... 736
19.23 E-Mail-Funktionen ... 737
19.23.1 E-Mail-Funktionen – Allgemeines ... 737

Inhaltsverzeichnis

Seite

19.23.2 CnfEmail ... 737
19.23.3 GetStatusSendEmailExt ... 740
19.23.4 SendEmail ... 740
19.23.5 SendEmailExt ... 741
19.24 Einwahlfunktionen ... 743
19.24.1 Einwahlfunktionen – Allgemeines ... 743
19.24.2 DialError ... 744
19.24.3 DialGetClientIP ... 750
19.24.4 DialGetServerIP ... 751
19.24.5 DialStatus ... 753
19.24.6 DialUp ... 755
19.24.7 DialUpToCE ... 756
19.24.8 FindAllDevices ... 758
19.24.9 FindModem ... 758
19.24.10 HangUp ... 759
19.24.11 PhoneDialUp ... 759
19.24.12 PhoneDisableListen ... 760
19.24.13 PhoneEnableListen ... 761
19.24.14 PhoneHangUp ... 761
19.24.15 PhoneStatus ... 762
19.25 ActiveX- und .NET-Control-Funktionen ... 763
19.25.1 ActiveX- und .NET-Control-Funktionen – Allgemeines ... 763
19.25.2 XGet ... 763
19.25.3 XRun ... 764
19.25.4 XSet ... 765
19.26 Funktionen zur Ereignisprotokollierung ... 765
19.26.1 Funktionen zur Ereignisprotokollierung – Allgemeines ... 765
19.26.2 SendEvent ... 766
19.27 FTP-Funktionen ... 767
19.27.1 FTP-Funktionen – Allgemeines ... 767
19.27.2 CnfFTP ... 767
19.27.3 FTPGet ... 768
19.27.4 FTPPut ... 770
19.27.5 FTPStatus ... 771
19.28 Datenbank-/ERP-Funktionen ... 772
19.28.1 Datenbank-/ERP-Funktionen – Allgemeines ... 772
19.28.2 DBCursorClose ... 773
19.28.3 DBCursorColumnCount ... 774
19.28.4 DBCursorColumnInfo ... 775
19.28.5 DBCursorCurrentRow ... 776
19.28.6 DBCursorGetValue ... 776
19.28.7 DBCursorMoveTo ... 777
19.28.8 DBCursorNext ... 778
19.28.9 DBCursorOpen ... 779
19.28.10 DBCursorOpenSQL ... 781
19.28.11 DBCursorPrevious ... 783

Inhaltsverzeichnis

Seite

19.28.12 DBCursorRowCount ... 783
19.28.13 DBDelete ... 784
19.28.14 DBExecute ... 785
19.28.15 DBInsert ... 787
19.28.16 DBSelect ... 788
19.28.17 DBUpdate ... 790
19.28.18 SyncAlarm ... 791
19.28.19 SyncAlarmStatus ... 792
19.28.20 SyncEvent ... 793
19.28.21 SyncEventStatus ... 793
19.28.22 SyncTrend ... 794
19.28.23 SyncTrendStatus ... 795

20 Anhang: VBScript-Sprache ... 797
20.1 Übersicht VBScript ... 797
20.2 VBScript-Schnittstellen in der Software ... 797
20.2.1 VBScript-Schnittstellen in der Software, Allgemeines ... 797
20.2.2 Globale Prozeduren ... 799
20.2.3 Grafikmodul ... 800
Grafikskript ... 800
Bildskript ... 803
Befehlsanimation ... 804
ActiveX-Ereignisse ... 805
20.2.4 Hintergrundtask ... 808
20.3 Sprachverweis ... 808
20.3.1 Sprachverweis – Überblick ... 808
20.3.2 Operatoren ... 808
20.3.3 Konstanten ... 809
20.3.4 Objekte und Kollektionen ... 813
20.3.5 Eigenschaften ... 813
20.3.6 Anweisungen ... 814
20.3.7 Methoden ... 817
20.3.8 Funktionen ... 817
20.3.9 Schlüsselwörter ... 818
20.3.10 Fehler ... 818
20.3.11 Liste der System-Tags ... 821
20.4 Tipps & Tricks ... 823
20.4.1 Tipps & Tricks – Überblick ... 823
20.4.2 VBScript Editor IntelliSense ... 823
20.4.3 VBScript im Vergleich mit VBA ... 825
20.4.4 Bildereignisse ... 827
20.4.5 Funktionen MsgBox und InputBox ... 827
20.4.6 VBScript-Prozeduren ... 828
20.4.7 Erstellen von Konstanten ... 829
20.4.8 Deklarieren von Variablen ... 830
20.4.9 Geltungsbereich und Lebensdauer von Variablen ... 830

Inhaltsverzeichnis

Seite

20.4.10 Boolesche Tags und Boolesche Variablen ... 831
20.4.11 Schreiben von reellen Werten in Integer-Tags ... 832
20.4.12 Priorität von VBScript-Operatoren ... 832
20.4.13 Logischer Operator NOT ... 833
20.4.14 Verwenden von bedingten Anweisungen ... 834
20.4.15 Schleifen mit Code ... 836
20.4.16 Unterstützung für ActiveX-Steuerelemente ... 839
20.4.17 Unterstützung durch Windows Embedded ... 839

21 Anhang: Treiber-Fehlercodes ... 841

22 Service und Support ... 843

Index ... 845

1 Zu dieser Dokumentation

1.1 Gültigkeit dieser Dokumentation

Dieses "Benutzerhandbuch und Technische Referenz" soll Sie dabei unterstützen, mit Ihrer Software "Rexroth WinStudio"™ optimale Ergebnisse zu erzielen. Dieses Dokument bietet technische Informationen und Schritt-für-Schritt-Anleitungen zum Anlegen von webfähigen HMI/SCADA-Programmen.

Zielgruppe

Dieses "Benutzerhandbuch und Technische Referenz" ist ein umfassendes Dokument mit nützlichen Informationen sowohl für neue als auch für erfahrene Benutzer von "WinStudio".

- **Neue Benutzer**: Dieser Leitfaden verwendet einen praxisorientierten Schritt-für-Schritt-Ansatz für den Projektentwicklungsprozess. Lesen Sie hierzu die Einführung über die Merkmale des Produkts und der Entwicklungsumgebung.
- **Erfahrene Benutzer**: Das Dokument beinhaltet Anleitungen für Spezialisten sowie Tipps und Informationen zur Fehlersuche.

Erforderliche Qualifikationen: Person, die aufgrund ihrer fachlichen Ausbildung, Kenntnisse und Erfahrungen sowie Kenntnis der einschlägigen Normen und Bestimmungen die ihr übertragenen Arbeiten beurteilen und mögliche Gefahren erkennen kann.

1.2 Gliederung der Dokumentation

Die Informationen in diesem Dokument sind in folgende Abschnitte gegliedert:

- Kap. 2 "Über diese Anwendung" auf Seite 25: Enthält eine Produktübersicht, eine Auflistung wesentlicher Produktmerkmale sowie Erläuterungen zur Funktionsweise der Software
- Kap. 3 "Installation" auf Seite 39: enthält eine Anleitung für das Installieren von"Rexroth WinStudio", Engineering, EmbeddedView und CE-View
- Kap. 4 "Lizenzierung" auf Seite 47: beschreibt die Arten des Lizenzschutzes, Lizenzeinstellungen, das Aktivieren einer WinStudio-Lizenz und das WinStudio-Verhalten bei ungültiger bzw. nicht ausreichender Lizenz
- Kap. 5 "Entwicklungsumgebung" auf Seite 53: beschreibt die "WinStudio"-Benutzeroberfläche (bzw. "Entwicklungsumgebung") und erläutert einige grundlegende Fertigkeiten und Techniken, die Sie vor dem Anlegen eines neuen Projekts kennen müssen
- Kap. 6 "WinStudio-Projekte" auf Seite 99: Schritt-für-Schritt-Anleitungen zum Erstellen und Konfigurieren eines neuen Projekts
- Kap. 7 "Arbeiten mit Tags" auf Seite 115: erläutert Grundkonzepte zu Projektdatenbank, Tag-Typen (Felder, Klassen und Zeiger), Tag-Werten und Parametern. Neben der konzeptionellen Betrachtung werden das Erstellen und Bearbeiten von Tags in Ihrem Projekt beschrieben
- Kap. 8 "Bilder und Grafiken" auf Seite 169: erläutert, wie die Entwicklungswerkzeuge von "WinStudio" zum Erstellen eigener Bilder und Grafiken eingesetzt werden
- Kap. 9 "Alarme, Ereignisse und Trends" auf Seite 273: beschreibt, wie die Tasks Alarm und Trend zum Protokollieren historischer Daten verwendet werden und die Objekte Alarm/Ereignis und Trend zur Anzeige historischer Daten in einem Projektbild zu nutzen sind

- Kap. 10 "Hintergrundtasks" auf Seite 357: erläutert, wie die verschiedenen "Tasks" von "WinStudio" für Projekte angelegt und konfiguriert werden
- Kap. 11 "Kommunikation mit anderen Geräten" auf Seite 379: beschreibt die Konfiguration von Projekten zum Lesen und Schreiben von Prozessdaten und die Anleitung zur Konfiguration von Treibern und der OPC-, TCP/IP- und DDE-Kommunikation
- Kap. 12 "Projektsicherheit" auf Seite 407: erläutert das Einrichten und Verwalten eines Sicherheitssystems für Projekte
- Kap. 13 "Übersetzung von Anwendertexten" auf Seite 437: erläutert das Übersetzen von Anwendertexten mit Unterstützung des WinStudio-Übersetzungswerkzeugs
- Kap. 14 "Test und Fehlerbeseitigung für Projekte" auf Seite 443: erläutert den Test und die Fehlerbeseitigung bei Projekten mit Hilfe von Werkzeugen wie dem Datenbankspion und den Ausgabefenstern
- Kap. 15 "Bereitstellung als Web-Anwendung" auf Seite 449: erklärt das Konfigurieren und Ausführen eines Projektes als Web-Anwendung
- Kap. 16 "Herunterladen auf ein Remote-Gerät" auf Seite 487: Erläutert das Übertragen, das Überwachen und die Fehlerbeseitigung in Projekten über Remote-Zugriff
- Kap. 17 "Datenbankschnittstelle" auf Seite 493: erläutert das Anbinden von zu "WinStudio" kompatiblen Datenbanken
- **Skriptsprachen**
  - Kap. 19 "Anhang: WinStudio-interne Skriptsprache" auf Seite 545: beschreibt alle Funktionen der internen Skriptsprache von Winstudio. Jede Funktionsbeschreibung enthält die vollständige Syntax, mögliche Ausgabewerte und Anwendungsbeispiele
  - Kap. 20 "Anhang: VBScript-Sprache" auf Seite 797: "WinStudio" implementiert Visual Basic Scripting Edition 5.5 oder höher

    Gegeben wird ein Überblick über die Verwendung von VBScript-Kommandos im "Rexroth WinStudio". Ferner sind zum schnellen Nachschlagen die am häufigsten verwendeten Merkmale der Sprache zusammengestellt. Eine vollständige Beschreibung der Sprache sowie der Schnittstellen und Funktionen erhalten Sie unter Microsoft Developer Network
- Kap. 18 "Fehlerbeseitigung und häufig gestellte Fragen" auf Seite 533: Anleitung zum Überprüfen von Projekten; beschreibt einige häufig vorkommende Fehler bei der Entwicklung und erläutert die Vorgehensweise zur Kontaktaufnahme mit einem Kundendienstmitarbeiter

1.3 Gebrauch der Sicherheitshinweise

1.3.1 Aufbau der Sicherheitshinweise

Die Sicherheitshinweise sind wie folgt aufgebaut:

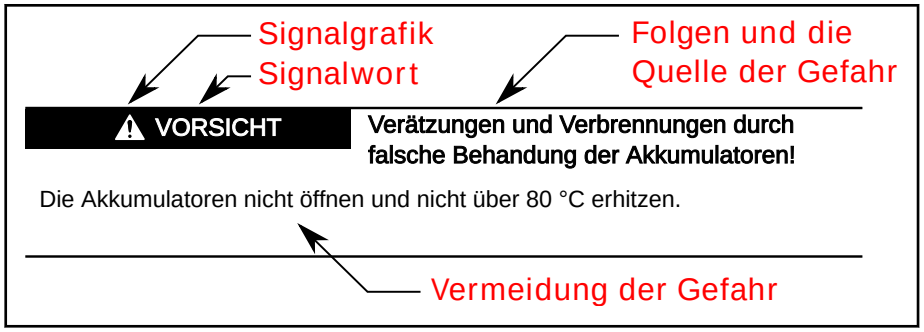


Abb.1-1: Aufbau der Sicherheitshinweise

1.3.2 Erläuterung der Signalwörter und der Signalgrafik

Die Sicherheitshinweise in der vorliegenden Dokumentation beinhalten bestimmte Signalwörter (Gefahr, Warnung, Vorsicht, Hinweis) und gegebenenfalls eine Signalgrafik (nach ANSI Z535.6-2006).

Das Signalwort soll die Aufmerksamkeit auf den Sicherheitshinweis lenken und bezeichnet die Schwere der Gefährdung.

Die Signalgrafik (Warndreieck mit Ausrufezeichen), welche den Signalwörtern Gefahr, Warnung und Vorsicht vorangestellt wird, weist auf Gefährdungen für Personen hin.

⚠ GEFAHR

Bei Nichtbeachtung dieses Sicherheitshinweises **werden** Tod oder schwere Körperverletzung eintreten.

⚠ WARNUNG

Bei Nichtbeachtung dieses Sicherheitshinweises **können** Tod oder schwere Körperverletzung eintreten.

⚠ VORSICHT

Bei Nichtbeachtung dieses Sicherheitshinweises können mittelschwere oder leichte Körperverletzung eintreten.

HINWEIS

Bei Nichtbeachtung dieses Sicherheitshinweises können Sachschäden eintreten.

1.3.3 Verwendete Symbole

Fingerzeige werden wie folgt dargestellt:

☞ Dies ist ein Hinweis.

Tipps werden wie folgt dargestellt:



Dies ist ein Tipp.

1.4 In dieser Dokumentation verwendete Konventionen

In dieser Dokumentation wird eine einheitliche Formatierung und Terminologie verwendet, um allen Anwendern das Verständnis zu erleichtern.

Textkonventionen

In dieser Dokumentation wird eine spezielle Textformatierung verwendet, damit Sie bestimmte Begriffe schnell erkennen können:

- Titel, Bezeichnungen, neue Begriffe und Meldungen werden in "nichtproportionaler" Schrift wiedergegeben (z.B. "Objekteigenschaften").
- Dateinamen, Bildschirmtext und einzugebender Text werden in **fettem** Schriftbild wiedergegeben (z.B. **D:\Setup.exe**).
- Schaltflächen, Menüoptionen und Tastaturtasten werden in **fettem** Schriftbild wiedergegeben (z.B. Menü **Datei**).

Maus- und Auswahlkonventionen

In dieser Dokumentation werden die folgenden Maus- und Auswahlkonventionen verwendet:

Beschrieben wird eine Maus, die für rechtshändigen Einsatz konfiguriert ist, so dass die linke Maustaste die Primärtaste und die rechte Maustaste die Sekundärtaste ist.

- Sowohl **Anklicken** als auch **Auswählen** bedeuten, mit der linken Maustaste einmal auf das Element zu klicken. Generell wird auf Schaltflächen "geklickt" und aus Menüs "ausgewählt".
- **Doppelklick** bedeutet, mit der linken Maustaste schnell zweimal hintereinander auf ein Element zu klicken.
- **Rechts-Klick** bedeutet, mit der rechten Maustaste auf ein Element zu klicken.
- **Auswählen** bedeutet auch, dass das Zeigegerät zum Markieren oder Festlegen eines Elements auf dem Rechnerbildschirm verwendet werden soll.

  Das Auswählen eines Elements mit Hilfe eines Touchscreens ist im Allgemeinen gleichbedeutend mit dem Auswählen mit der Maus, allerdings wird dabei ein Objekt oder Bereich des Bildschirms mit dem Finger berührt (ausgewählt).

  Um Elemente mit der Tastatur auszuwählen, wird üblicherweise die Tab-Taste zum Wechseln zwischen den Optionen verwendet, die Eingabetaste zum Öffnen von Menüs und die <Alt>-Taste zusammen mit einer Buchstabentaste zur Auswahl eines Objekts, das einen unterstrichenen Buchstaben hat.
- **Ziehen** bedeutet, die entsprechende Maustaste gedrückt zu halten und die Maus zu bewegen, bevor die Taste losgelassen wird. Im Allgemeinen bewegt sich dann ein Umriss des jeweiligen Elements zusammen mit dem Mauszeiger.

Windows-Konventionen

In dieser Dokumentation werden die folgenden Windows-Konventionen verwendet:

- **Dialoge** sind Fenster, mit denen Einstellungen konfiguriert und Informationen eingegeben werden können.
- **Textfelder** sind Bereiche zur Texteingabe in Dialogen.
- **Optionsfelder** sind weiße Kreise, in denen beim Anklicken der Schaltfläche ein schwarzer Punkt erscheint bzw. verschwindet.

  Ein Punkt zeigt an, dass die jeweilige Option bzw. Funktion "aktiviert" (ausgewählt) ist. Ein fehlender Punkt zeigt an, dass die jeweilige Option bzw. Funktion "deaktiviert" (nicht ausgewählt) ist.

- **Kontrollkästchen** sind weiße Quadrate, in denen beim Anklicken mit dem Cursor ein Häkchen erscheint bzw. verschwindet.

  Ein Häkchen zeigt an, dass die jeweilige Option bzw. Funktion "aktiviert" (ausgewählt) ist. Ein fehlendes Häkchen zeigt an, dass die jeweilige Option bzw. Funktion "deaktiviert" (nicht ausgewählt) ist.
- **Schaltflächen** sind Symbole in Kästchen, die "gedrückt" erscheinen, wenn sie angeklickt werden.
- **Listen** sind Felder (weiße Kästchen) in Fenstern oder Dialogen, die zwei oder mehr auswählbare Optionen enthalten.
- **Comboboxen** haben einen Pfeil, der beim Anklicken eine ansonsten versteckte Liste teilweise oder vollständig anzeigt.
- **Andockbare Fenster** sind Fenster, die an einen Rand der Benutzeroberfläche gezogen und so an diesen Rand angedockt werden können.

1.5 Weiterführende Dokumentationen

Ergänzend zu dieser Technischen Referenz können bei Bedarf die folgenden Handbücher zu Rate gezogen werden:

- "Handbuch Erste Schritte": Grundlegende Informationen über "Rexroth WinStudio"; dies umfasst eine systematische Lerneinheit, mit der Sie innerhalb kürzester Zeit ein Einzelprojekt entwickeln und sich mit dem Produkt vertraut machen können.
- "Benutzerhandbücher Treiber": Erläutern das Konfigurieren einzelner Direktkommunikationstreiber gemäß ihrer spezifischen Protokollmerkmale. Für jeden Treiber wird ein eigens erstelltes Benutzerhandbuch zur Verfügung gestellt.

☞ Alle Handbücher befinden sich im Ordner "Documentation" auf der Installations-CD von "WinStudio".

"WinStudio" installiert die Treiber-Benutzerhandbücher "BR_WS.pdf" im Ordner **\Drv** bzw. **\Drv\de-DE** im Programmverzeichnis.

Auf technische Informationen kann auch aus dem Menü **Hilfe** zugegriffen werden:

IndraWorks-Online-Hilfe ▸ IndraWorks-Komponenten ▸ HMI-Werkzeuge ▸ Bosch-Rexroth-WinStudio-Treiber

---

2 Über diese Anwendung

2.1 Über diese Anwendung, Allgemeines

"Rexroth WinStudio" (oder einfach "WinStudio" oder Studio) ist ein leistungsfähiges, vollständig integriertes Softwareprogramm für Entwurf und Erstellung von Projekten für "HMI" (Mensch-Maschine-Schnittstelle) oder "SCADA" (Supervisory Control and Data Acquisition) für:

- Datenerfassung
- Lokale Überwachungsstationen
- Remote-Überwachungsstationen
- Datenverdichtung auf verteilten Prozessen
- Datenkommunikation mit Unternehmenssystemen

2.2 Produktübersicht

"Rexroth-WinStudio"-Projekte werden auf Visualisierungsgeräten (z.B. IndraControlV) ausgeführt, die über programmierbare Steuerungen, Remote-E/A-Geräte und sonstige Datenerfassungsgeräte in Echtzeit mit Maschinen oder Prozessoren verbunden sind.

Diese Projekte bestehen aus animierten Bildern, konfigurierbaren SPS (Speicherprogrammierbare Steuerung) -Treibern und anderen steuerbaren E/A-Geräten, einer Datenbank mit Projekt-Tags und optionalen Modulen, wie z.B. Alarmüberwachungen, Logik, Trendgrafiken, Rezepturen, Scheduler und einem Sicherheitssystem. "WinStudio"-Projekte kommunizieren mit industriellen E/A-Systemen und anderen Windows-Anwendungen in der Laufzeitumgebung gemäß der folgenden Protokolle:

- ODBC ("Open Database Connectivity")
- DDE ("Dynamic Data Exchange")
- NetDDE ("Network""Dynamic Data Exchange")
- OPC ("Open Connectivity")
- TCP/IP ("Transmission Control Protocol/Internet Protocol")

Nach der Erstellung eines Projekts kann es in der Entwicklungsumgebung ausgeführt oder (mit Hilfe einer TCP/IP-Verbindung) auf das Zielgerät herunter geladen und mit Hilfe von "Rexroth WinStudio"-Laufzeit-Software ausgeführt werden.

Die Prozesse der Runtime-Umgebung kommunizieren mit den angeschlossenen Geräten und verarbeiten bzw. visualisieren die projektierten Daten.

Das Produkt besteht aus zwei Teilen:

- Entwicklungsumgebung: Die WinStudio-Entwicklungsumgebung ist Bestandteil von IndraWorks Engineering und auf Desktop-Computern, Laptops oder Industrie-PCs lauffähig, auf denen ein unterstütztes Microsoft-Windows-Betriebssystem für Desktop oder Server installiert ist.
- Laufzeitsystemsoftware: Diese ist auf Visualisierungsgeräten (z.B. IndraControlV) lauffähig, auf denen ein derzeit unterstütztes Microsoft-Windows-Betriebssystem für Desktop oder Windows Embedded installiert ist.



Der Laufzeit-Client für Windows CE-Betriebssysteme (CEView) ist meist auf dem HMI vorinstalliert.

Bei Bedarf können Sie die CE-View-Version aktualisieren, indem sie die aktuelle Version vom Entwicklungsrechner auf das HMI herunterladen.

2.3 Produktmerkmale

Das Produkt "Rexroth WinStudio" verfügt über folgende Merkmale:

- In IndraWorks Engineering integrierte

  Entwicklungsumgebung auf Seite 53 mit Editoren, Symbolleisten, Dialogen und Menüs:

  – anpassbare Symbolleisten
  – Tasks, Objekte und Steuerungen, die im Projektexplorer dargestellt Projektexplorer auf Seite 59 sind
- Vielseitig konfigurierbar Objekte und Animationen (Möglichkeit zum Verändern von Objekt-Eigenschaften, Ausführen von Befehlen oder Schreiben von Werten in Tags zur Erzeugung von Anzeigebildschirmen zur Laufzeit der Anwendung):

  – Konfigurierbare Objekte wie z.B. Schaltflächen, Rechtecke, Ellipsen, Polygone, Linien und Text
  – Objektanimationen auf Seite 243, wie z.B. Balkendiagramme, Farben, Größenänderungen, Positionsänderung, Ausblenden/Einblenden, Drehen, Kommandoausführung, Hyperlink und Ein-/Ausgabe von Text
  – Online und historische Anzeigen von Alarmlisten auf Seite 273
  – Online und historische Trends auf Seite 301
  – Die Formatierung von Bildobjekten auf Seite 257 wird durch die Symbolleisten "Ausrichtungsfunktionen" und "Objekteigenschaften" unterstützt. Es stehen Werkzeuge zum Anordnen, Verschieben und Ändern der Größe von ausgewählten Bildschirm-Objekten sowie zum Ändern der Stile und Schriftarten der Objekte zur Verfügung.
  – Hintergrundfarbe oder Hintergrundbild eines WinStudio-Bildes auf Seite 174 können bearbeitet bzw. erstellt werden.
  – Importieren von Grafiken
  – Container für ActiveX-Objekte auf Seite 229 und .Net-Controls
- Remote-Verwaltung und -Konfiguration der Laufzeit-Umgebung
- Einhaltung der Microsoft-DNA-Architektur, mit voller Unterstützung von OPC und XML
- Aktivierung einer Web-Schnittstelle, über die Bilder in ein internetfähiges Format konvertiert und von einem Client (Web-Browser) angezeigt werden können. Der Online-Datenaustausch erfolgt über das TCP/IP-Protokoll
- Symbolbibliothek auf Seite 219 mit mehr als 100 verfügbaren Objekten, wie Tastern, Messinstrumenten, Schiebereglern, Schaltern, Textanzeigen und numerischen Anzeigen, LED-Anzeigen, Leitungen, Pumpen, Symbolen, Fahrzeugen, Ventilen, Rahmen, Motoren, Messgeräten und Steuerungen

- Werkzeuge zur Fehlersuche:
  - Fenster Database Spy auf Seite 93 zum Überwachen/Setzen von Tag-Werten und Ausführen von Funktionen
  - LogWin-Modul auf Seite 93 zum Aufzeichnen von OPC-, DDE- und TCP/IP-Protokollen, zum Aktivieren von Modulen, zum Nachverfolgen von Tags usw.
  - Querverweissuche zum Auffinden von Tags innerhalb einer WinStudio-Anwendung
  - System und Netzwerk-Diagnose zur Laufzeit
- Leistungsfähige und flexible Tag-Datenbank (Boolean-, Integer-, Real- und String-Tags), Array-Tags, Klassen und indirekte Tags (Pointer)
- Offene Architektur mit API, Austausch von Tag-Werten mit externer Software
- Übersetzungseditor auf Seite 437 zum Übersetzen eines Projekts in verschiedene Sprachen und Umschalten zwischen den Sprachen während der Laufzeit des Projekts
- Client- und Server-Kommunikation auf Basis TCP/IP auf Seite 401 zum Austausch von Tag-Werten und Konfigurieren redundanter Systeme
- Treiber für Direktkommunikation auf Seite 379 zur Anbindung an Rexroth Steuerungen (BR_WS)
- OPC-Server und OPC-Client mit integriertem OPC-Browser auf Seite 390
- Laufzeit-Sicherheit auf Seite 407 mit Passwort-Schutz für Bilder und Objekte (256 Ebenen)
- Logische Ausdrücke und eine Skriptsprache auf Seite 545 mit mehr als 200 Funktionen
- Werkzeuge zur Erstellung von Rezepturen auf Seite 361 und Berichte auf Seite 363 (in den Formaten ASCII, UNICODE und RTF)
- Scheduler auf Seite 370 für Ereignisse auf Grundlage von Datums-, Zeit- oder Datenbedingung (Auflösung 100 ms)
- Import- Exportfunktion zur einfachen Übernahme von Bildern und Arbeitsblättern in andere Projekte.
- DFÜ-Funktionen auf Seite 545 zum Aufbauen, Überwachen und Trennen einer DFÜ-Verbindung mit dem RAS-Server von Remote-Stationen
- Funktionen zum Versenden von E-Mails auf Seite 545 aus "WinStudio" (oder CEView)

2.4 Funktionsweise der Software

2.4.1 Interne Struktur und Datenfluss

Die Laufzeitumgebung von "WinStudio" wird auf einem Visualisierungsgerät (Windows 2K/XP/Vista/7/CE) ausgeführt und besteht aus den folgenden Modulen bzw. "Threads" (Programmelemente, die voneinander unabhängig ausgeführt werden können):

- **Hintergrund-Task** (überwachender Task): Führt andere interne Tasks ("WinStudio"-Arbeitsblätter) aus. Zum Beispiel führt der "Hintergrund"-Task in den Arbeitsblättern Berechnung auf Seite 367 und Scheduler auf Seite 370 konfigurierte Skripte aus und verwaltet in den Arbeitsblättern Alarm auf Seite 273, Rezeptur auf Seite 361, Bericht auf Seite 363 und Trend auf Seite 301 konfigurierte Parameter.

- **Datenbankspion** (Werkzeug für die Fehleranalyse):
  - Führt Funktionen und/oder Ausdrücke zu Testzwecken aus
  - Liest Daten (z.B. Tag-Werte) aus der "Tag"-Datenbank
  - Schreibt Daten (z.B. Tag-Werte) in die "Tag"-Datenbank
- **DDE-Client**: verwaltet die DDE-Kommunikation mit einem DDE-Server (lokal oder fern), gemäß den in den Arbeitsblättern DDE-Client auf Seite 403 konfigurierten Parametern.
- **DDE-Server**: verwaltet die DDE-Kommunikation mit einem DDE-Client (lokal oder fern).
- **LogWin** (Werkzeug für die Fehleranalyse): zeichnet Meldungen von anderen Modulen/ Tasks auf.
- **Treiber-Laufzeitprogramm**: verwaltet die in den Treiber-Arbeitsblättern auf Seite 379 konfigurierten Lese-/Schreibbefehle
- **OPC-Client (OPC DA und OPC UA)**: Verwaltet die OPC-Kommunikation mit einem OPC-Server (lokal oder fern), gemäß den in den Arbeitsblättern OPC-Client auf Seite 390 konfigurierten Parametern.
- **OPC DA-Server**: Verwaltet die OPC DA-Kommunikation mit einem OPC DA-Client (lokal oder fern).
- **ODBC-Laufzeitprogramm**: Verwaltet die ODBC-Datenkommunikation mit einer relationalen SQL-Datenbank, gemäß den in den ODBC-Arbeitsblättern auf Seite 364 konfigurierten Parametern.
- **TCP/IP-Client**: Verwaltet die TCP/IP-Kommunikation mit einem fernen TCP/IP-Server-Modul (aus "WinStudio"), gemäß den in den Arbeitsblättern TCP-Client auf Seite 401 konfigurierten Parametern.
- **TCP/IP-Server**: Verwaltet die TCP/IP-Kommunikations-Meldungen mit einem fernen TCP/IP-Client-Modul (aus "WinStudio").
- **Viewer**: Führt alle für das Bild konfigurierten Skripte (On Open, On While, On Close, Command, Hyperlink usw.) aus und aktualisiert die Bildobjekte.

Keines der vorgenannten Laufzeitmodule tauscht direkt mit einem anderen Modul oder Task Daten aus. Die Laufzeitmodule schicken statt dessen Daten zur "Tag"-Datenbank oder erhalten von dort Daten. Diese ist sozusagen das „Herz“ von "WinStudio".

Die "Tag"-Datenbank verwaltet den Datenfluss zwischen Modulen. Außerdem speichert die "Tag"-Datenbank alle Tag-Werte und den Status aller mit jedem Tag verknüpften Eigenschaften (z.B. Alarmbedingungen, Zeitstempel, Qualität usw.).

☞ Tags sind Variablen (z.B. Kommunikationspunkte zu Feldgeräten, Berechnungsergebnisse, Alarmpunkte usw.), die in Bildern und Arbeitsblättern verwendet werden.

Detaillierte Informationen zu Tags, Tag-Werten und Tag-Eigenschaften unter Arbeiten mit Tags auf Seite Arbeiten mit Tags.

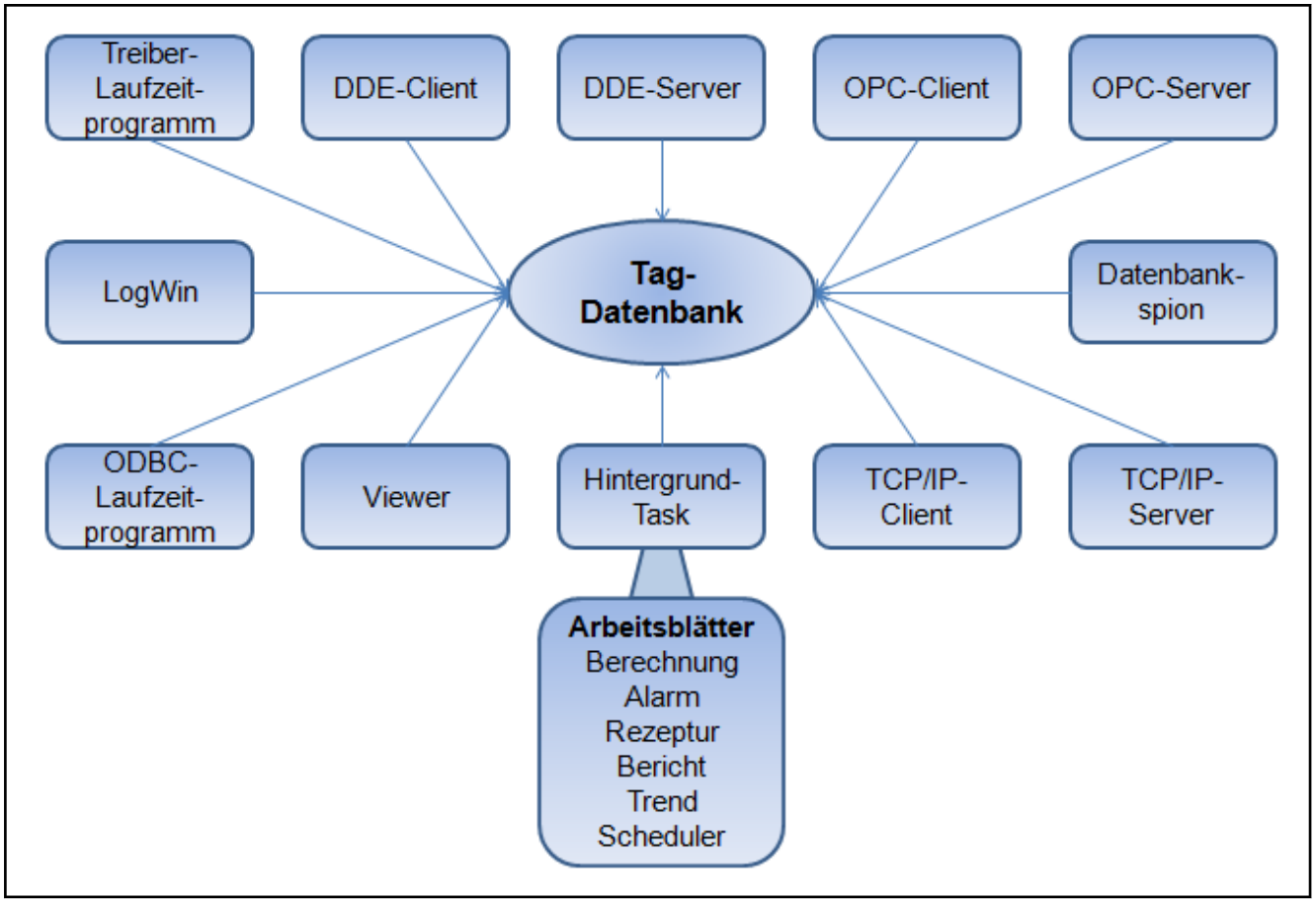


Abb.2-1: Datenfluss

Jedes "WinStudio"-Modul enthält eine virtuelle Tabelle der Tags, die für dieses Modul aktuell relevant sind. Mit Hilfe dieser Tabelle ermittelt die "Tag"-Datenbank, welche Informationen in jedem Modul aktualisiert werden müssen. Zum Beispiel enthält der "Viewer" eine virtuelle Tabelle, die alle Tags für alle offenen WinStudio-Bilder aufführt. Wenn sich ein Tag-Wert verändert, sendet die "Tag"-Datenbank eine Meldung zum "Viewer", der dann den Wert in allen Objekten aktualisiert.

Wenn beispielsweise ein Treiber einen neuen Wert aus der SPS ausliest, aktualisiert der Treiber den Wert in der "Tag"-Datenbank.

Wenn dann diese neue Information im WinStudio-Bild angezeigt werden muss, sendet die "Tag"-Datenbank den neuen Tag-Wert an den "Viewer"-Task und der "Viewer" aktualisiert das Bild.

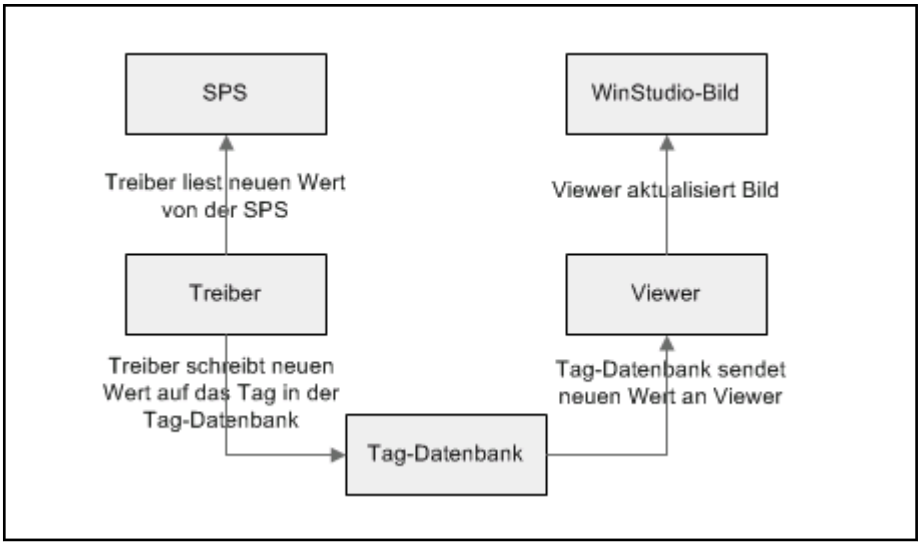


Abb.2-2: Beispiel für Datenfluss

Es ist zu beachten, dass der Treiber neue Tag-Werte nicht direkt an den "Viewer" schickt. Zudem gibt es kein Polling zwischen Tasks – die "Tag"-Datenbank erhält die aktualisierte Information und schickt sie unmittelbar an alle Laufzeit-Tasks weiter, die diese Information benötigen.

☞ Das "Laufzeitprogramm"-Modul aktualisiert ein Objekt nur, wenn sich (mindestens) ein Tag-Wert des Objekts ändert.

Wenn eine Objektanimation (z.B. Textdaten-Link auf Seite 249) mit einer Funktion konfiguriert wird, die keinen Tag benötigt (z.B. NoInputTime auf Seite 703), dann aktualisiert der "Viewer" das Objekt nicht.

Die Architektur von "WinStudio" verbessert die Leistung des internen Datenflusses erheblich und macht es Ihnen leicht, neue interne Tasks hinzuzufügen. Obwohl jeder Task unabhängig funktioniert, kann er über die "Tag"-Datenbank auf Informationen von jedem beliebigen anderen Task zugreifen.

2.4.2 Ausführen/Wechseln von Modulen

Bei "WinStudio" handelt es sich um ein SCADA-System aus Modulen, die gleichzeitig ausgeführt werden müssen. Jedes Laufzeitmodul ("Viewer", "Treiber" usw.) bildet - basierend auf dem Multitasking-Konzept - einen "Thread", und das Betriebssystem wechselt automatisch von einem Thread zum anderen. Es wird oft fälschlicherweise angenommen, dass beim Ausführen eines SPS-Programms ein SCADA-Programm ausgeführt wird. Ein SPS-Programm enthält eine einfache Schleife:

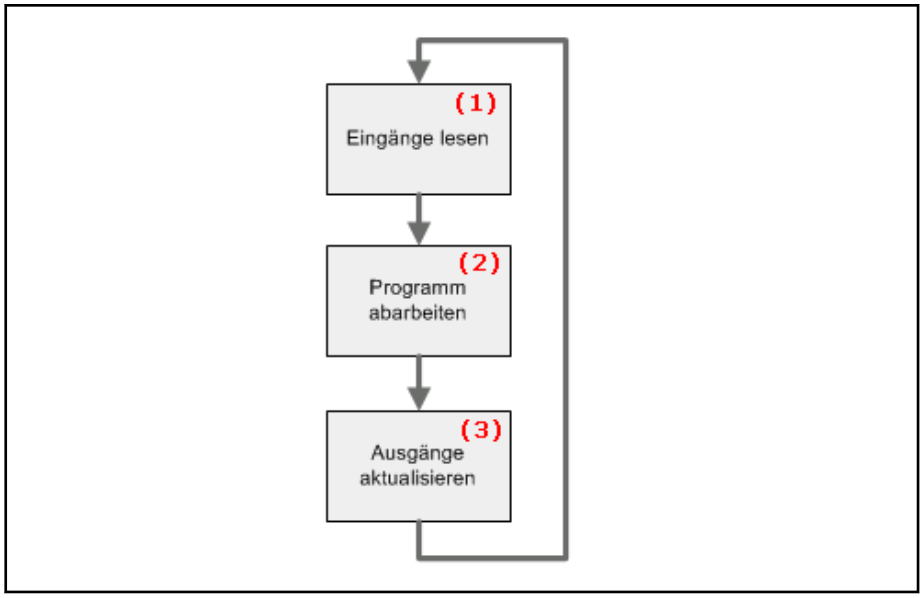


Abb.2-3: SPS-Programm-Schleife

In einem SCADA-System gibt es dagegen mehrere gleichzeitig ablaufende Module, von denen die meisten Daten lesen und schreiben können.

Da ein SCADA-System während der Task-Ausführung kontinuierlich Daten (Tag-Werte) verändert, trifft die vorstehende Grafik darauf "nicht" zu.

"WinStudio" wird nur in einem einzigen Prozess ausgeführt: "Studio Manager.exe". Wenn Sie ein Laufzeitprojekt ausführen, startet der Prozess "Studio Manager.exe" die "Tag"-Datenbank und alle für das Projekt konfigurierten Laufzeitmodule. Sie können angeben, welche Module (z.B. "Viewer" und "Treiber") während der Laufzeit starten.

Jeder Prozess in Windows führt eine Liste der "aktiven" Threads für das Betriebssystem. Tatsächlich aktiviert und deaktiviert jeder Prozess jeden Thread während der Laufzeit entsprechend dem Algorithmus des jeweiligen Prozesses. Zudem geben Sie beim Starten eines Threads einen Prioritätswert an. Das Betriebssystem scannt ständig alle aktuell aktiven Threads und führt die Threads gemäß ihrem jeweiligen Prioritätswert aus - wobei Threads mit höherer Priorität zuerst ausgeführt werden. Wenn Threads mit höheren Prioritätswerten aktiv sind, werden Threads mit niedrigeren Prioritätswerten überhaupt nicht ausgeführt. Wenn es mehrere Threads mit gleichem Prioritätswert gibt und es keine anderen Threads mit höheren Prioritätswerten gibt, wechselt das Betriebssystem fortlaufend zwischen diesen Threads mit gleicher Priorität.

Alle "WinStudio"-Threads werden auf die Priorität "7" gesetzt, was "THREAD_PRIORITY_NORMAL" ist. (Die meisten Programme enthalten diesen Prioritätswert.)

Threads für Echtzeitprogramme (z.B. SoftPLCs und Gerätetreiber) wird ein höherer Prioritätswert ("THREAD_PRIORITY_HIGHEST") zugewiesen; diese Programme müssen aber einen Mechanismus enthalten, der sie für eine gewisse Zeitdauer inaktiv werden lässt, ansonsten würden Threads mit normaler Priorität nie ausgeführt.

"WinStudio" verwendet die Bibliothek "UNICOMM.DLL" für serielle Treiber.

Diese Bibliothek legt einen Thread mit Priorität "THREAD_PRIORITY_HIGHEST" an, der so lange „schläft“ (inaktiv bleibt), bis Daten auf dem seriellen Kanal ankommen.

Wenn "WinStudio" neue Daten auf dem seriellen Kanal erkennt, wird der Thread "THREAD_PRIORITY_HIGHEST" „aufgeweckt“ (aktiviert) und überträgt die Daten vom Betriebssystempuffer in den Thread-Puffer, wo sie vom Treiber gelesen werden können. Dieser Thread ist der einzige Thread mit höchster Priorität, den "WinStudio" erzeugt.

Würde man Threads erlauben, jederzeit aktiv zu bleiben, betrüge die Auslastung des Hauptprozessors jederzeit 100%, was aus Gründen der Rechnerleistung vermieden werden muss. Jedes Programm stellt einen Mechanismus zur Verfügung, der eine ständige Aktivierung der Threads vermeidet.

"WinStudio" verwendet die folgenden Parameter, die verhindern, dass Threads fortlaufend aktiv bleiben:

- **TimeSlice** (vom Betriebssystem): Lässt das Betriebssystem automatisch zwischen aktiven Threads desselben Prioritätswerts hin- und herwechseln. Standardmäßig führt das Betriebssystem jeden aktiven Thread etwa 20 ms lang aus und wechselt dann zum jeweils nächsten aktiven Thread. Das bedeutet, dass im Fall von mehreren aktiven Threads mit demselben Prioritätswert kein aktiver Thread länger als 20 ms ausgeführt wird.
- **TimeSlice** (von "WinStudio"): Legt fest, wie lange jeder "WinStudio"-Thread fortlaufend aktiv bleiben kann.

  Dieser Parameter wird zusätzlich zum TimeSlice-Parameter des Betriebssystems eingesetzt. Ein TimeSlice-Wert wird für jeden "WinStudio"-Thread (außer dem Hintergrund-Task) konfiguriert unter Angabe, wie lange jeder Thread fortlaufend aktiv bleiben kann. Solange ein Thread aktiv ist, kann das Betriebssystem zu diesem Thread wechseln.
- **Period** (von "WinStudio"): Legt die maximale Zeitdauer fest, während der jeder "WinStudio"-Thread (außer dem Hintergrund-Task) inaktiv bleiben kann.

HINWEIS

Wir empfehlen "dringend", diese Standardwerte nicht zu verändern, wenn es nicht unbedingt erforderlich ist. Falsches Konfigurieren dieser Parameter kann zu Fehlfunktion des gesamten Systems führen (z.B. die Auslastung des Hauptprozessors steigt auf 100 % an) und/oder zu schlechter Leistung einiger Tasks.

Wenn Sie die Standardwerte für die Parameter abändern müssen, notieren Sie sich die Werte vor Ihren Änderungen, sodass Sie bei Auftreten einer Fehlfunktion zu den Originaleinstellungen zurückkehren können.

Um die Standardwerte für die "WinStudio"-Parameter TimeSlice und Period zu verändern, öffnen Sie die Datei "Program Settings.INI" im Verzeichnis "<Installationsverzeichnis von IndraWorks>\WinStudio\bin" mit einem Texteditor.

Die folgende Liste zeigt alle in dieser .ini-Datei enthaltenen Parameter und ihre zugehörigen Standardwerte (in Millisekunden).

Program Settings.INI

```
[Period]
DBSpy=1000
UniDDEClient=200
UniDDE=200
Driver=20
LogWin=100
UniODBCRT=100
```

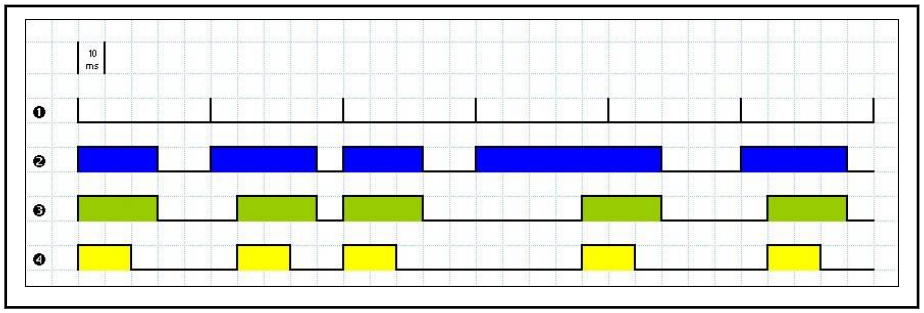
```
OPCClient=20
OPCServer=20
TCPClient=100
TCPServer=100
Viewer=50

[TimeSlice]
UniDDEClient=100
Driver=10
OPCClient=10
OPCServer=10
TCPClient=200
TCPServer=200
Viewer=200
```

Unter Umständen finden Sie nicht alle aufgelisteten Parameter in der Datei "Program Settings.INI". Die nicht enthaltenen Parameter verwendet "WinStudio" trotzdem und weist diesen Parametern Standardwerte zu.

- Um die Standardwerte eines angezeigten Parameters zu ändern, ersetzen Sie den vorhandenen Wert durch einen neuen und speichern Sie die Datei.
- Wenn Sie den Wert eines Parameters ändern möchten, der in der Datei nicht vorhanden ist, fügen Sie den Parameternamen und den gewünschten Wert wie in Program Settings.INI auf Seite 32 dargestellt ein.

Die folgende Abbildung veranschaulicht, wie "WinStudio" einen generischen Thread (wie den "Viewer") ausführt.



10ms Zeitraster des Diagramms

Abb.2-4: Ausführen eines generischen Threads

Hierbei gilt:

- Signal **1** ist die Zeitspanne "Period" (in diesem Beispiel auf 50 ms gesetzt).
- Signal **2** zeigt, wann der Thread für das Betriebssystem aktiv ist.
- Signal **3** ist die Zeitspanne "TimeSlice" (in diesem Beispiel auf 30 ms gesetzt).
- Signal **4** zeigt die Ausführung des Threads selbst.

In diesem Beispiel erzeugt "WinStudio" alle 50 ms eine Meldung Period (Signal **1**). Wenn "WinStudio" diese Meldung erzeugt, wird deren Thread aktiv und bleibt aktiv, bis die festgelegte Zeitspanne TimeSlice (von "WinStudio") abläuft. Dann bleibt der Thread inaktiv, bis "WinStudio" die nächste Meldung Period erzeugt (Signal **1**).

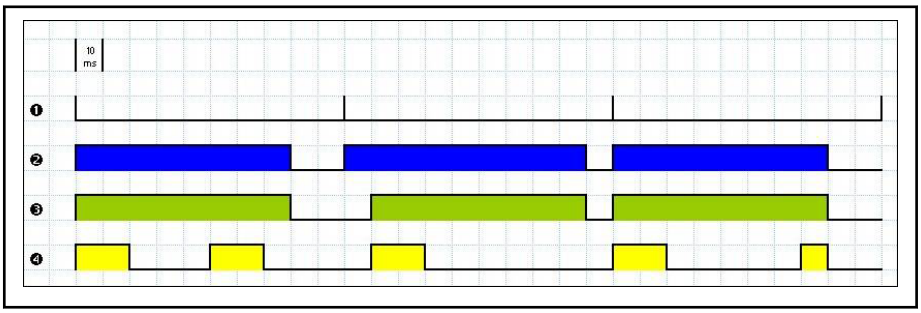
Während der Thread aktiv ist, ist das Betriebssystem für dessen Ausführung verantwortlich. Dass ein Thread aktiv ist, bedeutet jedoch noch nicht, dass das Betriebssystem ihn auch sofort ausführt. Dies ist zum Beispiel der Fall, wenn das Betriebssystem gerade andere Threads ausführt.

Wenn das Betriebssystem den Thread ausführt, beginnt der Timer "TimeSlice" zu laufen und der Thread wird 20 ms lang (TimeSlice vom Betriebssys-

tem) ausgeführt. Nach der Zeitspanne von 20 ms wechselt das Betriebssystem automatisch zum nächsten aktiven Thread (z.B. dem Treiber) usw.

Im obigen Beispiel wurde die Zeit TimeSlice auf 30 ms gesetzt, was bedeutet, dass das Betriebssystem den Thread nicht häufiger als einmal in jeder TimeSlice von "WinStudio" ausführen sollte. Wenn Sie jedoch höhere Werte für die "WinStudio"-Zeitspanne TimeSlice angeben, ist es wahrscheinlich, dass das Betriebssystem den gleichen Thread mehr als einmal in der gleichen TimeSlice-Zeitspanne ausführt.

Im nächsten Beispiel wurden die Werte Period und TimeSlice wie folgt geändert, die Standard-TimeSlice-Zeitdauer des Betriebssystems (20 ms) jedoch nicht.



10ms Zeitraster des Diagramms
Abb.2-5: Setzen eines höheren TimeSlice-Werts

Hierbei gilt:

- Signal **1** ist die Zeitspanne Period (auf 100 ms gesetzt).
- Signal **2** zeigt, wann der Thread für das Betriebssystem aktiv ist.
- Signal **3** ist die "WinStudio"-Zeitspanne TimeSlice (auf 80 ms gesetzt).
- Signal **4** zeigt die Ausführung des Threads selbst.

Bitte beachten Sie, dass der Thread in derselben TimeSlice-Zeitspanne öfter als einmal ausgeführt werden kann. Wenn die "WinStudio"-Zeitspanne TimeSlice abläuft, unterbricht das Betriebssystem die Ausführung des Threads; obwohl die "WinStudio"-Parameter Period und TimeSlice auf 100 ms bzw. 80 ms gesetzt sind, führt das Betriebssystem diesen Thread nicht kontinuierlich länger als 20 ms aus, da die Zeitspanne TimeSlice des Betriebssystems auf 20 ms gesetzt ist.

Wenn das Betriebssystem den Thread "Viewer" nicht ausführt, kann der Hauptprozessor jeden beliebigen anderen Thread ausführen oder inaktiv bleiben (wenn es keine anderen aktiven auszuführenden Threads gibt). Zur Erinnerung: Die "WinStudio"-Parameter Period und TimeSlice wurden eingeführt, um eine gleichzeitige Aktivierung aller Threads und damit eine 100%ige Auslastung des Hauptprozessors zu verhindern.

Während der Ausführung eines Threads muss der Thread seine ausstehenden Meldungen abarbeiten. Beispielsweise muss das Modul Viewer alle hiermit zusammenhängenden Bildobjekte aktualisieren. Wenn keine Meldungen anstehen, deaktiviert sich der Thread selbst und gibt die Kontrolle an das Betriebssystem zurück. Das Betriebssystem wechselt dann sofort zum nächsten aktiven Thread. Anders gesagt, kann ein Thread seine eigene Ausführung unterbrechen – sogar dann, wenn die Zeitspanne TimeSlice des Betriebssystems noch nicht abgelaufen ist (was bei realen Anwendungen häufig vorkommt).

☞ Die Module "Datenbankspion", "DDE-Server", "LogWin" und "ODBC-Laufzeitprogramm" unterstützen keinen Parameter Time-Slice. Daher wird jeder Thread nach Abarbeitung aller ausstehenden Meldungen inaktiv, bis die nächste Meldung Period für den jeweiligen Thread auftritt.

Der Hintergrund-Task bildet eine Ausnahme zum gerade beschriebenen Mechanismus der Thread-Ausführung. Die Ausführung des Hintergrund-Tasks wird nachstehend erläutert.

2.4.3 Ausführen/Wechseln des Hintergrund-Tasks

Der Hintergrund-Task führt Skripte aus den Arbeitsblättern "Mathematik" und "Scheduler" aus (z.B. Meldungen von den Arbeitsblättern "Alarm" und "Trend"). Außerdem führt der Hintergrund-Task alle Rezeptur- und Report-Befehle aus, wenn die Funktionen **Receipe()** bzw. **Report()** während der Laufzeit ausgeführt werden.

Bei den Tasks "Alarm", "Mathematik", "Scheduler" und "Trend" handelt es sich zwar nicht um Threads, jedoch kann deren Zeitdauer Period in der Datei **Program Settings.ini** im Programmverzeichnis von "WinStudio" festgelegt bzw. geändert werden.

Die Standardwerte für Period lauten wie folgt (in Millisekunden):

Program Settings.INI

```
[Period]
Math=100
Sched=50
Alarm=100
Trend=1000
```

Diese Werte bedeuten, dass "WinStudio" alle 100 ms eine Meldung Period an den Task Mathematik schickt. Alle 50 ms erzeugt "WinStudio" eine Meldung Period und schickt sie an den Task Scheduler usw.

HINWEIS

Wir empfehlen dringend, die Standardwerte für den Hintergrund-Task nicht zu verändern, wenn es nicht unbedingt erforderlich ist. Falsches Konfigurieren dieser Parameter kann zu Fehlfunktion des gesamten Systems führen (z.B. die Auslastung des Hauptprozessors steigt auf 100% an) und/oder zu schlechter Leistung einiger Tasks.

Wenn Sie die Standardwerte für die Parameter ändern müssen, notieren Sie sich die Werte vor Ihren Änderungen, sodass Sie bei Auftreten einer Fehlfunktion zu den Originaleinstellungen zurückkehren können.

Beachten Sie, dass der Hintergrund-Task-Thread die gleiche Priorität wie andere Threads in "WinStudio" ("Treiber", "Viewer" usw.) hat; das bedeutet, dass das Betriebssystem diesen Task nicht kontinuierlich länger als 20 ms ausführt.

Der Hintergrund-Task führt die Arbeitsblätter Rezeptur auf Seite 361 und Bericht auf Seite 363 aus, wenn die Funktionen Rezeptur auf Seite 634 und Bericht auf Seite 635 aufgerufen werden. Da die Funktionen "Recipe()" und "Report()" jeweils "synchron" sind, wechselt der Hintergrund-Task nach Beginn der Ausführung dieser Funktionen so lange nicht zu einem anderen Task ("Alarm", "Mathematik", "Scheduler" bzw. "Trend"), bis er die Funktio-

nen vollständig ausgeführt hat. Üblicherweise dauert die Ausführung einer Funktion **Recipe()** bzw. **Report()** einige Millisekunden.

Der Hintergrund-Task muss zwischen den Tasks "Alarm", "Mathematik", "Scheduler" und "Trend" wechseln. Wenn der Hintergrund-Task zum Task "Scheduler" wechselt, wechselt er danach so lange nicht zu einem anderen Task ("Alarm", "Mathematik" bzw. "Trend"), bis alle "Scheduler"-Arbeitsblätter abgearbeitet sind. Nachdem alle "Scheduler"-Arbeitsblätter abgearbeitet sind, führt der Hintergrund-Task den "Scheduler" so lange nicht wieder aus, bis er die nächste Meldung Period für den Task "Scheduler" empfängt.

Der Hintergrund-Task wendet bei der Ausführung der Tasks "Alarm" und "Trend" das gleiche Verhalten an – wenn der Hintergrund-Task zum Task "Alarm" bzw. "Trend" wechselt, wechselt er danach so lange nicht zu einem anderen Task, bis alle ausstehenden Meldungen bearbeitet sind. Dann führt der Hintergrund-Task den Task "Alarm" bzw. "Trend" so lange nicht wieder aus, bis "WinStudio" die nächste Meldung Period für den jeweiligen Task erzeugt.

Typischerweise führt der Hintergrund-Task die Tasks "Alarm", "Scheduler" und "Trend" in wenigen Millisekunden aus. Die Ausführung des Tasks "Mathematik" kann jedoch länger dauern, da dieser üblicherweise Schleifen und komplexe Skripte enthält. Daher kann der Mechanismus zum Ausführen der Tasks "Alarm", "Scheduler" und "Trend" nicht auf den Task "Mathematik" übertragen werden.

Der Hintergrund-Task führt den Task "Mathematik" nicht länger als 10 ms lang kontinuierlich aus und wechselt dann zu einem anderen Task (wie z.B. dem "Scheduler"). Dann kann der Hintergrund-Task den Task "Mathematik" in den nächsten 50 ms nicht wieder aufrufen, jedoch kann der Hintergrund-Task während dieser Zeitspanne von 50 ms andere Tasks ("Alarm", "Rezeptur", "Bericht", "Scheduler" oder "Trend") ausführen. Nachdem der Hintergrund-Task alle "Mathematik"-Arbeitsblätter ausgeführt hat, beginnt er erst dann wieder einen neuen Durchlauf durch die "Mathematik"-Arbeitsblätter, wenn "WinStudio" eine neue Meldung Period für den Task "Mathematik" erzeugt.

Hier ist es wichtig, noch einmal zu betonen, dass dieses Verfahren angelegt wurde, um eine ständige 100%ige Auslastung des Hauptprozessors zu vermeiden.

HINWEIS

Wir empfehlen, bei der Verwendung der Funktion Berechnung auf Seite 633 in einem Arbeitsblatt für den "Scheduler" oder für ein Bildobjekt (z.B. die Animation) vorsichtig vorzugehen.

Wenn der Task "Scheduler" eine Funktion **Math()** ausführt, kann so lange kein anderer Task durch den Hintergrund-Task ausgeführt werden, bis der "Scheduler" das gesamte "Mathematik"-Arbeitsblatt ausgeführt hat, das durch die **Math()**-Funktion aufgerufen wird. Dieser Vorgang dauert möglicherweise mehrere Millisekunden oder sogar Sekunden – je nachdem, wie das Script im Arbeitsblatt "Mathematik" konfiguriert wurde (vor allem für Schleifen).

Wenn Sie eine **Math()**-Funktion für ein Bildobjekt konfigurieren, unterbricht der "Viewer" die Bildschirmaktualisierung so lange, bis der "Viewer" das gesamte "Mathematik"-Arbeitsblatt ausgeführt hat, das durch die Funktion **Math()** aufgerufen wurde.

Wenn Sie die Funktion **Math()** für den Task "Scheduler" oder für ein Bildobjekt aufrufen müssen, schlagen wir das folgende Verfahren vor, um Verarbeitungsverzögerungen zu vermeiden: Durch Einführung eines Hilfs-Tags können Sie verhindern, dass der Hintergrund-Task das Arbeitsblatt "Mathematik" beim jeweils nächsten Scan erst wieder ausführt, wenn der Hilfs-Tag auf den Wert 1 zurückgesetzt wurde.

1. Erzeugen Sie ein Hilfs-Tags und tragen Sie dessen Namen im Feld "Ausführung" des auszuführenden "Mathematik"-Arbeitsblatts ein. Wenn der Hintergrund-Task das "Mathematik"-Arbeitsblatt scannt, führt "WinStudio" dieses Arbeitsblatt aus.
2. Setzen Sie den Tag in der letzten Zeile des "Mathematik"-Arbeitsblatts auf den Wert 0 zurück.
3. Setzen Sie im Scheduler oder im Bildobjekt den Hilfs-Tag auf den Wert 1. Dann führt der Hintergrund-Task das "Mathematik"-Arbeitsblatt einmalig aus. Erst wenn der Hilfs-Tag erneut auf den Wert 1 gesetzt wird, führt der Hintergrund-Task das"Mathematik"-Arbeitsblatt erneut aus.

3 Installation

3.1 Installation, Allgemein

Dieser Abschnitt enthält eine Anleitung für das Installieren von"Rexroth WinStudio", Engineering, EmbeddedView und CEView.

3.2 Systemanforderungen

Im folgenden werden die Mindestsystemanforderungen für die Installation und den Betrieb der Software "Rexroth WinStudio" beschrieben.

Entwicklung / Projekt-Server / Projekt-Client

Zur Installation und zum Ausführen der vollständigen Software "Rexroth WinStudio" benötigen Sie:

- Einen Windows-kompatiblen Rechner oder ein zu Windows Embedded kompatibles Gerät mit einer Standardtastatur, einer Maus sowie einem Monitor, der mindestens dem SVGA-Standard entspricht.
- Ein Betriebssystem „Windows“ bzw. „Windows Embedded“, das derzeit von Microsoft unterstützt wird; dies sind aktuell:
  - Microsoft Windows XP mit Service Pack 3 oder höher
  - Microsoft Windows 7, alle Versionen
  - Microsoft Windows XP Embedded Service Pack 3
  - Microsoft Windows Embedded Standard 7 (2009)
- Microsoft Internet Explorer 6.0 oder höher
- mindestens 500 MB freien Speicherplatz auf der Festplatte
- Ethernet-Adapter oder kabellose Übertragung
- CD-ROM-Laufwerk (optional, zum Installieren der Anwendung; diese kann auch von unserer Website herunter geladen werden)
- USB-Anschluss (optional, zur Verwendung mit einer Hardkey-Lizenz)
- serielle COM-Anschlüsse und -Adapter (optional, zur Verwendung für die direkte Kommunikation mit SPS und anderen Geräten)

Auf Geräten mit Windows Embedded Compact kann die vollständige Software "Rexroth WinStudio" nicht installiert werden.

Projekt-Server – Embedded

Um ein Gerät über EmbeddedView oder CEView als Projekt-Server zu betreiben (d.h. um es als Host für die Laufzeitdateien des Projekts einzusetzen), benötigen Sie:

- ein zu Windows Embedded kompatibles Gerät
- ein Betriebssystem mit Windows Embedded, das derzeit von Microsoft unterstützt wird; dies sind aktuell:
  - Microsoft Windows XP Embedded Service Pack 3
  - Microsoft Windows Embedded Standard 7 (2009)
  - Microsoft Windows Embedded Compact (früher Windows CE bzw. Windows Mobile genannt), Version 5.0 oder höher
- mindestens 500 MB freien Speicherplatz auf der Festplatte
- Ethernet-Adapter oder kabellose Übertragung
- USB-Anschluss (optional, zur Verwendung mit einer Hardkey-Lizenz)
- serielle COM-Anschlüsse und -Adapter (optional, zur Verwendung für die direkte Kommunikation mit SPS und anderen Geräten)

Projekt-Client – Embedded

Um ein Gerät als Projekt-Client mit EmbeddedView oder CEView zu betreiben, benötigen Sie:

- ein zu Windows Embedded kompatibles Gerät.
- ein Betriebssystem mit Windows Embedded, das derzeit von Microsoft unterstützt wird; dies sind aktuell:
  - Microsoft Windows XP Embedded Service Pack 3
  - Microsoft Windows Embedded Standard 7 (2009)
  - Microsoft Windows Embedded Compact (früher Windows CE bzw. Windows Mobile genannt), Version 5.0 oder höher
- Ethernet-Adapter oder kabellose Übertragung

Projekt-Client – Thin

Um ein Gerät als Projekt-Client unter Verwendung des Programms Secure Viewer oder des Browser-basierten Thin Client zu betreiben, benötigen Sie:

- einen Windows-kompatiblen Rechner oder ein zu Windows Embedded kompatibles Gerät.
- einen Windows Desktop, Server oder ein eingebettetes Betriebssystem, das derzeit von Microsoft unterstützt wird; dies sind aktuell:
  - Microsoft Windows XP mit Service Pack 3 oder höher
  - Microsoft Windows Vista mit Service Pack 1 oder höher
  - Microsoft Windows 7, alle Versionen
  - Microsoft Windows Server 2003 mit Service Pack 2 oder höher
  - Microsoft Windows Server 2008, alle Versionen
  - Microsoft Windows XP Embedded Service Pack 3
  - Microsoft Windows Embedded Standard 7 (2009)
  - Microsoft Windows Embedded Compact (früher Windows CE bzw. Windows Mobile genannt), Version 5.0 oder höher
- Microsoft Internet Explorer 6.0 oder höher
- Ethernet-Adapter oder kabellose Übertragung

3.3 Installation der Software

Engineering

Die Entwicklungsumgebung "Rexroth WinStudio" ist in IndraWorks Engineering integriert.

Die Anwendung zur Softwareentwicklung können Sie vom Datenträger "Rexroth IndraMotion MLC/XLC"installieren.

Runtime

Bei der Installation der Laufzeitumgebung auf dem Zielgerät ist hinsichtlich des Gerätetyps und des Applikationstyps zu unterscheiden:

- Für die Ausführung von "HMI-Standardanwendungen" auf VPx- oder VEx- Geräten mit Windows XP/Windows7 bzw Windows XP embedded installieren Sie "IndraWorks Operation" auf dem Zielgerät.
- Für die Ausführung von reinen "WinStudio-Anwendungen" auf VPx- oder VEx- Geräten mit Windows XP/Windows7 bzw Windows XP embedded installieren Sie "WinStudio-Runtime". Alternativ können Sie sich aus Indraworks Engineering mit dem VPx-Gerät verbinden und die Laufzeit notwendigen Systemdateien installieren.

- Für die Ausführung von "WinStudio-Anwendungen" auf VEx-Geräten mit Windows CE starten Sie IndraWorks Engineering, verbinden sich mit dem Zielgerät und installieren von dort die Laufzeitumgebung.



- Sie müssen über Administratorrechte auf Ihrem PC verfügen, um die Entwicklungsanwendung installieren oder deinstallieren zu können.
- Eine ggf. vorhandene ältere Version der Entwicklungsanwendung muss vor der Installation einer neuen Version deinstalliert (oder in ein anderes Verzeichnis verschoben) werden. Außerdem kann die gleiche Version der Entwicklungsumgebung nicht in zwei unterschiedlichen Verzeichnispfaden des selben PCs installiert werden.

Installation von CEView auf Ihrem Windows-Embedded-Gerät

CEView ist die Laufzeit-Komponente für "WinStudio"-Projekte auf Windows-Embedded-Geräten. CEView muss auf Ihrem Gerät installiert sein, "bevor" Sie Ihr Projekt dorthin übertragen und ausführen können.

Speicherort der Dateien

Aufgrund der unterschiedlichen Architekturen von Windows-Embedded-Geräten gibt es für jede Kombination von Betriebssystem-Version und Geräteprozessor eine eigene Laufzeitumgebung. Die verschiedenen Laufzeitumgebungen finden Sie in folgendem Verzeichnis:

[…]\WinStudio\Redist

Die spezielle Laufzeitumgebung für Ihr konkretes Gerät liegt in folgendem Verzeichnis:

[…]\WinStudio\Redist\<version>\<processor>

… wobei gilt:

- **version** ist die Betriebssystemversion des Gerätes, auf das CEView installiert wird:
  - Der Ordner **\WinCE 5.0** enthält die Dateien für Windows CE und Windows Mobile 5.0 oder höher; und
  - Der Ordner **\WinEmbedded** enthält die Dateien für Windows XP Embedded und Windows Embedded 7.
- **processor** ist der von Ihrem Windows-CE-Gerät verwendete Prozessor. WinStudio stellt für jeden Prozessor, der zur Zeit vom Betriebssystem Windows CE unterstützt wird (z.B. Pocket2003-ArmV4, ArmV4i, x86), eine Laufzeitumgebung zur Verfügung. Weitere Informationen entnehmen Sie bitte der Herstellerdokumentation für das jeweilige Gerät.

So installieren Sie CEView auf einem Windows-Embedded-Gerät



Wir empfehlen den Einsatz von TCP/IP, sofern möglich.

1. Stellen Sie sicher, dass Ihr Windows-Embedded-Gerät mit der Entwicklungsstation verbunden ist.
2. Schalten Sie das Gerät ein.

   Der Dialog "Remote Agent" sollte sich automatisch öffnen. Ist dies nicht der Fall, müssen Sie die Datei "CEServer.exe" manuell auf dem Gerät starten.

3. - Auf einer Windows Plattform befindet sich die Datei "CEServer.exe" im

     Ordner **[…]\WinStudio\Bin**.

   - Auf einer Windows-CE Plattform befindet sich die Datei im Ordner **[…]\WinStudio**.

4. Führen Sie "CEServer.exe" auf der Zielstation aus und klicken Sie auf "Setup", wenn der Dialog **Remote Agent** angezeigt wird.

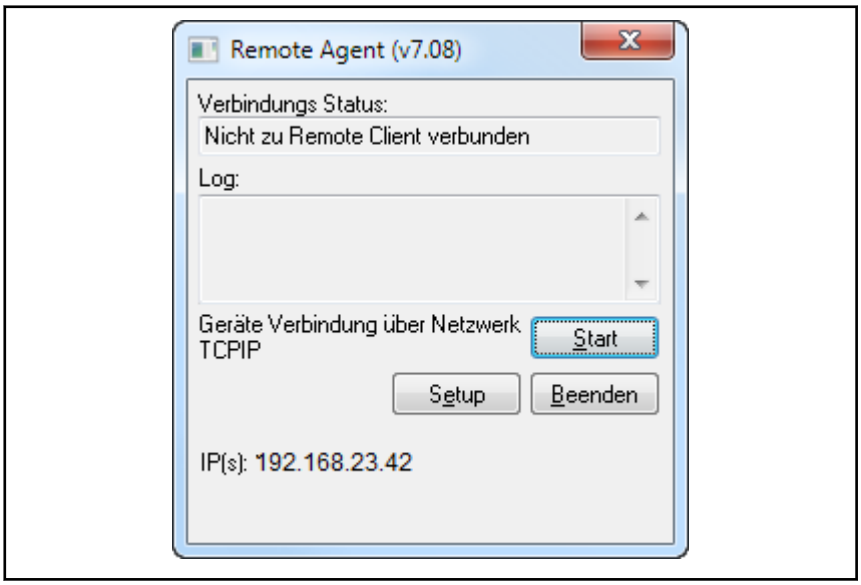


Abb.3-1: Dialog Remote Agent

Der Dialog "Setup" wird angezeigt:

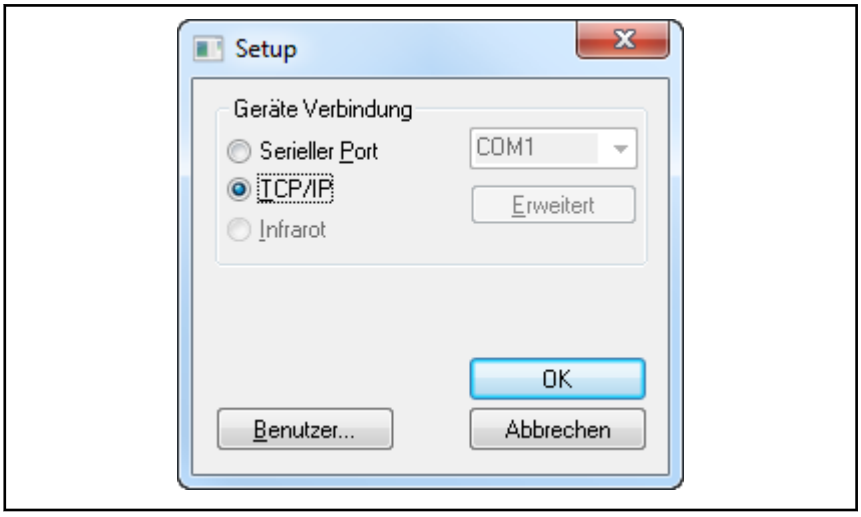


Abb.3-2: Dialog Setup

Die Parameter in diesem Dialog dienen der Konfiguration der Kommunikation zwischen Entwicklungs- und Zielsystemen:

- **Serieller Port**: Zum Herstellen einer seriellen Verbindung zum Projekt Server wählen Sie "Serieller Port" aus und wählen Sie dann den spezifischen Port aus der Auswahlliste aus. Durch Klicken auf **Erweitert** wird der Dialog "Erweiterte Einstellungen" angezeigt.

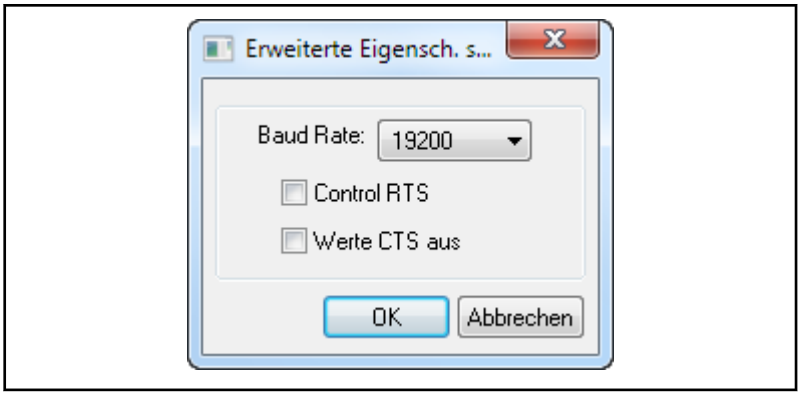


Abb.3-3: Dialog Erweiterte Einstellungen

Die Parameter in diesem Dialog dienen der Steuerung des Flusses zwischen Ihren Ziel- und Entwicklungsstationen: Ändern Sie diese bei Bedarf und klicken Sie auf **OK** zum Schließen des Dialogs "Seriell - Weitere Einstellungen".

- **TCP/IP**: Aktivieren Sie diese Option, um die Verbindung zur Entwicklungsstation über TCP/IP herzustellen.

Für bessere Leistung empfehlen wir die Verwendung einer TCP/IP-Verbindung anstelle einer seriellen Verbindung.

---

5. Nach Beendigung klicken Sie zum Schließen des Dialogs "Setup" auf **OK**. Lassen Sie das Programm Remote Agent aber in der Remote-Arbeitsstation weiter laufen.

Nach dem Konfigurieren der Zielgeräte zum Empfangen von Daten befolgen Sie die folgenden Schritte, um die Laufzeitumgebung (CEView) von der Entwicklungsstation auf das Zielgerät zu übertragen:

1. Starten Sie IndraWorks Engineering auf Ihrem PC.
2. Öffnen Sie ein Projekt oder erstellen Sie ein neues Projekt.
3. Fügen Sie ein Visualisierungsgerät "VPx oder VEx mit Windows XP/ Windows7 bzw Windows XP embedded oder ein VEx mit Windows CE" ein und wählen Sie im HMI-Assistenten den Applikationstyp "WinStudio-Anwendung" aus.
4. Öffnen Sie das Kontextmenü des Elements "WinStudio" und wählen den Eintrag **WinStudio-Zielplattform**.

   Der Dialog "Fern-Management" wird angezeigt.

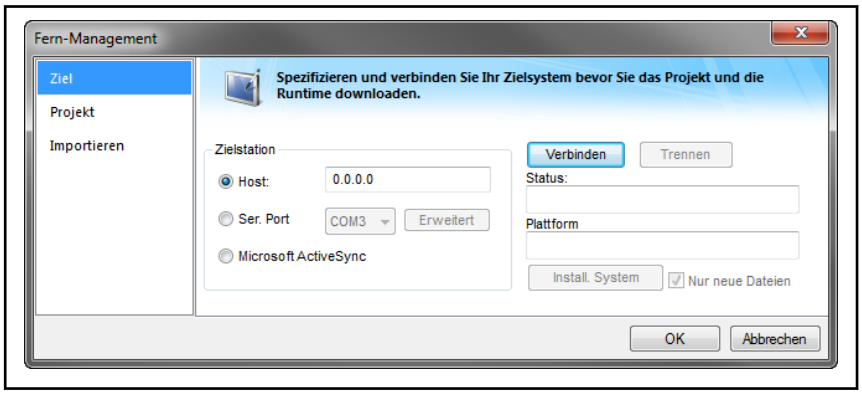


Abb.3-4: Dialog Remote-Verwaltung

5. Wählen Sie die Registerkarte **Ziel** und selektieren Sie eine der folgenden Optionen zum Festlegen der Verbindungsart mit dem Zielsystem.
   - **Netzwerk IP**: Wählen Sie diese Option aus und geben sie die IP-Adresse des Zielsystems in das Textfeld ein, wenn Sie beim Konfigurieren des Zielsystems eine TCP/IP-Verbindung angegeben haben.
   - **Ser. Port**: Wählen Sie diese Schaltfläche aus und wählen Sie einen Port aus der Liste der Combobox aus, wenn Sie in beim Konfigurieren des Zielsystems eine serielle Verbindung angegeben haben.
   - **Microsoft ActiveSync**: Wählen Sie diese Schaltfläche zum Verbinden mit Geräten aus, auf denen ActiveSync aktiviert ist.
6. Klicken Sie auf **Verbinden**, um eine Verbindung zwischen der Entwicklungsanwendung und dem Bediengerät herzustellen.

   Im Textfeld **Status**wird der aktuelle Status der Verbindung angezeigt.

   Bei erfolgreichem Verbindungsaufbau werden die Gerätespezifikationen der verbundenen Bedienstation im Textfeld **Plattform**dargestellt.
7. Klicken Sie auf die Schaltfläche **Install. System**, um die Dateien für die zugehörige Laufzeitumgebung auf die Bedienstation zu übertragen.
8. Wenn die Installation beendet ist, klicken Sie auf **Trennen**.

3.4 Starten der Engineering Software

So starten Sie "Rexroth WinStudio":

- Führen Sie einen Doppelklick auf das Symbol "IndraWorks Engineering" auf dem Desktop aus oder
- Wählen Sie

  Start ▸ **Alle Programme** ▸ **Rexroth** ▸ **IndraWorks** ▸ **Engineering**.

Öffnen oder erstellen Sie ein Projekt mit einem Visualisierungsgerät.

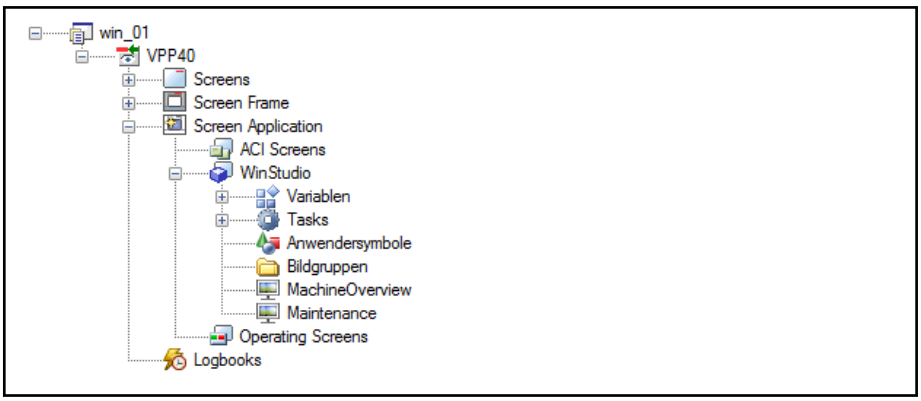


Abb.3-5: Der Ordner WinStudio

Wählen Sie ein Element unterhalb des WinStudio Ordners aus.

Öffnen Sie mit einem Doppelklick auf das Element die Engineering Software "Rexroth WinStudio".



Die Entwicklungsumgebung kann mit allen Anzeigeeinstellungen ausgeführt werden.

Zum Einrichten eines angenehmeren Umfelds empfehlen wir allerdings, die Anzeigeeinstellungen auf eine Auflösung von 1024x768 (oder höher) einzustellen und mehr als 256 Farben zu verwenden.

Die Bildschirmauflösung des Projekts (Bildschirmgröße) ist unabhängig von der Bildschirmauflösung des Betriebssystems.

4 Lizenzierung

4.1 Arten des Lizenzschutzes

"Rexroth WinStudio" und CEView unterstützen die folgenden Arten des Lizenzschutzes:

Hardkey

Ein gekapselter Chip, der physisch an die USB-Schnittstelle des Rechners angeschlossen werden muss.

Die Softwarelizenz befindet sich im Hardkey, und Sie können diese Lizenz nicht gleichzeitig mit mehr als einer weiteren Kopie dieser Software im Netzwerk teilen. Wenn Sie den Hardkey an einen anderen Rechner anschließen, übertragen Sie die Lizenz effektiv auf diesen Rechner.

Bei Verwendung des USB-Hardkeys kann der entsprechende USB-Anschluss mit keinem anderen Gerät geteilt werden.

Softkey

Bei der Installation von "Rexroth WinStudio" erzeugt das Programm eine eindeutige Kennung, die als "Site-Kennung" bezeichnet wird. Sie können diese Site-Kennung an Ihren Softwarehersteller schicken, der daraus einen zur Ihrer Site-Kennung passenden Lizenzcode – den "Site-Schlüssel" – erzeugt.

Mit dem Site-Schlüssel wird die Lizenz für "Rexroth WinStudio" bzw. CEView auf Ihrem Rechner bzw. Windows-Embedded-Gerät installiert.

4.2 Lizenzeinstellungen

Sowohl mit der Hardkey- als auch mit der Softkey-Lizenz werden die folgenden Parameter gesetzt:

Version

Die Gesamtversion auf Seite 49 der Software "WinStudio", z.B. "7.3". (Dies beinhaltet "nicht" ggf. vorhandene Service-Packs.)

Treiber

Die Anzahl der Direktkommunikations-Treiber auf Seite 379, die gleichzeitig aktiviert werden können.

Produkttyp

Gibt an, welche Funktionen und Beschränkungen für die jeweilige Anwendung freigegeben sind (z.B. die Höchstzahl unterstützter Tags).

Ausführungsmodus

Legt eine der folgenden Optionen fest:

- **Nur Engineering**: Möglichkeit zur Entwicklung eines "WinStudio"-Projekts und kurzzeitiger Ausführung des Projekts zu Testzwecken. Diese Lizenz kann nicht als langfristige Laufzeitlizenz genutzt werden.
- **Nur Laufzeit**: Möglichkeit zum unbegrenzten Ausführen eines "WinStudio"-Projekts. Diese Lizenz kann nicht zum Entwickeln oder Verändern des Projekts verwendet werden.
- **Engineering + Laufzeit**: Möglichkeit zum Entwickeln und zeitlich unbegrenzten Ausführen eines "WinStudio"-Projekts.

Weitere Informationen finden Sie in <Ausführungsmodi>.

Thin Clients

Die Anzahl an Thin Clients auf Seite 473, die sich gleichzeitig mit dem Server verbinden können. Jede Lizenz erlaubt die Verbindung eines einzelnen Thin Clients. Falls Sie mehr als einen Client verwenden möchten, benötigen Sie eine spezielle Lizenz.

Secure Viewers

Die Anzahl an Secure-Viewer-Clients auf Seite 473, die sich gleichzeitig mit dem Server verbinden können. Jede Lizenz erlaubt die Verbindung eines einzelnen Secure Viewer-Clients. Falls Sie mehr als einen Client verwenden möchten, benötigen Sie eine spezielle Lizenz.

SMA-Clients

Die Anzahl an <Studio-Mobile-Access-Clients> (SMA-Clients), die sich gleichzeitig mit dem Server verbinden können. Jede Lizenz erlaubt die Verbindung eines einzelnen SMA-Clients. Falls Sie mehr als einen Client verwenden möchten, benötigen Sie eine spezielle Lizenz.

4.3 Lizenzarten

"Rexroth WinStudio" und CEView unterstützen die folgenden Lizenzarten, die sich hinsichtlich des zur Verfügung stehenden Funktionsumfangs unterscheiden:

Ausführungsmodus	WinStudio	CEView
Demolizenz	J	J
Lite-Lizenz	J	J
Engineering-Lizenz	J	N
Runtime-Lizenz	J	J
Engineering- und Runtime-Lizenz	J	N

Demolizenz

Wenn Sie die WinStudio-Demolizenz aktivieren können Sie für einen Zeitraum vom 30 Tagen den vollen Funktionsumfang von WinStudio einschließlich der Engineering-Funktionen nutzen. Nach Ablauf des Testzeitraums ist wieder die auf dem Computer freigeschaltete Lizenz aktiv. Die Testlizenz kann auf jedem Gerät nur einmal aktiviert werden.

Auch ohne Aktivierung der Demolizenz erlaubt CEView die Ausführung der WinStudio-Laufzeit für einen Zeitraum von 2 Stunden. Danach wird die Ausführung beendet. Sie können jedoch CEView erneut starten.

Lite-Lizenz

Die Lite-Lizenz erlaubt die Verwendung der meisten Engineering-Funktionen sowie die Ausführung der WinStudio-Runtime für Projekte, die für den Produkttyp "WinStudio-Lite" erstellt wurden. Mit der Lite-Lizenz können Sie in der WinStudio-Applikation maximal 500 Variablen verwenden. Erweiterte Funktionen wie "Alarme", "Trends", "Reporte", "Rezepte" und "externe Datenbanken" können mit der Lite-Lizenz nicht verwendet werden.

Engineering-Lizenz

Gibt alle Engineering-Funktionen für unbegrenzte Zeit frei. Es stehen folgende Lizenzabstufungen zur Auswahl: 1500, 4000, 64000, 512000 Variablen.

In diesem Modus können außerdem die Laufzeit-Tasks und Werkzeuge zur Fehlerbeseitigung ("Datenbankspion", "Output"-Fenster und "LogWin"-Modul) 24 Stunden lang ununterbrochen ausgeführt werden. Nach der 24-stündigen Zeitspanne werden diese Tasks zwar beendet, können aber erneut gestartet und weitere 24 Stunden lang ausgeführt werden usw. Diese Lizenz kann nur zur Entwicklung und zum Testen verwendet werden.

Runtime-Lizenz

Ermöglicht das Ausführen aller Laufzeit- und Fehlerbeseitigungswerkzeuge ("Datenbankspion", "Ausgabe"-Fenster und "LogWin"-Modul) für unbegrenzte Zeit. Es stehen folgende Lizenzabstufungen zur Auswahl: 1500, 4000, 64000, 512000 Variablen. Engineering-Funktionen können mit diese Lizenz nicht genutzt werden.

Engineering- und Runtime-Lizenz

Schaltet alle Engineering-Werkzeuge, Laufzeit-Tasks und Werkzeuge zur Fehlerbeseitigung ("Datenbankspion", "Ausgabe"-Fenster und "LogWin"-Modul) für unbegrenzte Zeit frei.

☞ Das Remote-Verwaltungswerkzeug (**WinStudio-Zielplattform** im Kontextmenü des Ordners "WinStudio") ist unabhängig vom Ausführungsmodus immer verfügbar, so dass Dateien auf Remote-Stationen immer herunter geladen werden können.

4.4 Produktversionen

"Rexroth WinStudio" und CEView sollten beide die gleiche Versionsnummer haben. Diese hat den Aufbau "X.Y+SPWW" (z.B. "Rexroth WinStudio v7.3" und "CEView v7.3"), wobei gilt:

- "X" steht für die **Familienversion**. Die Familienversion ändert sich nur, wenn den Produkttechnologien und -konzepten wichtige Verbesserungen hinzugefügt werden.
- "Y" steht für die **Unterversion**: Die Unterversion ändert sich, wenn dem Produkt kleinere Verbesserungen und/oder neue Funktionen hinzugefügt werden.
- "WW" steht für das **Service-Pack**. Die Service-Pack-Version ändert sich, wenn Sie Ergänzungspakete installieren müssen, um folgendes zu erreichen:
  - Dateien der bereits installierten Version aktualisieren
  - Fehler im Produkt beheben (Showstopper und nicht umgehbare Fehler)
  - kleinere Verbesserungen vor Freigabe der nächsten Produktversion bereitstellen

  Jedes Service-Pack-Release ersetzt die jeweils vorhergehenden Service-Pack-Releases. Beispielsweise enthält SP2 den gesamten Inhalt von SP1 und alle neu aktualisierten Dateien, Fehlerbehebungen und Verbesserungen. SP3 enthält den gesamten Inhalt von SP2 und alle neu aktualisierten Dateien, Fehlerbehebungen und Verbesserungen, usw.

Sowohl "Rexroth WinStudio" als auch CEView können in Vorgängerversionen des Produkts erstellte Projekte ausführen. Jedoch können ältere Versionen von "WinStudio" und CEView keine jüngeren Versionen des Produkts erstellte oder veränderte Projekte ausführen.

Zum Beispiel können Projekte der Version 6.1 nicht mit "WinStudio" Version 6.0 ausgeführt werden, Projekte der Version 6.0 aber mit "WinStudio" Version 6.1.

Wir stellen jede Lizenz für eine konkrete "Familien-" und "Unterversion" (X.Y) aus und diese Lizenz ist für nur diese Version (einschließlich der Service-Packs) gültig. Für eine neuere "Familienversion" bzw. "Unterversion" des Produkts ist die Lizenz nicht gültig. Wenn also eine neue Version von "WinStudio" oder CEView installiert wird, muss ein Upgrade der Lizenz auf die neu zu installierende Version durchgeführt werden. Wenn nur ein Service-Pack installiert wird, muss für Lizenz kein Upgrade durchgeführt werden.

4.5 Aktivieren einer WinStudio-Lizenz

Die einzelnen "WinStudio Runtime"- und "WinStudio Engineering und Runtime"-Lizenzen sind kostenpflichtig und müssen durch Eingabe eines Lizenzierungsschlüssels aktiviert werden.

Ihren persönlichen Lizenzschlüssel finden Sie auf dem Lizenzblatt.

Um eine Lizenz zu aktivieren, öffnen Sie in IndraWorks Engineering über**Extras ▸ Optionen ▸ Allgemein ▸ Softwarelizenzen** den Lizenzierungsdialog und wählen Sie den Eintrag "WinStudio" aus.

Mit der Schaltfläche "Lizenzieren… " können Sie den Aktivierungsdialog öffnen und den Aktivierungsschlüssel eingeben.

Nach der Aktivierung der Lizenz ist es ggf. notwendig, WinStudio-Engineering oder IndraWorks Operation neu zu starten.

Lizenz auf Dongle speichern

Der Aktivierungsschlüssel ist immer nur für einen PC gültig. Um eine Lizenz nacheinander auf mehreren PCs nutzen zu können, gibt es die Möglichkeit die Lizenz auf einen Dongle zu übertragen. Die Lizenz ist dann solange gültig wie der Dongle an dem USB-Anschluss des Computers eingesteckt ist.

Windows CE-Lizenzen (VEx) können nicht auf einen Dongle übertragen werden.

Lizenzaktivierung auf VEx-Geräten

Um die WinStudio-Lizenz auf einem VEx-Gerät mit dem Betriebssystem Windows CE zu aktivieren, öffnen Sie auf dem VEx-Gerät im Windows-Startmenü über **Programme ▸ Rexroth ▸ WinStudio ▸ Lizenzverwaltung** den Lizenzierungsdialog und geben Sie den Aktivierungsschlüssel ein. Auf neuen VEx-Geräten ist die Lizenzierungssoftware bereits vorinstalliert. Falls Sie bereits ein VEx-Geräte ohne vorinstallierte Lizenzierungssoftware besitzen, installieren Sie zunächst die IndraWorks-Visualisierungssoftware auf dem VEx-Gerät und aktivieren Sie danach die WinStudio-Lizenz.

☞ Die IndraWorks-Lizenzkomponenten für Windows CE sind auf Neugeräten bereits vorinstalliert. Falls Sie ein bereits vorhandenes VEx-Gerät mit der neuen Software aktualisieren möchten, installieren Sie zunächst die Laufzeitkomponenten auf dem VEx-Gerät über das Kontextmenü des VEx-Gerätes in IndraWorks Engineering mit der Funktion Software installieren. Ggf. ist die Verknüpfung nicht vorhanden. Starten Sie in diesem Fall den Lizenzierungsdialog über \StorageCard\Rexroth\WinStudio\bin \IndraWorks.LicenseImport.CE.exe.

Die Lizenzierungskomponente benötigt ".Net CompactFramework 2.0", so dass ggf. die Firmware zu aktualisieren ist.

4.6 Ungültige Lizenzen

Wenn Sie ein Projekt mit dem WinStudio-Editor öffnen, das auf einem anderen Computer mit einer höheren WinStudio-Lizenz erstellt wurde, sind die lizenzpflichtigen Elemente in WinStudio-Engineering deaktiviert, d.h. eine Erstellung oder Bearbeitung dieser Elemente ist in diesem Fall nicht möglich.

Beim Start von IndraWorks Operation erscheint eine Hinweismeldung, falls die vorhandene WinStudio-Lizenz für das aktuelle Projekt nicht ausreicht.

Bei unzureichender oder fehlender Lizenz werden keine WinStudio-Bilder angezeigt.

☞ Falls Sie das aktuelle Projekt dennoch nutzen möchten, können Sie die Demo-Lizenz aktivieren. Verwenden Sie zur Lizenzaktivierung den IndraWorks-Lizenzierungsdialog (wie oben beschrieben).

5 Entwicklungsumgebung

5.1 Die Entwicklungsumgebung, Allgemeines

Die Entwicklungsumgebung von "Rexroth WinStudio" ist in IndraWorks Engineering integriert. Diese umfasst die folgenden Komponenten:

- Projektexplorer: Ordner "WinStudio" mit Unterordnern
- Verschiedene Editoren für WinStudio-Screens, Mathematikblätter, etc.
- Symbolleisten
- Werkzeugfenster "WinStudio Screen Tools"
- Werkzeugfenster "WinStudio Bibliothek"
- Ausgabefenster "LogWin"
- Ausgabefenster "Winstudio Querverweissuche"
- Ausgabefenster "Winstudio Datenbank-Spion"
- Ausgabefenster "Winstudio Symbole"

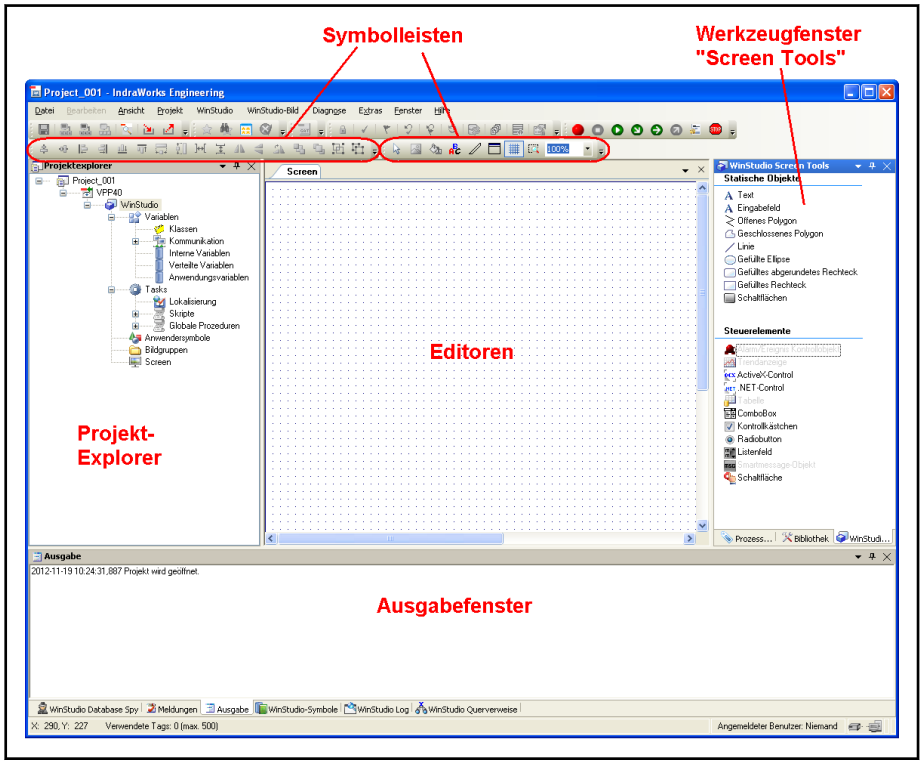


Abb.5-1: Die WinStudio-Entwicklungsumgebung in Indraworks

5.2 WinStudio-Symbolleisten

5.2.1 WinStudio-Symbolleisten, Anpassen

Bei der Erstellung einer WinStudio-Anwendung unterstützen Sie mehrere Symbolleisten mit zahlreichen, kontextabhängigen Funktionen. Einen Überblick über die verfügbaren Symbolleisten erhalten Sie, wenn Sie im Hauptmenü von IndraWorks **Extras** ▸ **Anpassen** auswählen. Es öffnet sich ein Konfigurationsdialog, in dem Sie die häufig benötigten Symbolleisten aktivieren und selten verwendete Symbolleisten ausblenden können.

Entwicklungsumgebung

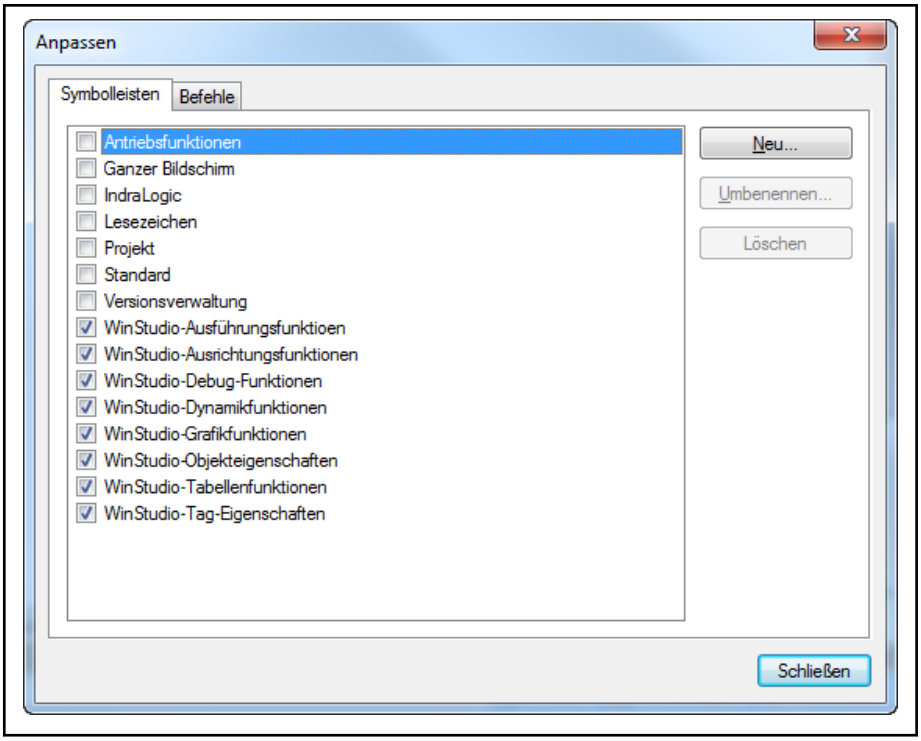


Abb.5-2: Verfügbare WinStudio-Symbolleisten

Wenn Sie eine oder mehrere Symbolleisten hinzufügen, werden diese an einer voreingestellten Position angezeigt. Sie können die Symbolleisten jederzeit beliebig anordnen, indem Sie auf den linken Rand einer Symbolleiste klicken und diese mit gedrückter Maustaste verschieben.

Die WinStudio-Funktionen, die in den Symbolleisten angeboten werden, sind kontextabhängig. d

Sie können nur aktiviert werden, wenn der entsprechende Editor geöffnet und ggf. ein oder mehrere Objekte ausgewählt sind.

Alle Funktionen sind zusätzlich bei geöffnetem WinStudio-Editor im Hauptmenü von IndraWorks im Menüpunkte "WinStudio-<Elementname>", also z.B. "WinStudio-Mathematikblatt" oder "WinStudio-Bild", verfügbar. Auf diese Weise können Sie auf das Einblenden selten verwendeter Symbolleisten verzichten.

Im Folgenden finden Sie ein Beschreibung der einzelnen Symbolleisten.

5.2.2 Symbolleiste WinStudio-Ausführungsfunktionen

Mit Hilfe der Symbolleiste" WinStudio-Ausführungsfunktionen" können Sie die grundlegende Funktion ein WinStudio-Bildes, das im Editor geöffnet ist, testen.



Abb.5-3: Symbolleiste WinStudio-Dynamikfunktionen

Bild testen

Mit der Schaltfläche , **Bild testen**, aktivieren Sie den Testmodus des aktuell bearbeiteten WinStudio-Bildes.

Im Testmodus können Sie Animationen (z.B. dynamische Positions- oder Größenänderung von Objekten) testen.

Sie können Bildobjekte verschieben und deren Eigenschaften ändern.

Es werden jedoch keine Tasks gestartet, so dass in diesem Testmodus keine Hintergrundskripte und keine Funktionen getestet werden können, die Kommunikation mit externen Geräten erfordern.

Test beenden

Mit der Schaltfläche , **Test beenden**, beenden Sie den Testmodus und aktiviert den normalen Editiermodus.

5.2.3 Symbolleiste WinStudio-Ausrichtungsfunktionen

Die Symbolleiste "WinStudio-Ausrichtungsfunktionen" bietet zahlreiche Funktionen zum Ausrichten, Gruppieren und zur Gestaltung von Bildobjekten an.



Abb.5-4: Symbolleiste "Ausrichtungsfunktionen"

Einige Funktionen in der Symbolleiste sind nur verfügbar, wenn ein oder mehrere Objekte im WinStudio-Bild ausgewählt wurden.

Bewegt man die Maus über ein aktives "Werkzeug", wird die Funktion im Klartext angezeigt.

- **Vertikal zentrieren** : alle markierten Objekte werden an der vertikalen Mittellinie des zuletzt markierten Objekts ausgerichtet
- **Horizontal zentrieren** : alle markierten Objekte werden an der horizontalen Mittellinie des zuletzt markierten Objekts ausgerichtet
- **Linksbündig** : alle markierten Objekte werden an der linken Kante des zuletzt markierten Objekts ausgerichtet
- **Rechtsbündig** : alle markierten Objekte werden an der rechten Kante des zuletzt markierten Objekts ausgerichtet
- **Unten ausrichten** : alle markierten Objekte werden an der Unterkante des zuletzt markierten Objekts ausgerichtet
- **Oben ausrichten** : alle markierten Objekte werden an der Oberkante des zuletzt markierten Objekts ausgerichtet
- **Gleiche Breite für ausgewählte Objekte** : das zuerst markierte Objekt wird auf die Breite des zweiten gedehnt / gestaucht.
- **Gleiche Höhe für ausgewählte Objekte** : das zuerst markierte Objekt wird auf die Höhe des zweiten gedehnt / gestaucht.
- **Horizontal verteilen** : der gleiche horizontale Abstand wird zwischen mehreren markierten Objekten eingefügt
- **Vertikal verteilen** : der gleiche vertikale Abstand wird zwischen mehreren markierten Objekten eingefügt
- **Horizontal spiegeln** : das markierte Objekt wird um eine gedachte horizontale Mittellinie gedreht, bis es ein Spiegelbild des Ursprungsobjekts ist

- **Vertikal spiegeln** : das markierte Objekt wird um eine gedachte vertikale Mittellinie gedreht, bis es ein Spiegelbild des Ursprungsobjekts ist
- **Drehen** : das markierte Objekt wird um 90 Grad im Uhrzeigersinn gedreht
- **Objekt in den Hintergrund blenden** : verschiebt ein oder mehrere markierte Objekte hinter alle anderen Objekte im Bild
- **Objekt in den Vordergrund blenden** : verschiebt ein oder mehrere markierte Objekte vor alle anderen Objekte im BILD
- **Objekte gruppieren** : fasst mehrere markierte Objekte zu einem einzigen Objekt zusammenzufassen
- **Gruppierung aufheben** : teilt ein gruppiertes Objekt in seine einzelnen Bestandteile

5.2.4 Symbolleiste WinStudio-Tag-Eigenschaften



Abb.5-5: Symbolleiste WinStudio-Tag-Eigenschaften

Mit Hilfe der Symbolleiste"WinStudio-Tag-Eigenschaften" können Sie die Eigenschaften von WinStudio-Objekten bearbeiten.

- **Variable global ersetzen** : erlaubt den globalen Austausch einer Variablen durch eine äquivalente
- **Objekt-Eigenschaften ändern** : öffnet das Objekt-Eigenschaftenfenster für ein markiertes Objekt / eine markierte Objektgruppe

5.2.5 Symbolleiste WinStudio-Objekteigenschaften

Die Symbolleiste "WinStudio-Objekteigenschaften" bietet Funktionen zur Beeinflussung von Objekteigenschaften.

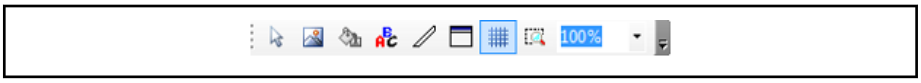


Abb.5-6: Symbolleiste "Objekteigenschaften"

Einige Funktionen in der Symbolleiste sind nur verfügbar, wenn ein oder mehrere Objekte im WinStudio-Bild ausgewählt wurden.

Bewegt man die Maus über ein aktives "Werkzeug", wird die Funktion im Klartext angezeigt.

- **Objekt auswählen** :
- **Hintergrundbild bearbeiten** :
- **Füllfarbe** :
- **Schriftart und Farbe auswählen** : öffnet für den markierten Text / die markierten Texte das Auswahlfenster in dem Fonts, Stil, Schriftgröße, Farbe und Effekte festgelegt werden können.

- **Linienfarbe** : öffnet das Auswahlfenster für die Linienfarbe für markierte Objekte und legt die Standardfarbe für neue Objekte fest
- **Hintergrundfarbe** : öffnet das Auswahlfenster für die Hintergrundfarbe des WinStudio-Bildes
- **Raster ein-/ausschalten** : steuert das Anzeigen / Verbergen des Rasters im Bild-Editor
- **Zoombereich wählen** 100% : steuert den Zoomvorgang für den mit der Lupe markierten Bereich / für das gesamte WinStudio-Bild

5.2.6 Symbolleiste WinStudio-Tabellenfunktionen



Abb.5-7: Symbolleiste WinStudio-Tabellenfunktionen

In der Symbolleiste "WinStudio-Tabellenfunktionen" sind nachstehende Schaltflächen zusammengefasst:

- **Anlegen eines Trendobjektes** : erlaubt das Anlegen eines Trendobjektes im Editorfenster
- **Anlegen einer Trendanzeige** : erlaubt das Anlegen einer Trendanzeige (alternativ zum gleichnamigen Steuerobjekt) im Editorfenster
- **Sicherheitsstufe ändern** : öffnet den Sicherheitsdialog für ein WinStudio-Bild, siehe auch Sicherheitszugriffsebenen auf Seite 408
- **Text suchen** : erlaubt das Suchen von Texten im aktiven Fenster
- **Ersetzen** : öffnet das Dialogfenster zum Ersetzen von "Nur Variablen", "Variablen und Eigenschaften", "Benutzereigenschaften", siehe auch Tags umbenennen und entfernen auf Seite 148
- **Zeile einfügen** : erlaubt das Einfügen von Zeilen in ein Arbeitsblatt
- **Objekt auswählen** : öffnet das Dialogfenster zur Eingabe einer Objekt-ID, deren Objekt im aktiven Editorfenster gefunden wird

5.2.7 Symbolleiste WinStudio-Debug-Funktionen

Die Symbolleiste "WinStudio-Debug-Funktionen" unterstützt Sie beim Testen und Fehlersuche von VB-Skripten. Sie können somit Ihre Skripte "debuggen", d.h. die Ausführung an bestimmten Stellen anhalten, Skripte in Einzelschritten ausführen und dabei die Werte von Variablen beobachten.



Abb.5-8: Symbolleiste WinStudio-Debug-Funktionen

- **Debug-Modus starten** : startet die WinStudio-Anwendung im Debug-Modus
- **Haltepunkte im Viewer-Task aktivieren/deaktivieren** : aktivieren Sie diese Option, wenn Sie Skripte debuggen möchten, die im Viewer-Task ausgeführt werden. Dazu zählen
  - Grafik-Skript

  - Bild-Skripte
  - Kommando-Skripte (z.B. auf einer Schaltfläche)

- **Haltepunkte im Hintergrund-Task aktivieren/deaktivieren**  : Aktivieren Sie diese Option, wenn Sie Skripte debuggen möchten, die im Hintergrund-Task ausgeführt werden. Dazu zählen
  - Start-Skript
  - Hintergrund-Skripte

- **Haltepunkt setzen/entfernen**  : fügt einen Haltepunkt in das aktuell geöffnete Skript an der aktuellen Cursor-Position ein oder entfernt ihn

---



Haltepunkte sind nicht an jeder Stelle von Skripten zulässig (z.B. Leerzeilen, Kommentarzeilen oder Zeilen mit Variablendeklarationen). Wenn Sie sich mit dem Cursor in einer dieser Zeilen befinden und einen Haltepunkt setzen, wird dieser automatisch auf die nächste zulässige Zeile verschoben.

- **Ausführung unterbrechen**  : hält die Ausführung an und wechselt in den Einzelschrittmodus
- **Ausführung fortsetzen**  : verlässt den Einzelschrittmodus und setzt die Ausführung fort
- **Einspringen**  : wechselt in die Unterfunktion, wenn sich der Cursor im Einzelschrittmodus auf einer Zeile mit Funktionsaufruf befindet
- **Nächstes Kommando**  : führt das nächste Kommando aus
- **Zurückspringen**  : springt von einer Unterfunktion zum übergeordneten Funktionsaufruf zurück
- **Debug-Modus beenden**  : beendet den Debug-Modus

5.2.8 Symbolleiste WinStudio-Dynamikfunktionen

Mit Hilfe der Symbolleiste "WinStudio-Dynamikfunktionen" können Sie Eigenschaften von Bildobjekten dynamisch verändern, d.h. animieren.

Wählen Sie dazu ein Bildobjekt aus, indem Sie es anklicken, und drücken Sie eine oder mehrere Schaltflächen der Symbolleiste. Bei erneutem Drücken entfernen Sie die dynamische Eigenschaft wieder.



Abb.5-9: Symbolleiste WinStudio-Dynamikfunktionen

- **Kommando**  : Fügt Schnittstellen für Kommandos hinzu, die beim Anklicken des Bildobjekts ausgeführt werden sollen.
- **Hyperlink**  : Fügt die Eigenschaft "Hyperlink" hinzu. Sie können anschließend eine Verknüpfung (URL) konfigurieren, die beim Anklicken des Bildobjekts geöffnet wird.

- **Balkendiagramm** : Fügt die Eigenschaft "Balkendiagramm" für gefüllte Rechtecke, Ellipsen und Polygone hinzu.
- **Textein-/ausgabe** : Fügt die Eigenschaft "Textein-/ausgabe" hinzu, so dass Sie in einem Textobjekt den angezeigten Text dynamisch ändern können.
- **Farbumschlag** : Fügt die Eigenschaft "Farbumschlag" hinzu, so dass Sie die Hintergrundfarbe in Abhängigkeit eines Variablenwerts ändern können.
- **Position** : Fügt die Eigenschaft "Position" hinzu, so dass Sie die Position des Bildobjekts in Abhängigkeit eines Variablenwerts ändern können.
- **Größenänderung** : Fügt die Eigenschaft "Größenänderung" hinzu, so dass Sie die Größe des Bildobjekts in Abhängigkeit eines Variablenwerts ändern können.
- **Linienrotation** : Fügt die Eigenschaft "Linienrotation" zu Linien und Polygonen hinzu, so dass Sie das Objekt in Abhängigkeit eines Variablenwerts drehen können.

5.3 Projektexplorer

5.3.1 Projektexplorer, Allgemeines

Der "Projektexplorer" gliedert alle Bilder, Arbeitsblätter und sonstigen Elemente, aus denen Ihr Projekt besteht und zeigt sie in einer Baumstruktur an.

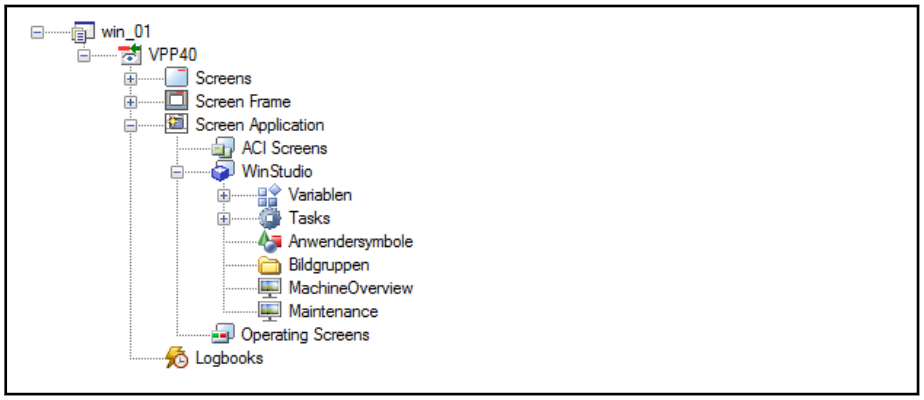


Abb.5-10: Die Ordnerstruktur von WinStudio

Klicken Sie auf das Symbol Erweitern ▷ oder führen Sie einen Doppelklick auf den jeweiligen Ordner aus, um dessen Inhalt anzuzeigen.

Klicken Sie das Symbol Reduzieren ◢ oder führen Sie einen Doppelklick auf den jeweiligen Ordner aus, um diesen zu schließen.

Alle WinStudio-Elemente im Projektbaum verfügen über ein Kontextmenü. Dieses können Sie durch Anklicken mit der rechten Maustaste öffnen. Im Kontextmenü finden Sie allgemein anwendbare Funktionen sowie Funktionen, die nur für den jeweiligen Elementtyp verfügbar sind.

Die einzelnen Bereiche des WinStudio-Projektbaums fassen verschiedene WInStudio-Funktionalitäten zusammen.

Entwicklungsumgebung

5.3.2 Ordner "WinStudio"

Der Ordner **WinStudio** enthält alle Bilder, Bildgruppen und Symbole in Ihrem Projekt.

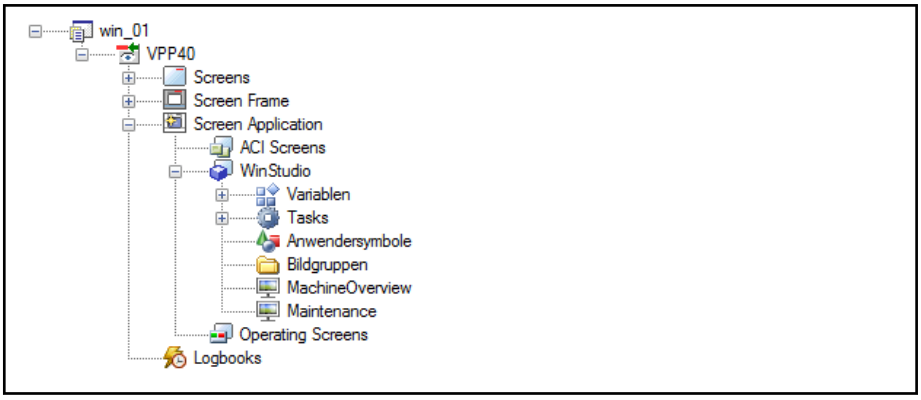


Abb.5-11: Der Ordner "WinStudio"

Die Ordner der Registerkarte "Grafik" werden auf den folgenden Seiten beschrieben:

- **WinStudio** enthält alle Bilder auf Seite 170, die für das aktuelle Projekt erstellt wurden.
- **Variablen** enthält alle Funktionen und Editoren zum Bearbeiten von Variablentypen sowie zur Konfiguration der Prozessanbindung von Variablen mit einem externen Gerät.
- **Tasks** enthält verschiedene Editoren und Funktionen, die Sie für die Erstellung komplexer Anwendungen benötigen ("Mathematikblätter", "Alarme", etc.)
- **Bildgruppen** enthält alle Bildgruppen auf Seite 175 (Einzelne WinStudio-Bilder, die zu handhabbaren Gruppen zusammengefasst sind), die für das aktuelle Projekt angelegt sind.
- **Anwendersymbole** enthält alle benutzerdefinierten Symbole auf Seite 220, wobei es sich um Gruppen von Bildern und/oder Text handeln kann. Sie können für das Projekt kundenspezifische Symbole anlegen und sie in diesem Ordner speichern.
- **Grafikskript** enthält vordefinierte Funktionen, die beim Auftreten bestimmter Bildaktionen ausgeführt werden auf Seite 800, z.B. wenn der Thin Client auf einer Remote-Station gestartet wird.

5.3.3 Ordner "Variablen"

Im Ordner **Variablen** finden Sie alle Funktionen und Editoren, die Sie zur Verwaltung von WinStudio-Tags, zur Erstellung eigener Datenstrukturen ("Klassen") sowie zur Kommunikationsanbindung an eine Steuerung mittels eines der verschiedenen zur Verfügung gestellten Treiber.

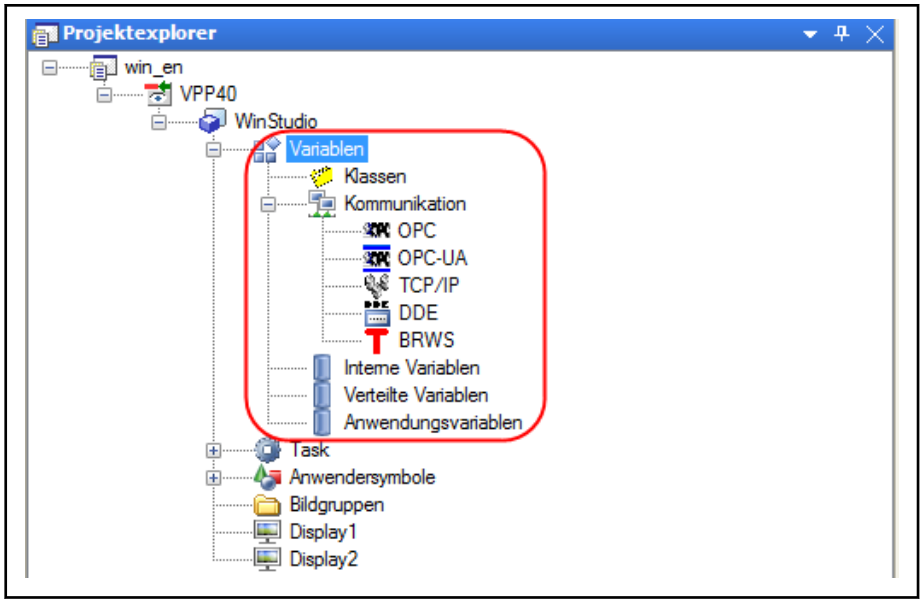


Abb.5-12: Ordner "Variablen"

In diesem Ordner finden Sie im einzelnen folgende Elemente:

- Klassen auf Seite 118 enthält komplexe Tags – "Klassen"-Tags genannt – die zum Verknüpfen einer Wertegruppe (im Gegensatz zu einem Einzelwert) mit einem Objekt erstellt wurden
- System-Tags auf Seite 121 enthält vordefinierte Tags mit speziellen Funktionen, die vom Projekt für konkrete Überwachungsaufgaben verwendet werden (zum Beispiel enthält der Tag "Date" das aktuelle Datum im Zeichenkettenformat). Alle internen Tags sind "schreibgeschützt"; dies bedeutet, dass Sie diese Tags nicht zur Datenbank hinzufügen, ändern oder aus der Datenbank entfernen können
- Gemeinsame Datenbank auf Seite 120 enthält Tags, die automatisch für die Prozessanbindung generiert wurden
- Projekt-Tags auf Seite 116 enthält während der Projektentwicklung angelegte Tags (wie z.B. Bild-Tags oder Tags, die von angeschlossenen Geräten lesen oder dorthin schreiben)
- Prozeduren auf Seite 799 enthält VBScript-Funktionen und Unterprogramme, die von jedem beliebigen anderen Skript im Projekt aufgerufen werden können
- Ereignis-Protokollierung auf Seite 285 enthält Möglichkeiten zur Protokollierung und zum Auslesen von Ereignissen
- Übersetzung auf Seite 437 enthält das Übersetzungs-Arbeitsblatt, das festlegt, wie die Benutzeroberfläche Ihres Projekts in eine andere Sprache übersetzt werden soll

5.3.4 Ordner "Tasks"

Der Ordner **Tasks** organisiert die Arbeitsblätter, die während der Projektlaufzeit als Hintergrund-Tasks ausgeführt werden.

Entwicklungsumgebung

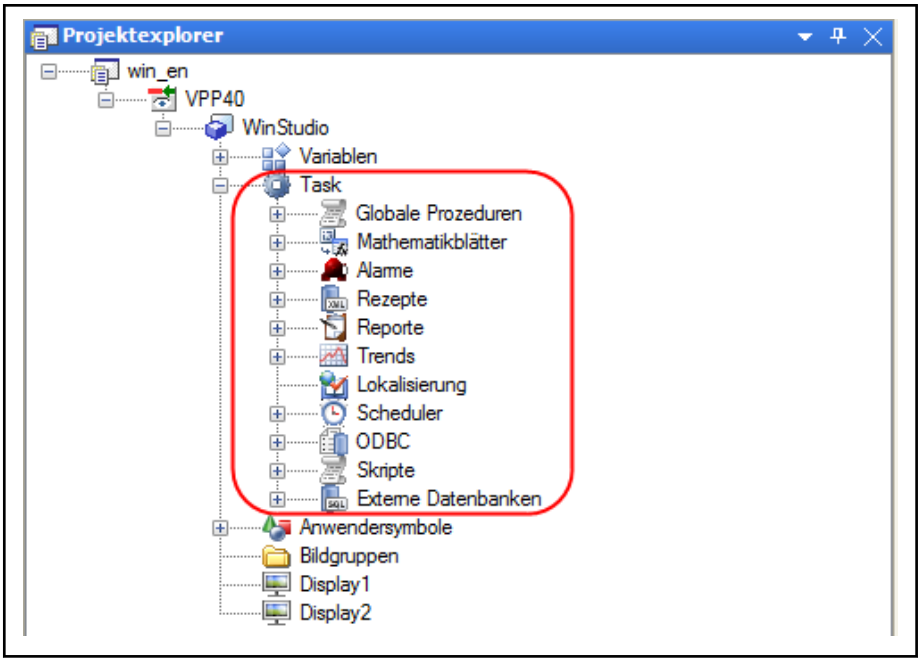


Abb.5-13: Ordner "Tasks"

Die Ordner der Registerkarte Tasks werden auf den folgenden Seiten beschrieben:

- **Mathematik** enthält die Berechnungs-Arbeitsblätter auf Seite 367 zum Konfigurieren und Erstellen von zusätzlichen Routinen, die mit verschiedenen Tasks zusammenarbeiten. WinStudio führt "Mathematik"-Arbeitsblätter als Hintergrund-Tasks während der Laufzeit aus. Sie können "Mathematik"-Arbeitsblätter konfigurieren, um Abläufe und mathematische Berechnungen für Ihr Projekt zur Verfügung zu stellen
- **Alarme** enthält Alarm-Arbeitsblätter auf Seite 273, in denen Sie Alarmmeldungen festlegen können, die zur Laufzeit angezeigt werden, wenn ein zugeordneter Tag einen bestimmten Wert annimmt oder überschreitet. Die zur Laufzeit anstehenden Alarmmeldungen können Sie automatisch in einer Datei protokollieren und mit Hilfe des Alarm-Controls in einem WinStudio-Bild anzeigen.
- **Rezepte** enthält die Rezeptur-Arbeitsblätter auf Seite 361, mit denen Sie Variablen konfigurieren können, deren Daten zur Laufzeit in eine ASCII- oder XML-Datei geschrieben oder aus einer Datei gelesen und in die Projektdatenbank übernommen werden.
- **Reporte** enthält die Bericht-Arbeitsblätter auf Seite 363, mit denen Sie Textdokumente erstellen können, die zur Laufzeit generiert und an einen Drucker oder eine Datei gesendet werden. In den Reporten können Sie nicht nur festen Texte, sondern auch aktuelle Werte von Variablen ausgeben lassen. Diese Funktion ist besonders zur Dokumentation von Systemdaten hilfreich.
- **Trend** enthält die Trend-Arbeitsblätter auf Seite 301.Sie können mit Hilfe von Trend-Arbeitsblättern zeitliche Verläufe von Variablenwerten aufzeichnen. Diese Wertefolgen können Sie verwenden, um diese in einem WinStudio-Bild mit dem Trend-Control anzuzeigen. In Trend-Arbeitsblättern können Sie diejenigen Variablen deklarieren, die in einer sogenannten binären Historiendatei (***.hst**) gespeichert werden sollen
- **Lokalisierung** öffnet den Übersetzungseditor. Hier können Sie alle Texte, die Sie in Ihrer Anwendung verwenden, in andere Sprachen überset-

zen. Zur Laufzeit können Sie dann die Texte zwischen den verschiedenen Sprachen umschalten.

- **Scheduler** enthält die Scheduler-Arbeitsblätter auf Seite 370, mit denen Sie beim Auftreten verschiedener Ereignisse eine Funktion ausführen können. Bei dem Ereignis kann es sich um eine Werteänderung einer Variablen, das Erreichen eines Wertes oder um eine bestimmte Uhrzeit bzw Datum handeln.
- **ODBC** enthält die ODBC-Arbeitsblätter auf Seite 364, mit denen die ODBC-Schnittstelle in einer Netzwerkumgebung konfigurieren können. Mit Hilfe von ODBC-Arbeitsblättern können Sie Daten zwischen Ihrem Projekt und einer beliebigen Datenbank austauschen, die die ODBC-Schnittstelle unterstützt.
- **Skripte** enthält das Startup-Skript auf Seite 369 und das **Grafik-Skript**. Skriptgruppen auf Seite 368. Diese vordefinierten Skripte können Sie verwenden, um beim Start der Anwendung oder während der Ausführung des Viewers Funktionen auszuführen. Außerdem können Sie in diesem Ordner beliebige eigene Skripte erstellen, mit denen Sie Abläufe in VB-Skript programmieren können.
- **Globale Prozeduren** enthält standardmäßig das Element **Hauptprozeduren**. In dieser Datei können Sie eigenen Funktionen deklarieren, die Sie in einem Skript aufrufen können. Zur Strukturierung Ihrer Funktionen können Sie weitere Skripte hinzufügen, in denen Sie weitere Funktionen deklarieren können.
- **Externe Datenbanken** enthält Datenbank-Arbeitsblätter auf Seite 371, die mit externen Datenbanken mittels der Standard-ADO.NET-Schnittstelle (als Alternative zu ODBC) kommunizieren.

5.3.5 Ordner Kommunikation

Im Ordner **Kommunikation** befinden sich weitere Unterordner, in denen Sie Arbeitsblätter erstellen, mit denen die Kommunikation mit anderen Geräten oder Software über verfügbare Protokolle aufgebaut wird.

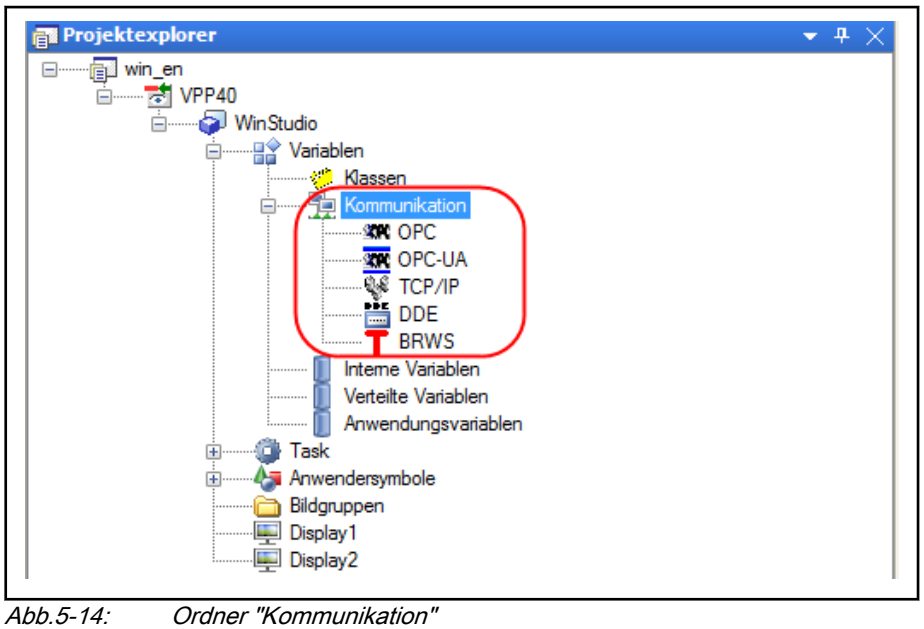


Abb.5-14: Ordner "Kommunikation"

Der Ordner Kommunikation die folgenden Unterordner und Funktionen.

- **OPC** enthält Arbeitsblätter OPC-Arbeitsblätter auf Seite 390, die Sie für die Kommunikationen zwischen Ihrem Projekt (das als OPC-Client fun-

giert) und einem anderen Gerät, in dem der klassische OPC-Standard (OPC-DA) implementiert ist, erstellen.

- **OPC UA** enthält Arbeitsblätter OPC-UA-Arbeitsblätter auf Seite 395, die zum Herstellen einer Verbindung zu OPC-Servern mit Hilfe des neuen Protokolls „OPC Unified Architecture“ verwendet werden.
- **TCP/IP** enthält Arbeitsblätter TCP/IP-Arbeitsblätter auf Seite 401, mit denen Sie TCP/IP-Kommunikation zu anderen "WinStudio"-Stationen konfigurieren können.

  Mit Hilfe von "WinStudio"-TCP/IP-Client- und -Server-Modulen können Sie die Datenbanken von zwei oder mehr Projekten synchronisieren.
- **DDE** enthält Arbeitsblätter DDE-Arbeitsblätter auf Seite 403, mit denen WinStudio Daten per DDE-Protokoll (Dynamic Data Exchange) mit einer Windows-Anwendung (z.B. Microsoft Excel oder ein anderes Windows-Programm, das diese Schnittstelle unterstützt) austauscht. Bei einer DDE-Kommunikation handelt es sich um eine Interaktion zwischen Server- und Client-Programmen. "WinStudio" stellt Schnittstellen zur Verfügung, die als Client oder als Server ausgeführt werden können.
- **BR_WS** enthält Arbeitsblätter Treiber-Arbeitsblätter auf Seite 379, mit denen WinStudio Daten über die BR_WS-Kommunikationsschnittstelle mit einer SPS-Steuerung austauscht. Bei einem "Kommunikationstreiber" handelt es sich um eine **.DLL**-Datei, die bestimmte Information über die Remote-Geräte enthält und das Kommunikationsprotokoll implementiert.
- **Weitere Treiber:** falls Sie Daten mit einem Gerät austauschen möchten, das keines der oben geschriebenen Protokolle unterstützt, können Sie einen entsprechenden Treiber hinzufügen. Wenn Sie für das Gerät ein Treiberdatei (.DLL) besitzen, kopieren Sie diese in das Verzeichnis ..\WinStudio\Drv. Anschließend können Sie über das Kontextmenü des Ordners "Kommunikation" über den Menüpunkt **Treiber einfügen/entfernen** den Treiberkonfigurationsdialog öffnen. Dieser listet alle verfügbaren Treiber auf. Nach dem Hinzufügen eines Treibers wird ein entsprechender Unterordner erstellt, in dem Sie die zugehörigen Arbeitsblätter erstellen können.

5.4 Anwendung konfigurieren und testen

5.4.1 Anwendung verifizieren

Wählen Sie **Anwendung verifizieren** aus, um alle Berechnungs-Arbeitsblätter auf Seite 367, alle Ausdrücke, die Bildlogik erneut zu kompilieren und die HTML-Dateien zu aktualisieren, unter Verwendung der Einstellungen in der Registerkarte Web des Dialogs Projekteinstellungen auf Seite 86.

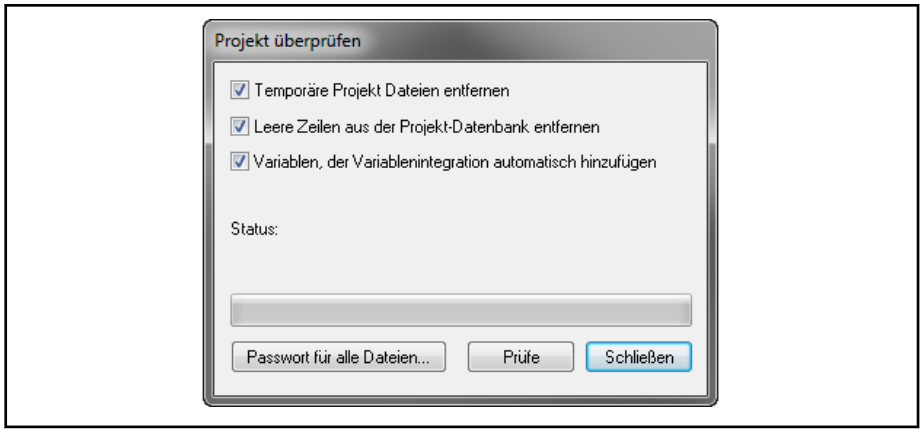


Abb.5-15: Anwendung verifizieren

Der Dialog erlaubt ferner die Neuvergabe eines Passworts für aller Dateien:

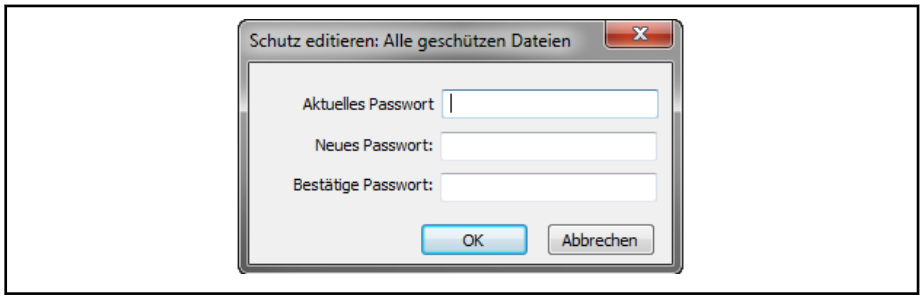


Abb.5-16:

Wenn in der Anwendung Variablen verwendet werden, die in der Datenbank nicht definiert sind, zeigt eine Fehlermeldung, wo (d.h. in welchem Bild bzw. Arbeitsblatt) diese Tags verwendet werden.

Beim Speichern von Bildern oder Arbeitsblättern fügt WinStudio einen Zeiger auf die aktuelle Datenbankversion hinzu. Beim Ausführen Ihres Projekts vergleicht WinStudio die Datenbankversion mit der aktuellen Projektdatenbank und kompiliert die Ausdrücke bei Nichtübereinstimmung der Datenbank neu.

Um zu vermeiden, dass diese Kompilierung während der Laufzeit des Projekts erfolgt, empfehlen wir, das Werkzeug **Anwendung verifizieren** vor dem Herunterladen/Abschließen eines Projekts auszuführen.

Diese Funktion sollte ebenfalls eingesetzt werden, wenn ein Projekt für eine neue WinStudio-Version konvertiert wird.

Wenn Sie **Anwendung verifizieren** auswählen, löscht dies alle temporären Dateien (z.B. die Dateien ***.txt**, ***.mac** und ***.tag**) aus dem Projektordner, um Festplattenplatz zu sparen und bereinigt das Projekt, bevor es auf die Laufzeitstation geladen wird.

Hierbei lädt WinStudio das Projekt jedoch "nicht" automatisch auf die Laufzeitstation hoch. Zum Übertragen des Projekts auf die Laufzeitstation muss das Werkzeug **Projekt zum Ziel senden** verwendet werden.

5.4.2 Anwendung starten

Klicken Sie im Kontextmenü des Ordners "WinStudio" auf **Anwendung starten**, um die im Dialog <Ausführungs-Tasks> (Registerkarte "Ausführungs-Tasks") als **Automatisch** gekennzeichneten Laufzeitmodule auszuführen.

Wenn Sie das Modul "Viewer" starten, öffnet dies die zurzeit bearbeitete(n) Bild(er).

Wenn Sie keine "automatischen" Tasks angeben, startet "WinStudio" den "Viewer" und die "Hintergrund"-Tasks automatisch, wenn Sie **Anwendung starten** anklicken.

Wenn Sie aktuell keine Bilder im Editor geöffnet haben, öffnet das Modul "Viewer" das eingestellte Startbild (siehe Feld **Startbild** auf der Registerkarte "Viewer" im Dialog Projekteinstellungen auf Seite 67)

Anwendung starten bezieht sich auf das Projekt auf dem Zielsystem (im Dialog Remote-Verwaltung auf Seite 66 konfiguriert). Vergewissern Sie sich, welches Zielsystem konfiguriert ist (lokal oder fern), bevor Sie **Anwendung starten** anklicken.

5.4.3 Anwendung beenden

Klicken Sie im Kontextmenü des Ordners "WinStudio" auf **Anwendung beenden**, um alle Laufzeit-Tasks anzuhalten.

Anwendung beenden betrifft das Projekt des ausgewählten Zielsystems (siehe Dialog Remote-Verwaltung auf Seite 66). Vergewissern Sie sich, welches Zielsystem konfiguriert ist (lokal oder fern), bevor Sie **Anwendung beenden** anklicken.

5.4.4 WinStudio-Zielplattform

Wählen Sie diese Funktion aus, um den Dialog "Fern-Management" zu öffnen, mit dem Sie Remote-Stationen verwaltet können (Dateien hoch- und herunterladen und Remote-Projekte ausführen bzw. anhalten).

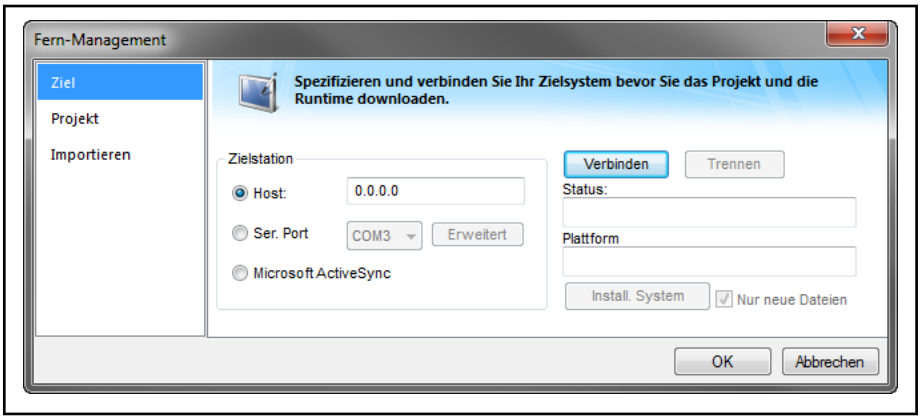


Abb.5-17: Dialog Fern-Management

Dieser Dialog enthält die folgenden Registerkarten:

- **Ziel**: Mit dieser Registerkarte geben Sie das Zielsystem an; dies umfasst:
  - **Netz-IP**, **Serielle Schnittstelle** oder **Microsoft ActiveSync** (aktuell funktioniert Microsoft ActiveSync nur mit Windows Embedded Compact sowie Windows Mobile 5.0 und höheren Versionen.)
  - **Verbinden**/ **Trennen** mit/vom Zielsystem
  - **Status** der Station überprüfen
  - Nach dem Verbinden Wiedergabe der **Plattform** (Betriebssystem + Prozessorart) des Zielsystems.
  - **Systemdateien installieren** (oder **Nur neuere Dateien**)
- **Projekt**: Mit dieser Registerkarte geben Sie das Projekt an; dies umfasst:

  - den **lokalen** und den **Ziel-** Verzeichnispfad
  - Angabe, ob das Projekt **zum Zielsystem** zu **senden** ist
    - **Nur neuere Dateien**: Wenn dieses Option aktiviert ist, werden nur die neueren Anwendungsdateien zum Gerät gesendet.
    - **Benutzerdateien beibehalten**: Wenn dieses Option aktiviert ist, werden alle ggf. vorhandenen Benutzerdateien in Ihrem Projektordner beibehalten.
  - **Datei senden**
  - Projekt **Ausführen/Anhalten**
  - **Status** des Projekts überprüfen
- **Importieren**: Mit dieser Registerkarte importieren Sie ein Projekt; dies umfasst:
  - Angabe von **Ziel**- und **Quell**-Projektpfad
  - **Vom Ziel holen**
  - **Protokolldatei holen**
  - **Status** des Projekts überprüfen

Nähere Informationen zum Verbinden mit Windows-Embedded-Geräten finden Sie unter Installation von CEView auf Ihrem Windows-Embedded-Gerät auf Seite 41.

5.4.5 Projekteinstellungen

Registerkarte Informationen

Wählen Sie den Eintrag **Eigenschaften** im Kontextmenü des Ordners "WinStudio" aus, um den Dialog "Projekteinstellungen" zu öffnen.

Dieser steuert Einstellungen, die das Gesamtprojekt betreffen.

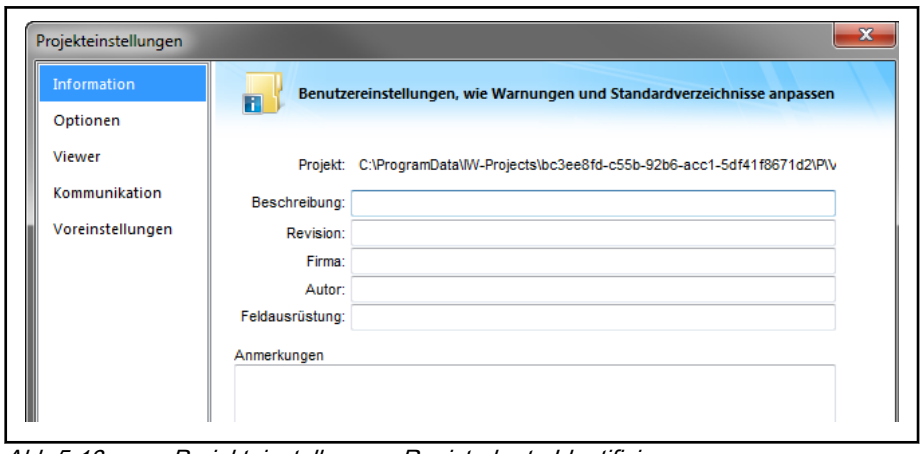


Abb.5-18: Projekteinstellungen: Registerkarte Identifizierung

Mithilfe der folgenden Parameter identifizieren Sie das Projekt (nur zu Dokumentationszwecken).

- **Beschreibung**
- **Revision**
- **Firma**
- **Autor**
- **Feldausrüstung**
- **Anmerkungen**

Entwicklungsumgebung

Registerkarte Optionen

Registerkarte Optionen, Allgemeines

Mit Hilfe dieser Registerkarte geben Sie Parameter an, die sich auf Ihr Projekt im Allgemeinen beziehen.

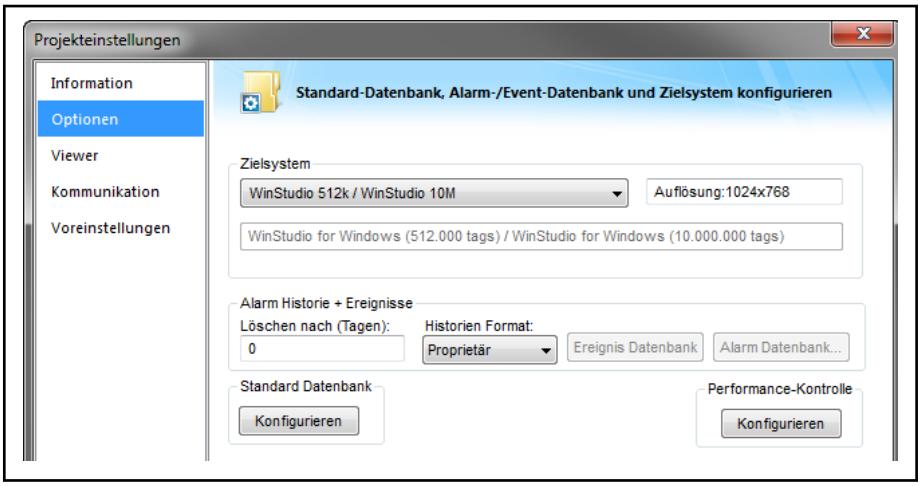


Abb.5-19: Projekteinstellungen: Registerkarte Optionen

Im Folgenden wird eine Beschreibung dieser Parameter gegeben:

- **Zielsystem**: Mit Hilfe der Auswahlliste geben Sie das Zielsystem des aktuellen Projekts an. Das Zielsystem legt die Beschränkungen des Projekts (z.B. die unterstützte Anzahl an Tags) fest und muss zu Ihrer Lizenz passen. Die Beschreibung der wichtigsten Lizenzbeschränkungen für jedes Zielsystem wird unterhalb der Auswahlliste angezeigt, in der Sie dieses festlegen.

HINWEIS	**Wenn Sie eine Zielsystem-Ebene angeben, die nicht der aktuellen Lizenzebene auf dem Zielsystem entspricht, wird Ihr Projekt möglicherweise nicht ordnungsgemäß ausgeführt.**

- **Auflösung**: Zeigt die Bildauflösung Ihres Projekts an.
- **Alarm-Historie und Ereignisse**: Geben Sie einen Wert in das Feld „Maximaldauer der Historie (Tage)“ ein, um anzugeben, wie lange die Dateien mit der Alarm- und Ereignis-Historie vorgehalten werden sollen. Nach der festgelegten Anzahl an Tagen löscht das Projekt automatisch bestehende Alarm-/Ereignis-Historie-Dateien, die älter als die angegebene Zeit sind. Wenn Sie in dieses Feld eine Null eintragen, löscht das Projekt keine Historien-Dateien automatisch. In einem solchen Fall sollten Sie eine externe Prozedur zum Löschen der alten Historien-Dateien anlegen, sonst geht auf Dauer der freie Speicherplatz im Rechner zur Neige.
- **Historienformat**: Wählen Sie das Format der Alarm/Ereignis-Historie wie folgt aus:

Format	Beschreibung
Proprietär	Speichert die Historien-Daten im Unterordner "Alarm" Ihres Projektordners (Standard), und zwar in Textdateien unter Verwendung des proprietären Formats.
Datenbank	Speichert die Historien-Daten in der benutzerdefinierten relationalen SQL-Datenbank mit Hilfe der eingebauten ADO-Schnittstelle.
Binär	Speichert die Historien-Daten im Unterordner "Alarm" Ihres Projektordners (Standard), und zwar in Binärdateien unter Verwendung des proprietären Formats.

Weitere Informationen finden Sie unter Speichern von Alarm-Historie/Ereignisprotokoll in einer externen Datenbank auf Seite 281.

- **Standarddatenbank**: Hiermit können Sie eine Standarddatenbank konfigurieren, die von verschiedenen Tasks und Objekten gemeinsam verwendet wird. Weitere Informationen erhalten Sie unter Konfigurieren einer Standarddatenbank für die gesamte Task-Historie auf Seite 508.

Die PC-basierte Steuerung verfügt über eine eigene angepasste Schnittstelle, bei der Sie Informationen über die PC-basierte Steueranwendung angeben müssen, damit Tags mit dem "WinStudio"-Projekt gemeinsam verwendet werden können.

Alarm-Historie in einer externen Datenbank speichern

Standardmäßig werden die Alarm-Historie und das Ereignisprotokoll in Textdateien eines proprietären Formats im Ordner Alarme Ihres Projekts gespeichert. Sie können jedoch die Projekteinstellungen so abändern, dass diese statt dessen in einer externen SQL-Datenbank gespeichert werden.

1. Öffnen Sie die Registerkarte **Optionen** des Dialogs "Projekteinstellungen".

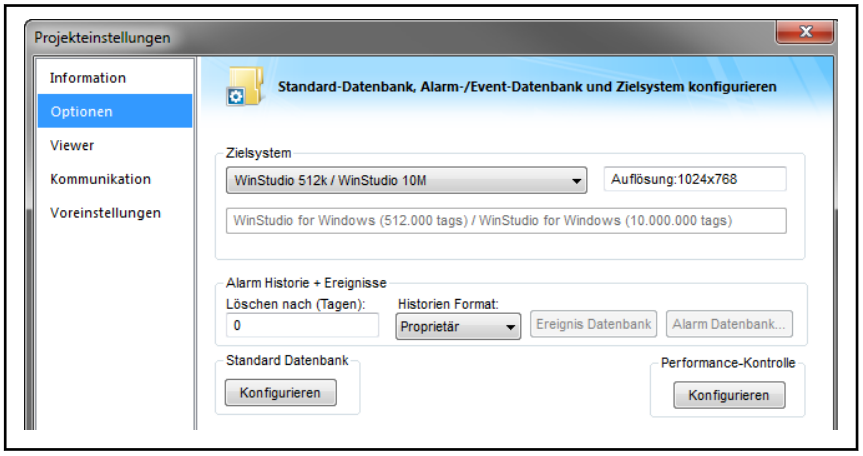


Abb.5-20: Projekteinstellungen: Optionen

2. Im Bereich **Alarm-Historie und Ereignisse** im Eingabefeld **Löschen nach (Tagen)** tragen Sie die Anzahl Tage der Historie ein, die Sie speichern wollen. In dem Maß, wie die Historie die festgelegte Anzahl an Tagen überschreitet, wird sie automatisch gelöscht, wobei die ältesten Einträge zuerst gelöscht werden. Wenn keine Zahl angegeben ist – wenn also das Feld leer gelassen oder auf 0 gesetzt wird – wird die Historie nie gelöscht. Wie viel Historie Sie speichern können, ist nicht beschränkt, aber je mehr Sie speichern, desto mehr Festplattenspeicher wird belegt.
3. Wählen Sie aus der Liste **Historien-Format** den Punkt **Datenbank** aus.

4. Um eine einzelne Standarddatenbank zu konfigurieren, die sowohl für die Alarm-Historie als auch das Ereignisprotokoll (und auch alle anderen Laufzeit-Tasks) verwendet werden soll, klicken Sie im Bereich **Standarddatenbank** auf den Eintrag **Konfigurieren**. Der Dialog "Konfiguration der Standarddatenbank" wird angezeigt. Konfigurieren Sie mithilfe dieses Dialogs die Datenbankverbindung. Nähere Informationen finden Sie unter Konfigurieren einer Standarddatenbank für alle Task-Historien auf Seite 508.
5. Um eine separate Datenbank für Ihr Ereignisprotokoll oder auch Ihre Alarm-Historie zu konfigurieren, klicken Sie jeweils auf **Ereignis-Datenbank** bzw. **Alarm-Datenbank**. In beiden Fällen wird der Dialog "Konfiguration der Datenbank" angezeigt. Konfigurieren Sie mithilfe dieses Dialogs die Datenbankverbindung. Weitere Informationen erhalten Sie in Konfiguration der Datenbank auf Seite 502.
6. Klicken Sie auf **OK**.

Konfigurieren einer Standarddatenbank für alle Task-Historien

Sie können eine Standarddatenbank konfigurieren, die die Historie-Daten aller Tasks im Projekt speichert. Nachdem Sie eine Standarddatenbank konfiguriert haben, können Sie immer, wenn Sie ein neues Task-Arbeitsblatt anlegen, wählen, ob Sie die Standarddatenbank verwenden oder für dieses spezielle Arbeitsblatt eine neue Datenbank konfigurieren möchten.

Konfigurieren der Verbindungseinstellungen für die Standarddatenbank:

1. Öffnen Sie den Eigenschaftendialog des Ordners "WinStudio" und wählen Sie das Register "Optionen" an
2. Klicken Sie im Bereich "Standard Datenbank" auf **Konfigurieren**. Der Dialog "Konfiguration der Standarddatenbank" wird angezeigt

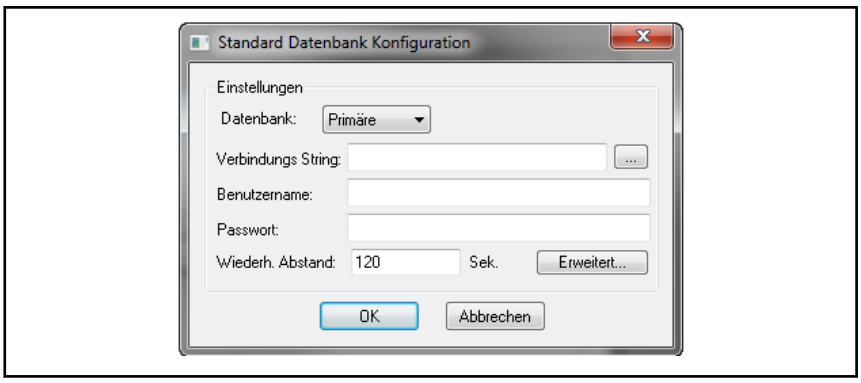


Abb.5-21: Dialog Konfiguration der Standarddatenbank

Hilfestellung zum Ausfüllen der Felder in diesem Fenster finden Sie unter Dialog Datenbankkonfiguration auf Seite 502.

Datenbankkonfiguration

Datenbankverbindung konfigurieren

Der Austausch von Daten mit einer externen Datenbankdatei erfolgt analog zu den Gerätetreibern. Zunächst müssen Sie eine neue Datenbankverbindung konfigurieren. Anschließend können Sie Arbeitsblätter erstellen, in denen Sie die zuvor erstellte Datenbankverbindung auswählen, und WinStudio-Tags mit einzelnen Datensätzen der externen Datenbank verknüpfen können.

Um eine neue Datenbankverbindung zu konfigurieren, öffnen Sie das Kontextmenü des Ordners **WinStudio ▸ Tasks ▸ Externe Datenbanken** und wählen den Eintrag "Datenbankverbindung konfigurieren" aus.

Der Ordner "Externe Datenbanken" ist nur verfügbar, wenn Sie in den WinStudio-Eigenschaften als Zielsystem "WinStudio 1.5k" oder höher auswählen und die Geräteversion in den VPx-/VEx-Eigenschaften "13V06" oder höher ist.

Im Übersichtsdialog können Sie mit der Schaltfläche "Hinzufügen" den Konfigurationsdialog für eine neue Datenbankverbindung öffnen. Geben Sie dort im Feld "Name" einen beliebigen Namen ein, den Sie später in den Arbeitsblättern auswählen können.

Mit Hilfe des Dialogs "Datenbankkonfiguration" können Sie die erforderlichen Einstellungen zum Anbinden von "WinStudio" an eine externe Datenbankdatei konfigurieren.

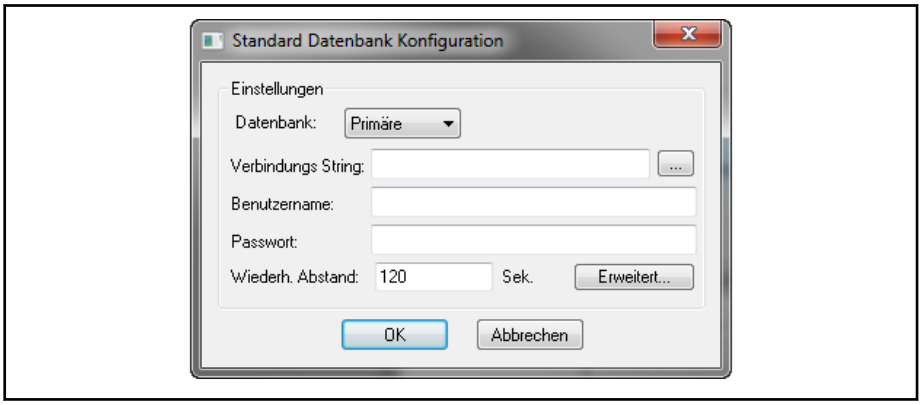


Abb.5-22: Dialog Datenbankkonfiguration

- Combobox **Datenbank**: Zur Auswahl von Primär oder Sekundär. Bei der Auswahl "Primär" betreffen alle Einstellungen der Datenbankkonfiguration die Primärdatenbank-Schnittstelle. Ansonsten betreffen sie die Sekundärdatenbank-Schnittstelle. Die Sekundärdatenbank können Sie in folgenden Modi konfigurieren:
  - **Deaktiviert**: In diesem Modus speichert "WinStudio" Daten nur in der Primärdatenbank. Wenn die Primärdatenbank aus irgendeinem Grund nicht zur Verfügung steht, werden die Daten nicht anderweitig gespeichert. Bei dieser Option können Daten verloren gehen, wenn die Primärdatenbank nicht verfügbar ist.
  - **Redundant**: In diesem Modus speichert "WinStudio" Daten in der Primär- und auch in der Sekundärdatenbank. Wenn eine dieser Datenbanken nicht zur Verfügung steht, speichert "WinStudio" die Daten weiterhin, und zwar nur in der zur Verfügung stehenden Datenbank. Wenn die nicht verfügbare Datenbank wieder verfügbar wird, synchronisiert "WinStudio" beide Datenbanken automatisch.
  - **Speichern und Weiterleiten**: In diesem Modus speichert "WinStudio" Daten nur in der Primärdatenbank. Wenn die Primärdatenbank nicht mehr zur Verfügung steht, speichert "WinStudio" die Daten in der Sekundärdatenbank. Wenn die Primärdatenbank wieder zur Verfügung steht, verschiebt "WinStudio" die Daten aus der Sekundär- in die Primärdatenbank.

Bei der Primär- und Sekundärdatenbank kann es sich um unterschiedliche Datenbanktypen handeln. Sie müssen jedoch die gleichen Felder enthalten.

Durch Verwendung der Sekundärdatenbank können Sie die Systemzuverlässigkeit erhöhen und die Sekundärdatenbank als Backup verwenden, wenn die Primärdatenbank nicht zur Verfügung steht. Diese Architektur ist besonders nützlich, wenn die Primärdatenbank in der Remote-Station angesiedelt ist. In diesem Fall können Sie eine Sekundärdatenbank in der lokalen Arbeitsstation so konfigurieren, dass die Daten vorübergehend dort gespeichert werden, wenn die Primärdatenbank (z.B. während eines Netzausfalls) nicht zur Verfügung steht.

- **Projektstandard verwenden**: Wenn diese Option angehakt ist, verwendet "WinStudio" für den gerade konfigurierten Task die Einstellungen in der Standarddatenbank (Verbindungszeichenkette, Benutzername, Passwort, Intervall für Wiederholungsversuche und Weitere Einstellungen). Wenn diese Option nicht aktiviert ist, können diese Einstellungen individuell für den derzeitigen Task konfiguriert werden.
- **Verbindungszeichenkette**: In diesem Feld wird die Datenbank festgelegt, in die "WinStudio" Werte schreibt bzw. aus der es sie liest, sowie die Parameter, die es hauptsächlich beim Verbinden mit der Datenbank einsetzt. Statt der manuellen Eingabe der Verbindungszeichenkette kann auch die Durchsuchen-Schaltfläche (...) angeklickt und der Datenbanktyp aus dem Fenster **Datenverbindungseigenschaften** ausgewählt werden.

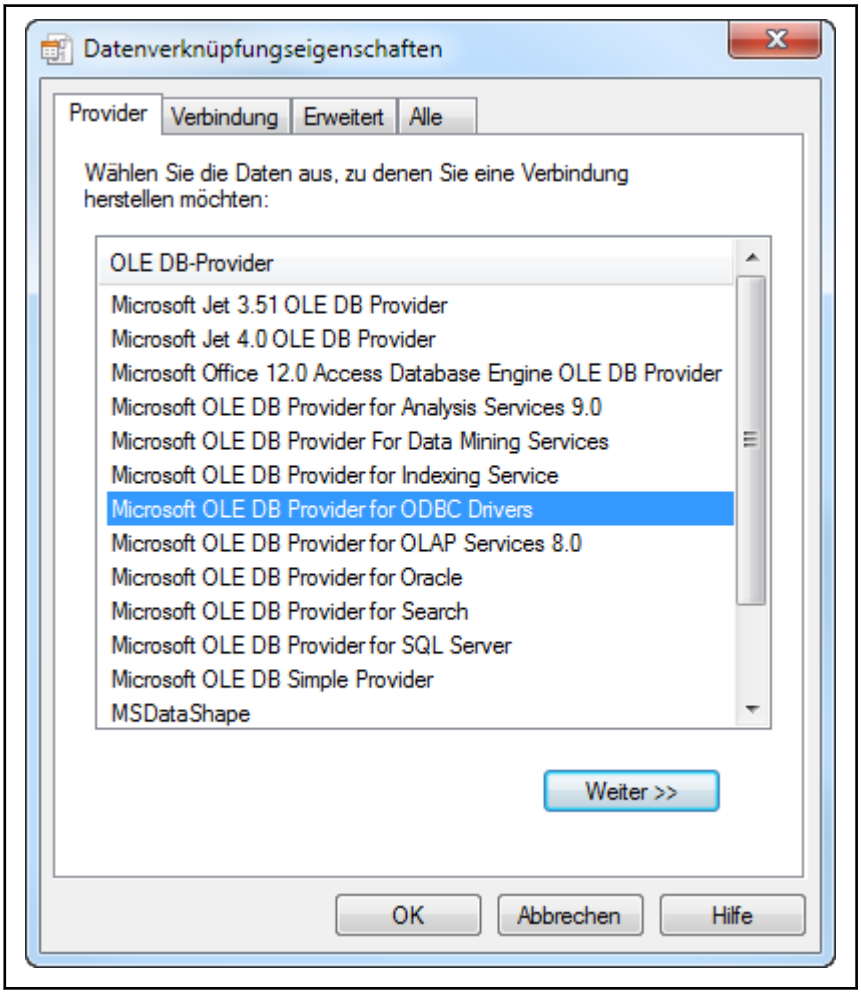


Abb.5-23: Dialog Datenverbindungseigenschaften



Die im Fenster Datenverbindungseigenschaften aufgeführte Liste der Datenbankanbieter ist abhängig von den tatsächlich auf dem Rechner mit "WinStudio" installierten und verfügbaren Datenbankanbietern. Konsultieren Sie die Dokumentation des Betriebssystems (bzw. der Datenbank) für weitere Informationen über das Einstellen des Datenbankanbieters für die von Ihnen verwendete Datenbank.

- **Benutzername**: Der Benutzername, der für die Verbindung zur Datenbank verwendet wird. Der in diesem Feld eingetragene Benutzername muss zum Benutzernamen passen, der in der Datenbank konfiguriert ist.
- **Passwort**: Das Passwort, das für die Verbindung zur Datenbank verwendet wird. Das in diesem Feld eingetragene Passwort muss zum Passwort passen, das in der Datenbank konfiguriert ist.
- **Intervall für Wiederholungsversuche**: Wenn sich "WinStudio" aus irgendeinem Grund nicht mit der Datenbank verbinden kann, versucht es automatisch nach Verstreichen der in diesem Feld konfigurierten Sekunden erneut, sich mit der Datenbank zu verbinden.
- **Weitere Einstellungen**: Durch Klicken auf diese Schaltfläche erhalten Sie Zugang zur Konfiguration erweiterter Einstellungen. Für die meisten Projekte muss der Standardwert dieser Einstellungen nicht geändert werden und sollte beibehalten werden.

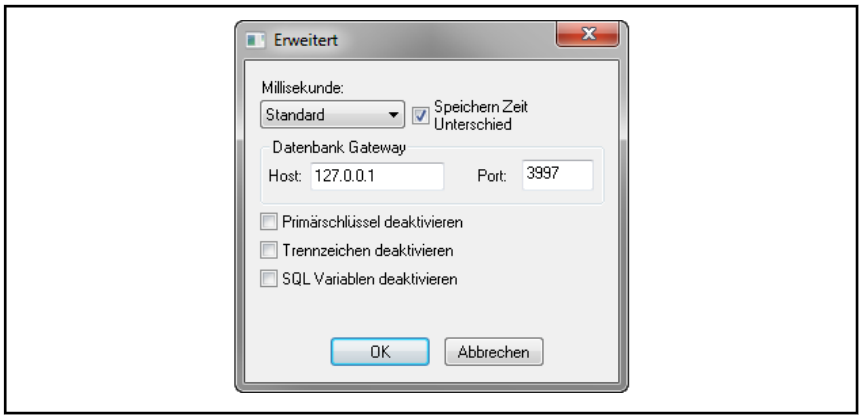


Abb.5-24: Datenbankkonfiguration: Dialog Weitere Einstellungen

– **Millisekunden**

  Hier können Sie einstellen, wie die Millisekunden beim Speichern des Datums in der Datenbank abgespeichert werden. Verschiedene Datenbanken speichern das Datum in verschiedenen Formaten. Beispielsweise unterstützen manche Datenbanken in einem **Datumsfeld** keine Millisekunden. Die folgenden Optionen sind verfügbar:

  – **Standard**: Verwendet das für die aktuelle Datenbank vordefinierte Format. Die bereits von "Bosch Rexroth" getesteten Datenbanken sind mit der Option vorkonfiguriert, die jeweils am besten passt. Bei Auswahl von "Standard" verwendet "WinStudio" die für den derzeitigen Datenbanktyp vorkonfigurierte Einstellung. Wenn Sie eine Datenbank verwenden, die nicht vorkonfiguriert ist, wird bei Auswahl der Option Standard

versucht, die Millisekunden in einem eigenen Feld abzuspeichern.



Die Standardoption für jede Datenbank ist in der Datei **StADOSvr.ini** vorkonfiguriert; diese liegt im Unterverzeichnis **\BIN** von "WinStudio". Nähere Informationen zur Konfiguration der Datei **StADOSvr.ini** finden Sie unter Studio-Datenbank-Gateway auf Seite 496.

- **Deaktivieren**: Speichert die Millisekunden gar nicht, wenn das Datum in der Datenbank abgespeichert wird.
- **Aktivieren**: Speichert die Millisekunden im gleichen Feld, in dem das Datum gespeichert wird.
- **Eigene Spalte**: Speichert die Millisekunden in einer eigenen Spalte. In diesem Fall wird das Datum (ohne Millisekunden) in einem Feld und die Anzahl der Millisekunden in einem anderen Feld gespeichert. Diese Option ist angezeigt, wenn Sie Zeitstempel in Millisekunden-Genauigkeit abspeichern wollen, die von Ihnen verwendete Datenbank für **Datumsfelder** aber keine Millisekunden unterstützt.

- **Zeitdifferenz speichern**: Wenn diese Option aktiviert ist (Standard), speichert "WinStudio" in jedem Register der Datenbank die Zeitzone des Rechners, auf dem das Projekt ausgeführt wird. Diese Option muss aktiviert sein, um Probleme mit der Sommerzeit zu umgehen.
- **Datenbank-Gateway**: Tragen Sie hier den Hostnamen/ die IP-Adresse des Netzknotens ein, auf dem das "WinStudio"-Datenbank-Gateway laufen wird. Die Nummer des TCP-Ports kann zwar auch angegeben werden, aber wenn Sie nicht den Standard verwenden, müssen Sie das "WinStudio"-Datenbank-Gateway mit dem gleichen TCP-Port konfigurieren. Nähere Informationen zur Konfiguration der weiteren Einstellungen für das "WinStudio"-ADO-Gateway finden Sie unter Studio-Datenbank-Gateway auf Seite 496.
- **Primärschlüssel deaktivieren**: Für einige Module versucht "WinStudio", zum Beschleunigen der Abfragen einen Primärschlüssel für die Tabelle zu definieren. Wenn Sie eine Datenbank verwenden, die keine Primärschlüssel unterstützt (z.B. Microsoft Excel), dann sollten Sie diese Option durch Anklicken abwählen.
- **Feldbegrenzer deaktivieren**: Wählen Sie diese Fehlerbehandlungsoption zum Deaktivieren der Feldbegrenzer aus, die zur Formatierung der Kommunikation mit der Datenbank eingesetzt werden. Feldbegrenzer können zu Problemen führen, wenn ein TrendControl bzw. ein Raster eine Abfrage aufbaut, die Gesamtsummen - wie z.B. Min und Max - beinhaltet.
- **SQL-Variablen deaktivieren**: Wählen Sie diese Fehlerbehandlungsoption zum Deaktivieren von SQL-Variablen - wie z.B. "@Value1" und "?" - aus, die in SQL-Befehlen und -Anfragen häufig eingesetzt werden. Einige bestimmte Datenbankanbieter unterstützen diese Variablen nicht.

Bereich Tabelle

In diesem Bereich können Sie die Einstellungen der Tabelle konfigurieren, in der die Daten gespeichert werden. Alle Tasks können die gleiche Datenbank verwenden. Jedoch muss jeder Task (Alarm-, Ereignis-, Trend-Arbeitsblätter) mit jeweils einer eigenen Tabelle verknüpft werden. "WinStudio" prüft dieses

Feld nicht auf ungültige Konfigurationen ab. Sie sollten daher sichergehen, dass die jeweilige Einstellung sich für die von Ihnen verwendete Datenbank eignet.

- **Standardnamen verwenden**: Wenn diese Option angehakt ist (Standard), speichert bzw. lädt "WinStudio" die Daten in die Tabelle, wobei der Standardname in das Feld **Name** geschrieben wird.
- **Automatisch anlegen**: Wenn diese Option angehakt ist (Standard), legt "WinStudio" eine Tabelle mit dem Namen, der im Feld **Name** steht, automatisch an. Wenn diese Option nicht aktiviert ist, legt "WinStudio" die Tabelle nicht automatisch an. Daher können Daten in der Datenbank nur abgespeichert werden, wenn bereits in der Datenbank eine Tabelle mit dem im Feld **Name** konfigurierten Namen manuell angelegt wurde.
- **Name**: Benennt den Namen der Tabelle aus der Datenbank, in der die Historien-Daten abgespeichert werden. Um ein Arbeitsblatt in einer Microsoft-Excel-Arbeitsblatt-Datei anzugeben, verwenden Sie folgende Syntax:

  ```
  [arbeitsblattname$]
  ```

- **Aktualisieren**: Wenn die eingestellte Datenbank aktuell verfügbar ist, können Sie die Schaltfläche **Aktualisieren** drücken und so die Combobox **Name** mit dem Namen der derzeit in der Datenbank vorhandenen Tabellen füllen. So können Sie die Tabelle, in der die Historiendaten gespeichert werden sollen, auswählen, anstatt den Tabellennamen manuell in das Feld **Name** einzutragen.

Bereich Laufzeit

In diesem Bereich können Sie Laufzeitwerte festlegen. Folgende Felder stehen zur Verfügung:

- **Status** (Ausgabe): Der Tag in diesem Feld erhält einen der folgenden Werte:

Wert	Beschreibung
0	Von der Datenbank getrennt. Die Datenbank ist nicht verfügbar, ihre Konfiguration ist fehlerhaft oder es handelt sich um eine unzulässige Operation
1	Die Datenbank ist erfolgreich verbunden
2	Die Datenbank wird derzeit synchronisiert

- **Neu laden** (Ausgabe): Geben Sie einen „Neu laden“-Tag an, wenn Sie in einem Konfigurationsfeld geschweifte Klammern verwenden. Wenn Sie sich mit der Datenbank neu verbinden und hierbei die aktualisieren Werte Ihrer Tags verwenden wollen, setzen Sie den Tag auf diesem Feld auf 1. "WinStudio" aktualisiert dann die Konfiguration, wenn es mit der Datenbank eine Aktion ausführen will, und setzt den Tag zurück auf 0, wenn diese abgeschlossen ist.

Weitere Informationen unter:

Konfigurieren einer Standarddatenbank für die gesamte Task-Historie auf Seite 508.

Registerkarte Viewer

Registerkarte Viewer, Allgemeines

Mithilfe dieser Registerkarte konfigurieren Sie den Projekt-Viewer und ändern bestimmte Verhaltensweisen zur Laufzeit.

Entwicklungsumgebung

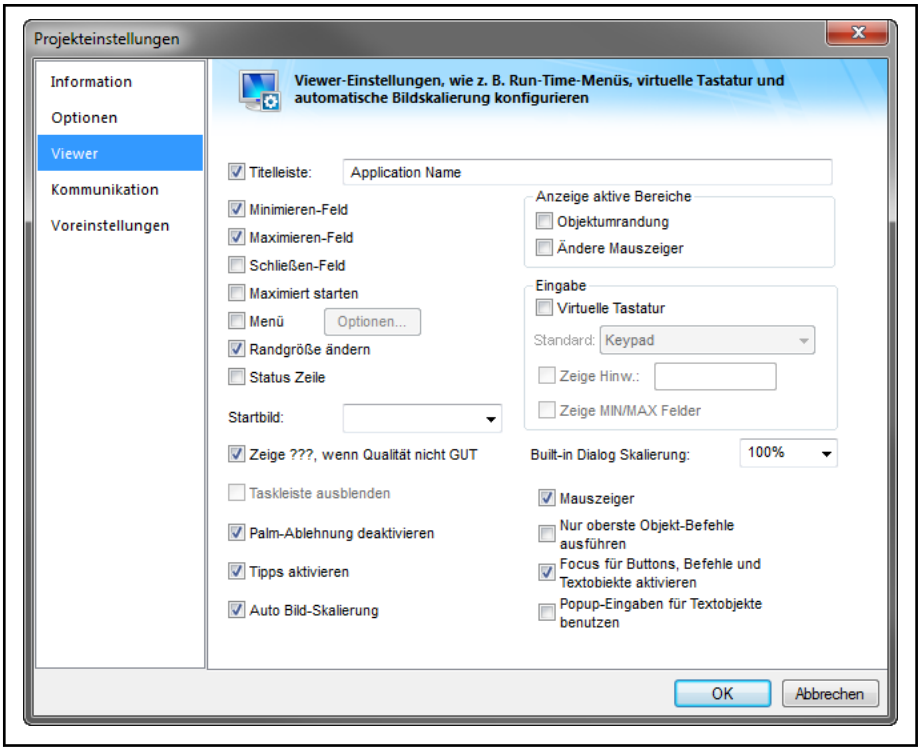


Abb.5-25: Projekteinstellungen: Registerkarte Laufzeit-Desktop

- **Titelleiste**: Klicken (aktivieren) Sie dieses Kästchen und geben Sie einen neuen Namen in das dazugehörige Feld ein, um den Standardtext der Titelleiste für das Viewer-Fenster anzugeben oder zu ändern.
- **Minimieren**, **Maximieren** und **Schließen**: Klicken (aktivieren) Sie diese Kontrollkästchen an, um diese Schaltflächen im Viewer-Fenster einzublenden (aktivieren) oder auszublenden (deaktivieren).
- **Vollbild starten**: Klicken (aktivieren) Sie dieses Kästchen , um das Viewer-Fenster automatisch zu maximieren, wenn Sie Ihr Projekt ausführen.
- **Menü** und Schaltfläche **Optionen ...**: Klicken (aktivieren) Sie das Kontrollkästchen und die Schaltfläche **Optionen ...** an, um anzugeben, welche Menüoptionen während der Laufzeit verfügbar sind. Wenn der Dialog "Laufzeit-Menüoptionen" (wie nachstehend) angezeigt wird, klicken Sie die jeweiligen Kontrollkästchen an, um die zugehörigen Menüoptionen anzuzeigen (zu aktivieren) bzw. auszublenden (zu deaktivieren).



Abb.5-26: Dialog Laufzeit-Menüoptionen

- **Randgröße**: Klicken (aktivieren) Sie dieses Kontrollkästchen, um dem Benutzer die Möglichkeit zu geben, die Größe des Viewer-Fensters während der Laufzeit zu verändern.

- **Statuszeile**: Klicken (aktivieren) Sie dieses Kontrollkästchen, um die Statuszeile im Laufzeitprojekt anzeigen zu lassen.
- **Startbild**: Klicken Sie auf die Auswahlliste und wählen Sie ein WinStudio-Bild oder eine Bildgruppe aus, die beim Öffnen des Laufzeitprojekts automatisch angezeigt werden soll.

☞ Eine andere Möglichkeit, ein Bild bzw. Bildgruppe als Startbildschirm anzugeben, ist ein Rechtsklick auf das Element im "Projektexplorer" und die Auswahl **Als Startbildschirm**.

- **Zeige ????, wenn die Qualität nicht GUT ist**: Klicken (aktivieren) Sie dieses Kästchen, um Fragezeichen (???) anstelle des jeweiligen Tag-Wertes anzuzeigen, wenn dessen Qualität nicht gut ist.
- **Taskleiste ausblenden**: Klicken (aktivieren) Sie dieses Kästchen, um die Windows-Taskleiste standardmäßig auszublenden.
- **Palm Rejection deaktivieren**: Wählen Sie diese Option aus, um Palm Rejection während der Projektlaufzeit zu deaktivieren. Palm Rejection ist ein Merkmal von Windows-7-Touchscreen-Geräten; hierbei werden versehentliche Berührungen mit der Handfläche des Bedieners erkannt und nicht als Signal angenommen. Dies kann jedoch das Ansprechverhalten des Touchscreens merklich verlangsamen, sodass die Deaktivierung der Funktion die Leistungsgeschwindigkeit des Projekt-Laufzeitobjekt verbessert kann.

☞ Palm Rejection ist ausschließlich auf Windows-7-Touchscreen-Geräten verfügbar, daher hat die Auswahl von **Palm Rejection deaktivieren** keine Auswirkungen auf Projekte, die auf anderen Zielsystemen ausgeführt werden.

☞ Windows-7-Touchscreen-Geräte verfügen auch über eine Option, mit der der Anwender auf den Bildschirm drücken und den Druck halten kann, um einen Rechtsklick auszuführen („press and hold“). Auf Endverbrauchergeräten wie Tablets und Smartphones ist das nützlich, aber bei der Arbeit mit einem "WinStudio"-Projekt kann es stören, wenn der Benutzer Schaltflächen und andere Bildobjekte drücken und gedrückt halten muss. Diese Option ist Standard. So deaktivieren Sie diese Option im Gerät:

1. Klicken Sie auf die Schaltfläche Start und klicken Sie im Startmenü auf Systemsteuerung. Das Fenster Systemsteuerung wird angezeigt.
2. Doppelklicken Sie auf Eingabestift und Touch. Der Dialog Eingabestift und Touch wird angezeigt.
3. Klicken Sie auf die Registerkarte Touch.
4. Wählen Sie aus der Liste der Touch-Aktionen „Drücken und Halten“ und klicken Sie dann auf Einstellungen. Der Dialog Drücken und Halten – Einstellungen wird angezeigt.
5. Löschen Sie „Drücken und Halten für Rechtsklick aktivieren“.
6. Klicken Sie auf OK, um den Dialog „Drücken und Halten – Einstellungen“ zu schließen.
7. Klicken Sie auf OK, um den Dialog „Eingabestift und Touch“ zu schließen.

- **Tipps aktivieren**: Klicken (aktivieren) Sie dieses Kontrollkästchen , um während der Ausführung Ihres Projekts Windows-ToolTips zu sehen. Tooltips können Sie im Feld **Tipp** des Dialogs "Objekteigenschaften" des jeweiligen Objekts konfigurieren.
- **Autom. Bildskalierung**: Klicken (aktivieren) Sie dieses Kästchen , um die Größe der WinStudio-Bilder automatisch anzupassen, wenn Sie die Größe des Viewer-Fensters ändern. Diese Funktion ist für lokale Projekte verfügbar, die auf einem Windows-PC oder im Thin Client ausgeführt werden. "Dieser Parameter ist nicht für Projekte verfügbar, die auf Windows-Embedded-Zielsystemen ausgeführt werden."
- Bereich "Anzeige von aktiven Bereichen": Klicken Sie in diesem Bereich die Kontrollkästchen **Objektumrandung** bzw. Mauszeiger ändern , um den Objektrand bzw. den Mauszeiger zu ändern, wenn der Cursor sich über ein Objekt bewegt, auf das die Befehls-Animation angewendet wurde.
- **Virtuelle Tastatur**: Wenn diese Option angehakt ist, ist die virtuelle Tastatur auf Seite 80 für Ihr Projekt freigegeben. (Diese Option ist "nicht" anwendbar, wenn Ihr Projekt auf einem Thin Client ausgeführt wird; nähere Informationen hierzu finden Sie unter Projekteinstellungen: Web auf Seite 86.) Mithilfe dieser Tastatur kann der Benutzer während der Laufzeit Daten auf Touchscreens eingeben – also ohne Eingabe auf einer physischen Tastatur. Zum Beispiel ein Textobjekt auf Seite 190, auf das die Animation Text-Daten-Link auf Seite 249 angewendet wurde und bei dem die Option **Eingaben freigegeben** aktiviert wurde. Sie können für die virtuelle Tastatur eine Standardkonfiguration einstellen:
  - **Standard:** Wählen Sie den Standard-Tastaturtyp, der in Ihrem Projekt verwendet werden soll, wenn durch das aufrufende Objekt bzw. die aufrufende Funktion kein Tastaturtyp festgelegt wird.
  - **Skalierung:** Mit dieser Option können Sie die Tastatur so verkleinern bzw. vergrößern, dass sie sich der Größe des Zielbildschirms anpasst. Ein Wert von 100 % steht für die Standardgröße des jeweiligen Tastaturtyps.
  - **Zeige Hinweise**: Wenn diese Option angehakt ist, wird ein Tipp in der Titelleiste der Tastatur angezeigt. Dieser Tipp wird für die einzelnen Objekte im Dialog Objekteigenschaften festgelegt. Für andere Fälle tragen Sie einen String-Wert in das Feld „Tipp anzeigen“ ein, der dann als Standardtipp dient.
  - **Zeige Min/Max-Felder**: Wenn diese Option angehakt ist, werden die erlaubten Mindest- und Höchstwerte am unteren Rand der Tastatur angezeigt. Diese Werte werden bei Objekten im Dialog Objekteigenschaften festgelegt. Für andere Fälle ist Standard, dass die Eigenschaften Min und Max des zugehörigen Tags verwendet werden.

☞ Die Felder Min und Max werden nur auf dem Tastaturtyp Tastenfeld angezeigt, und nur dann, wenn der zugeordnete Tag als Integer oder Real definiert ist. Wenn Min größer ist als Max, wird die Eingabe deaktiviert. Wenn das am Objekt eingestellte Paar Min/Max sich von den Eigenschaften Min/Max im Tag unterscheidet, dann versucht Ihr Projekt, die Größe der Eingabe entsprechend umzurechnen.

- **Mauszeiger**: Klicken (aktivieren) Sie dieses Kontrollkästchen, um den Mauszeiger im Laufzeitprojekt anzuzeigen.

Die Option Mauszeiger wird unter Windows CE, wenn es auf Armv4l-Prozessoren ausgeführt wird, nicht unterstützt.

- **Nur oberste Objekt-Befehle ausführen**: Diese Option steuert, wie sich Ihr Projekt verhält, wenn der Benutzer in einen Bereich klickt, in dem sich zwei oder mehr Bildobjekte überschneiden. Wenn diese Option angehakt ist, werden nur die Befehle des obersten Objekts ausgeführt. Wenn diese Option nicht angehakt ist, dann werden die Befehle aller überlappenden Objekte ausgeführt.

Das Objekt mit der höchsten ID-Nummer gilt als das oberste. (Die ID-Nummer eines Objekts wird in der Statuszeile auf Seite 96 am unteren Rand der Entwicklungsumgebung angezeigt.) Mithilfe der Werkzeuge Ganz nach hinten / Ganz nach vorne verschieben auf Seite 258 und Nach hinten / Nach vorne verschieben auf Seite 259 können Sie die Reihenfolge verändern, in der die Objekte übereinander liegen.

- **Fokus für Schaltflächen, Befehle und Textobjekte aktivieren**: Diese Option ist standardmäßig aktiviert. Wenn sie aktiviert ist, bewirkt ein Klick auf eine Befehlsschaltfläche oder eine Texteingabe während der Laufzeit dazu, dass der Fokus auf dem jeweiligen Objekt liegt, wie in der unten stehenden Illustration gezeigt:

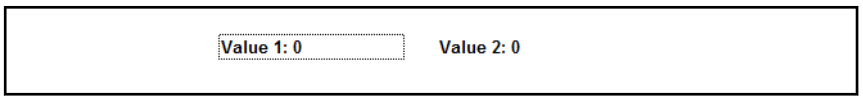


Abb.5-27: Fokus auf Texteingabe (links), kein Fokus auf Texteingabe (rechts)

Danach können Sie mithilfe der **Tab**-Taste nacheinander zu allen fokussierbaren Objekte im Bild navigieren oder mithilfe der **Enter**-Taste das Objekt aktivieren (z.B. auf eine Befehlsschaltfläche klicken, etwas in ein Textfeld eingeben), auf dem gerade der Fokus liegt, ähnlich wie bei einer normalen Windows-Anwendung. Das ist nützlich, wenn die Client-Arbeitsstation mit einer physischen Tastatur ausgestattet ist und Sie sich schnell durch viele solche Objekte durcharbeiten müssen.

Löschen Sie diese Option, wenn Sie den Endbenutzer dazu zwingen wollen, Bildschirmobjekte immer mit der Maus bzw. dem Touchscreen zu aktivieren.

- **Popup-Eingabe für Textobjekte verwenden**: Wählen Sie diese Option, wenn für Texteingaben ein kleines Popup angezeigt werden soll, wie in der unten stehenden Illustration gezeigt:

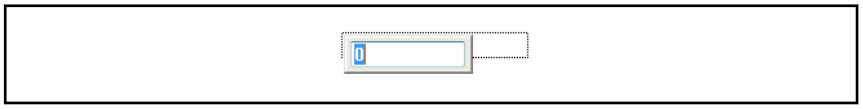


Abb.5-28: Popup für Texteingaben

Dies ist eine Alternative zur direkten Eingabe in das Textobjekt, und zur Anzeige einer virtuellen Tastatur für die Eingabe (was die Verwendung der Maus oder eines Touchscreens erfordert).

Wenn die (oben beschriebene) Option **Virtuelle Tastatur** ausgewählt ist, setzt sie diese Option außer Kraft.

Zudem gilt: Wenn die (oben beschriebene) Option **Fokus auf Schaltflächen, Befehle und Textobjekte aktivieren** "nicht" gesetzt ist, dann ist diese Option automatisch ausgewählt. Hiermit soll sichergestellt werden, dass es im Bild einen Hinweis gibt, welche Texteingabe jeweils aktuell aktiv ist.

Tastatur

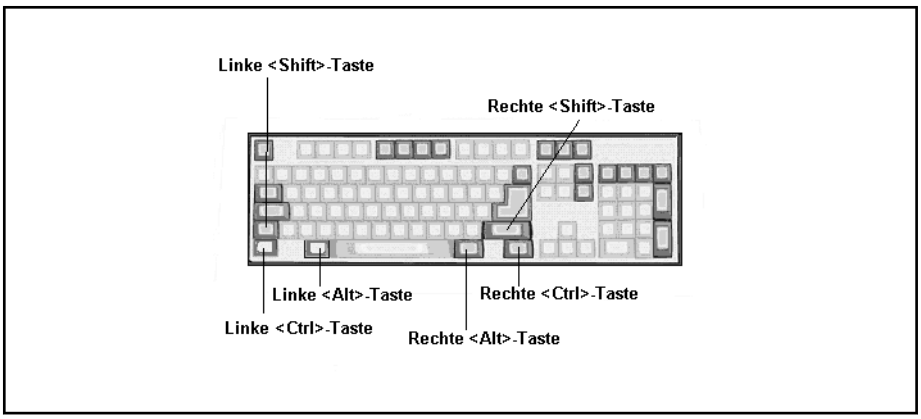


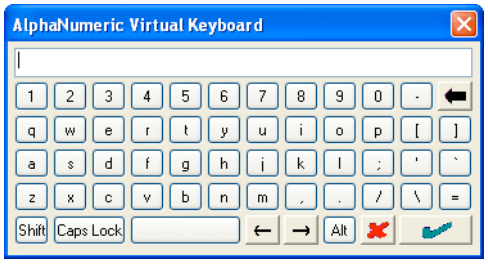
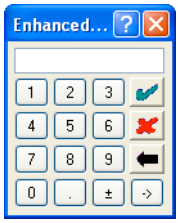
Abb.5-29: Tastatur

Virtuelle Tastatur

Mithilfe der virtuellen Tastatur (VT) können Sie Daten (Text bzw. Zahlen) während der Laufzeit über ein Touchscreen-Gerät anstatt einer physischen Tastatur eingeben.

Die VT können Sie für Ihr Projekt über den Dialog Laufzeit-Desktop auf Seite 75 freigeben. Sie können einen Standard-Tastaturtyp vorgeben und eine Skalierung (Anpassungsfaktor) für alle zur Laufzeit eingeblendeten virtuellen Tastaturen anwenden lassen.

Wenn Sie Objekte und Animationen konfigurieren, bei denen die Dateneingabe unterstützt wird, können Sie dem jeweiligen Objekt eine VT mithilfe der VT-Combobox zuweisen (entweder die Standard-VT oder eine spezielle). Folgende Standard-VT stehen zur Verfügung:

Name	Beschreibung	
Alphanumerisch	Diese VT dient zum Eingeben alphanumerischer Werte (Buchstaben und/oder Ziffern).	
ErwTastenfeld	Diese VT dient zum Eingeben alphanumerischer Werte (Buchstaben und/oder Ziffern) auf Geräten, die nur über eine kleine Anzeige verfügen (z.B. PDAs). Die Benutzer können mithilfe der Schaltfläche "->" durch verschiedene Tastaturbeschriftungen blättern, um das gewünschte Zeichen zu finden.	
Tastenfeld	Diese VT dient zum Eingeben numerischer Werte (nur Ziffern).	

Es gibt zwei Möglichkeiten zum Starten einer virtuellen Tastatur während der Laufzeit:

- Anklicken des Objekts an der Stelle, an der ein neuer Wert (neue Daten) eingetragen werden kann (z.B. die Animation Text mit Textdaten-Link mit freigegebener Eingabe).
- Ausführen der eingebauten Funktion Tastenfeld() auf Seite 630.

Die Sprache der virtuellen Tastatur können Sie mithilfe der eingebauten Funktion SetKeyboardLanguage() auf Seite 707 ändern.

- Für jede virtuelle Tastatur können Sie neue Sprachen anlegen, indem Sie die Datei **VKlanguage.INI** im Unterordner **\BIN** von "WinStudio" bearbeiten; hierbei steht **language** für die Beschreibung der Sprache für die virtuelle Tastatur (z.B. gilt **VKEN.INI** für englische virtuelle Tastaturen, **VKGE.INI** für deutsche virtuelle Tastaturen usw.).

  Für weitere Informationen zum Bearbeiten und Konfigurieren dieser Dateien wenden Sie sich bitte an Ihren Softwarehersteller.
- Standardmäßig wird die virtuelle Tastatur in der Nähe des Objekts gestartet, das mit ihr verbunden ist. Jedoch können Sie die folgenden Einstellungen in der Projektdatei (**project_name.app**) so konfigurieren, dass die VT immer an einer festen Stelle geöffnet wird:

  [Keypad] PosX=0 //Koordinate (in Pixel) der OBEREN Koordinate, bei der die VT zu starten ist.

  PosY=0 //Koordinate (in Pixel) der LINKEN Koordinate, bei der die VT zu starten ist.

Registerkarte Kommunikation

Mit dieser Registerkarte geben Sie Kommunikationsparameter an, die sich auf Ihr Projekt im Allgemeinen beziehen.

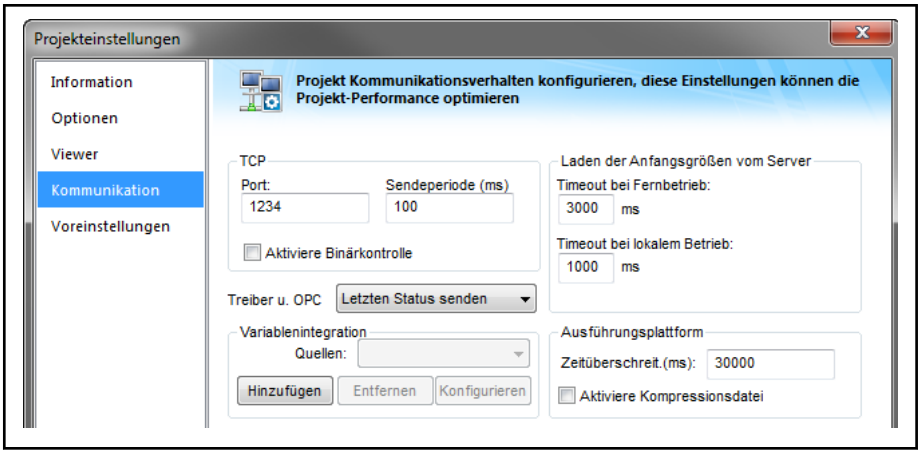


Abb.5-30: Projekteinstellungen: Registerkarte Kommunikation

- **Treiber und OPC**: Wählen Sie beim Schreiben von Werten in des Remote-Gerät/die Remote_PLC die von allen Kommunikationstreibern auf Seite 379 und OPC-Client-Arbeitsblättern auf Seite 390 verwendete Methode im aktuellen Projekt aus:
  - **Jeden Zustand senden**: Wenn der Kommunikations-Task so konfiguriert ist, dass Werte bei Änderung eines Tag-Werts geschrieben werden, dann werden alle Änderungen im Tag-Wert in einer Warteschlange gepuffert und zum Gerät geschickt, wenn der Kommunikations-Task (Treiber bzw. OPC) ausgeführt wird.
  - **Letzten Zustand senden**: Wenn der Kommunikations-Task so konfiguriert ist, dass Werte bei Änderung eines Tag-Werts geschrieben werden, so wird nur der aktuelle (letzte) Wert des Tags zum Gerät geschickt, wenn der Kommunikations-Task (Treiber bzw. OPC) ausgeführt wird. Wenn diese Methode ausgewählt ist und sich der Tag-Wert mehr als einmal während der Ausführung des während der Kommunikations-Tasks geändert hat, so werden die vorübergehend gültigen Werte des Tags nicht an das Gerät geschickt. Dies stellt für die meisten Projekte das gewünschte Verhalten dar.
- Bereich "TCP": Konfigurieren Sie hier die Kommunikationseinstellungen für die TCP/IP-Client- und -Server-Module auf Seite 401:
  - **Port**: Der von den auf dem aktuellen Rechner laufenden TCP/IP-Client- und TCP/IP-Server-Modulen eingesetzte TCP-Port. Wenn Sie diesen Wert im lokalen Projekt ändern, stellen Sie unbedingt in einem Remote-Projekt, das mit dem lokalen kommuniziert, den gleichen Wert ein.

☞ Diese Einstellung ist für alle Projekte ein Pflichtfeld und kann nicht leer gelassen werden. Stellen Sie sicher, dass Ihre Netzkonfiguration Verbindungen zu diesem Port auf der Station zulässt, die den Projektserver bilden wird.

  - **Sendeperiode (ms)**: Zeitspanne (in Millisekunden) zwischen zwei aufeinanderfolgenden Mitteilungen, die von den TCP/IP-Client- oder TCP/IP-Server-Modulen gesendet werden, um Tag-Werte in der Remote-Station zu aktualisieren. Typischerweise bedeutet ein kleinerer Wert bessere Leistung und höheren Traffic im Netzwerk.
  - **Aktiviere Binärsteuerung**: Wählen Sie diese Option aus, um die binäre Steuerung zu aktivieren, wenn das TCP/IP-Client- oder

TCP/IP-Server-Modul Mitteilungen an die Remote-Station sendet. Mit binärer Steuerung wird die Sicherheit des Systems erhöht; sie verringert allerdings die Effizienz (Leistung) der Kommunikation. Wenn Sie zwei Stationen über das TCP/IP-Client- und TCP/IP-Server-Modul verbinden, stellen Sie sicher, dass diese Einstellung entweder in beiden Projekten aktiviert oder in beiden Projekten deaktiviert ist.

- Bereich "Ausführungsplattform": Konfigurieren Sie hier die Kommunikationseinstellungen für den Remote-Verwaltungsbetrieb auf Seite 66, mit dem Ihre Projektdateien auf ein Zielsystem geschickt werden:
  - **Zeitüberschreitung (ms)**: Legt die Zeit (in Millisekunden) fest, die das Projekt auf die Kommunikation mit dem Zielsystem wartet.
  - **Dateikomprimierung aktivieren**: Wählen Sie diese Option aus, um die System- und Projektdateien zu komprimieren, bevor sie an das Zielsystem gesendet werden. Möglicherweise verringert dies die Ladezeit, wenn zwischen Ihrem Server und dem Zielsystem eine langsame Verbindung besteht. (Wenn Sie jedoch eine schnelle Verbindung haben, "verringert" das Auswählen dieser Option möglicherweise die Leistung, da jede komprimierte Datei auf dem Zielsystem vor dem Senden der nächsten Datei erst dekomprimiert werden muss. Wählen Sie diese Option nur aus, wenn Sie eine äußerst langsame Verbindung, z.B. eine DFÜ-Verbindung, haben). Standardmäßig ist die Datenkomprimierung deaktiviert.
- Bereich "Laden der Ausgangsgrößen vom Server": Zur Verbesserung der Leistung lädt der Laufzeit-Viewer vor dem Anzeigen eines Bildes alle Server-Tags dieses Bildes in den Speicher. Konfigurieren Sie hier die Zeitüberschreitungs-Einstellungen für die Remote- und für die lokalen Viewer:
  - Feld **Zeitüberschreitung bei Ausführung im Remote-Gerät**: Legt die Zeit (in Millisekunden) fest, die ein auf einer Remote-Station ausgeführter Secure Viewer oder Thin Client auf das Laden der Tags wartet.
  - Feld **Zeitüberschreitung bei lokaler Ausführung**: Legt die Zeit (in Millisekunden) fest, die ein auf einer lokalen Station (d.h. auf dem Server) ausgeführter Viewer auf das Laden der Tags wartet.
- Bereich **Variablenintegration**: Hiermit können Sie Tags auf Zielgeräten verbinden und gemeinsam verwenden. Nähere Informationen hierzu finden Sie unter „Tags auf IndraLogic-Geräten gemeinsam verwenden“ auf Seite 163.

Registerkarte Voreinstellungen

Mit dieser Registerkarte konfigurieren Sie die Einstellungen, die Sie persönlich bevorzugen.

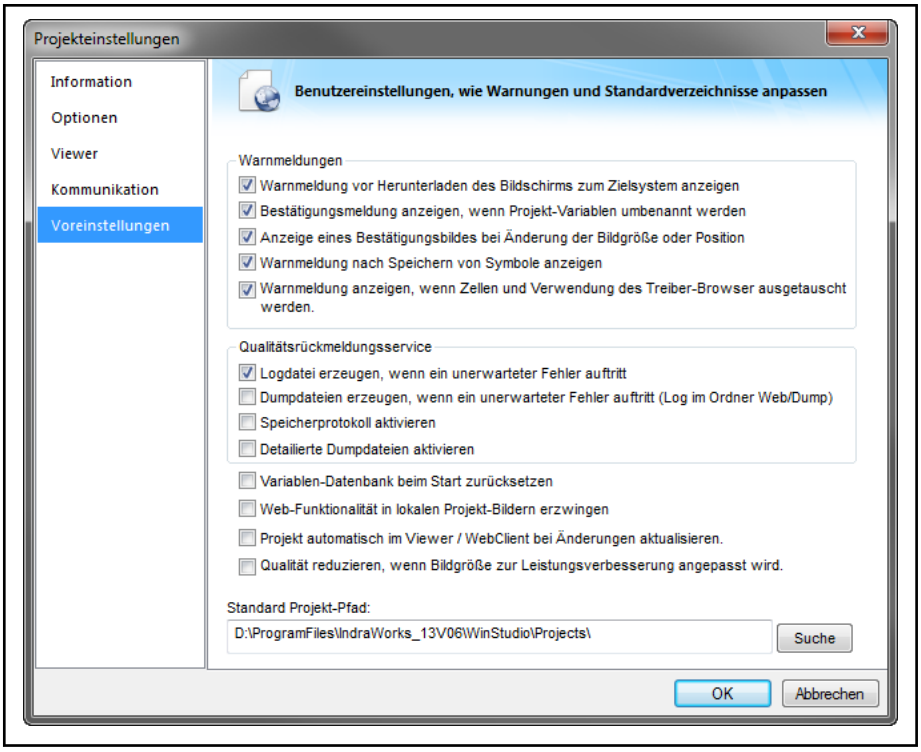


Abb.5-31: Projekteinstellungen: Registerkarte Voreinstellungen

Warnmeldungen

- **Warnmeldung vor dem Herunterladen des Bildschirms zum Zielsystem anzeigen**: Wenn diese Option ausgewählt ist und "WinStudio" mit einer Remote-Station (Dialog Remote-Verwaltung auf Seite 66) verbunden ist, werden Sie direkt nach dem Speichern eines Bildes im Bild-Editor aufgefordert, das aktualisierte Bild auf die Remote-Station herunter zu laden. Wenn diese Option nicht angehakt ist, wird das Bild automatisch herunter geladen, egal ob dies bestätigt wird oder nicht.
- **Bestätigungsmeldung anzeigen, wenn Projekt-Variablen umbenannt werden**: Wenn diese Option ausgewählt ist und Sie den Namen eines Tags aus dem Arbeitsblatt Projekt-Tags auf Seite 116 ändern, werden Sie aufgefordert, im gesamten Projekt den alten Tag-Namen durch den neuen Tag-Namen zu ersetzen. Wenn diese Option angehakt ist, führt "WinStudio" den Befehl „Globales ersetzen“ aus und tauscht so in allen Dokumenten (Bildern und Arbeitsblättern) Ihres Projekts den alten gegen den neuen Tag-Namen aus.
- **Anzeige einer Bestätigungsmeldung bei Änderung der Bildgröße oder Position**: Wenn diese Option ausgewählt ist, werden Sie nach Änderung von Bildschirmattributen (Breite, Höhe, Oben und Links) auf der Benutzeroberfläche <Layout> dazu aufgefordert, diese zu aktualisieren.
- **Warnmeldung nach Speichern von Symbolen anzeigen**: Wenn diese Option ausgewählt ist, wird nach dem Speichern von Symbolen auf Seite 220 eine Warnmeldung angezeigt.

Qualitätsrückmeldungsservice

In diesem Abschnitt können Sie Ihr Projekt so konfigurieren, dass Protokolldateien und/oder Speicherauszugsdateien erzeugt werden, die zum Diagnostizieren von Hardware- und Softwareproblemen wie Speicherlecks und unerwartete Fehler eingesetzt werden können. Diese Dateien werden im Unterordner **\Web\Dump** des laufenden Projekts abgespeichert.

- **Logdatei erzeugen, wenn ein unerwarteten Fehler auftritt**: Wenn diese Option ausgewählt wird, hängen die Laufzeitmodule die Protokolldatei

(**\Web\Dump\Dump.txt**) immer an, wenn eine interne Ausnahme (ein Fehler) auftritt. Diese Ausnahmen führen nicht notwendigerweise zum Absturz der Laufzeitmodule, sie beeinträchtigen jedoch möglicherweise die Stabilität des Systems und sollten daher untersucht werden.

☞ An die Protokolldatei werden neue Einträge laufend angehängt, bis sie ihre maximale Größe von 2 MB erreicht. Nachdem die Datei ihre Maximalgröße erreicht, wird die bestehende Datei gelöscht und eine neue angelegt.

- **Dumpdateien erzeugen, wenn ein unerwarteten Fehler auftritt**: Wenn diese Option ausgewählt ist, erzeugen die Laufzeitmodule eine neue Speicherauszugsdatei (**\Web\Dump*.dmp**) mit nützlichen Informationen über die Fehlerumstände. Dies ist eine Binärdatei, die nur vom Softwarehersteller gelesen werden kann.

☞ Speicherauszugsdateien tragen den Namen **WinXXX.dmp** – hierbei steht "XXX" für eine vom System generierte Kennnummer (im Hexadezimalformat) – um zu verhindern, dass bei Auftreten eines neuen Fehlers eine bestehende Datei überschrieben wird. Wenn also mehr als ein Fehler auftritt, finden Sie im Verzeichnis mehrere Speicherauszugsdateien. Die Protokolldatei nennt zu jedem jeweiligen Fehler den Namen der zugehörigen Speicherauszugsdatei.

- **Speicherprotokoll aktivieren**: Wenn diese Option ausgewählt wird, hängen die Laufzeitmodule alle 15 Minuten Informationen über die derzeitigen Speicherzuweisungen an die Protokolldatei an. (Der erste Protokolleintrag wird 15 Minuten nach dem Starten des jeweiligen Laufzeitmoduls geschrieben). Diese Informationen lassen sich zum Auffinden von Speicherlecks verwenden.
- **Detaillierte Dumpdateien aktivieren**: Wenn diese Option ausgewählt wird, werden zahlreiche Details protokolliert, die bei der Analyse des Fehlers helfen.

Wenn ein Laufzeitmodul durch einen fatalen Fehler beendet wird, dann wird zur Analyse des Absturzes selbst dann ein Speicherauszug erzeugt (**\Web\Dump\WinDump.dmp**), wenn keine dieser Optionen zur Qualitätsrückmeldung angehakt sind. Zu Fehlerbeseitigungszwecken wird jedoch dringend empfohlen, dass Sie dann alle Optionen dieses Abschnitts aktivieren und die Protokolldatei und alle Speicherauszugsdateien an Ihren Softwarehersteller schicken.

Andere Voreinstellungen

- **Variablen-Datenbank beim Start zurücksetzen**: Wenn diese Option ausgewählt ist, werden die Projekt-Tags beim Ausführen der Laufzeit immer automatisch zurückgesetzt Weitere Einzelheiten zu dieser Funktion finden Sie unter Tag-Datenbank zurücksetzen auf Seite 152.
- **Web-Funktionalität in lokalen Projekt-Bildern erzwingen**: Wenn diese Option ausgewählt ist, warnt WinStudio automatisch, wenn Sie Funktionen oder Merkmale auswählen, die zu den Remote-Laufzeitmodulen (z.B. Thin Client und Secure Viewer) inkompatibel sind.

☞ Standardmäßig ist diese Option nicht angehakt, um die Kompatibilität mit Vorversionen von "Rexroth WinStudio" zu wahren.

- **Projekt automatisch im Viewer / Web-Client bei Änderungen aktualisieren**: Wenn diese Option ausgewählt ist, überprüfen Remote-Stationen (d.h. Thin Clients und Secure Viewers) auf dem Server, ob sie über die neueste Version des Projekts verfügen. Falls nicht, laden sie automatisch die neue Version vom Server herunter.

Registerkarte Web

Die Registerkarte Web erreichen Sie über **WinStudio** ▸ **Werkzeuge** ▸ **Web-Einstellungen**.

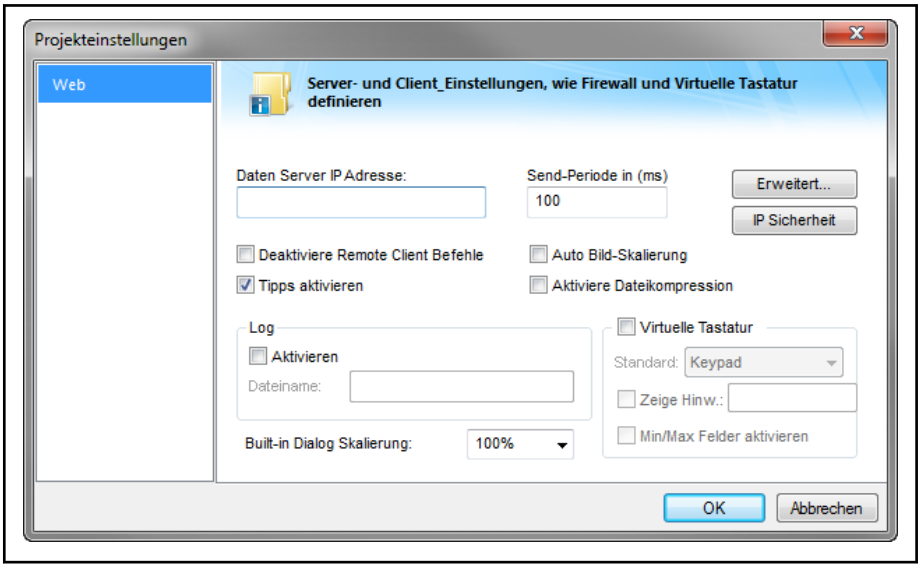


Abb.5-32: Projekteinstellungen: Registerkarte Web

☞ Wenn Sie hier Einstellungen ändern, sollten Sie Ihr Projekt überprüfen auf Seite 64, bevor Sie es wieder an das Zielgerät schicken.

Die Parameter auf dieser Registerkarte werden wie folgt konfiguriert:

- **Datenserver-IP Adresse**: Geben Sie hier die IP-Adresse (bzw. den Host-Namen) Ihrer Datenserverstation ein. Die Datenserverstation ist der Rechner bzw. das Gerät, auf dem das TCP/IP-Server-Modul von "WinStudio" ausgeführt wird. Wenn dieses Feld leer gelassen wird, nimmt der Thin Client an, dass es sich bei dem Web-Server (d.h. bei der in den Browser eingegebenen Adresse) auch um den Daten-Server handelt.

☞ Über die IP-Adresse "127.0.0.1" (localhost) können Sie auf den TCP/IP-Server auf dem lokalen Rechner zugreifen, ungeachtet dessen tatsächlicher Adresse im Netzwerk. Diese Option ist nützlich für lokale Tests, Sie können mit dieser Konfiguration jedoch nicht von fernen Rechnern aus auf den Datenserver zugreifen.

- **Sende-Periode in ms**: Geben Sie hier einen Wert ein, um die Zeitspanne (in Millisekunden) festzulegen, mit der Daten zwischen dem Server und den Thin-Client-Stationen ausgetauscht werden. Das bedeutet, dass der Thin Client alle "n" Millisekunden ein Paket mit den neuen Tag-Werten zum Server schickt. Die **Sende-Periode** des Servers wird in den Projekteinstellungen (auf der Registerkarte **Kommunikation** in den Projekt-Eigenschaften) konfiguriert. Der Standardwert beträgt "1000" (Millisekunden). Sie können in diesem Feld einen geringeren Wert einstellen, um die Aktualisierungsrate zwischen Server und Thin Clients zu erhö-

hen. Dies könnte zu höherem Datenverkehr im Netzwerk führen (auf das Netzwerk wird häufiger zugegriffen), wenn sich die Tags laufend (schneller als einmal pro Sekunde) ändern.

- **Deaktiviere Remote Client Befehle**: Klicken Sie dieses Kästchen an, um zu verhindern, dass ein Remote Client Befehle von Ihrem Thin Client an Ihren Server schickt. Wenn diese Option aktiviert ist, kann der Thin Client zwar Daten vom Server lesen, aber keine Daten (Tag-Werte, Sollwerte) an den Datenserver senden. In diesem Fall wird die Thin-Client-Station zu einer "schreibgeschützten" Station.
- **ToolTips aktivieren**: Klicken Sie dieses Kontrollkästchen an, um die für die Bildobjekte konfigurierten "ToolTips" anzuzeigen, wenn diese auf dem Thin Client (Web-Browser) angezeigt werden.
- **Autom. Bildskalierung**: Klicken Sie dieses Kontrollkästchen an, um die in einem Web-Browser-Fenster angezeigten Bilder automatisch in der Größe anzupassen. Mit dieser Option passt sich das Bild an die Abmessungen des Web-Browser-Fensters an, unabhängig von dessen Abmessungen.

☞ Die Option **Autom. Bildskalierung** gilt nicht für Web-Browser, die unter Windows CE Betriebssystemen laufen.

- **Aktiviere Dateikomprimierung**: Klicken Sie dieses Kontrollkästchen an, um die Dateien im "Web"-Unterordner Ihres Projektordners zu komprimieren. Diese Option ist zur Verringerung der Ladezeit nützlich, besonders bei einer langsamen Verbindung zwischen Ihrem Server und dem Thin Client haben.
- **Log** (**Aktivieren** und **DateiName**): Klicken Sie das Kontrollkästchen an und geben Sie einen Dateinamen im Textfeld ein, um eine Protokolldatei auf der Thin-Client-Station zu erzeugen. Diese Protokolldatei können Sie zu Fehlerbeseitigungszwecken einsetzen.
- **Virtuelle Tastatur**: Wenn diese Option angehakt ist, ist die virtuelle Tastatur auf Seite 80 für Ihr Projekt mit Thin Clients freigegeben. Mithilfe dieser Tastatur kann der Benutzer während der Laufzeit Daten auf Touchscreens eingeben – also ohne Eingabe auf einer physischen Tastatur. Zum Beispiel ein Textobjekt auf Seite 190, auf das die Animation Text-Daten-Link auf Seite 249 angewendet wurde und bei dem die Option **Eingaben freigegeben** aktiviert wurde.

  Sie können für die virtuelle Tastatur eine Standardkonfiguration einstellen:

  – **Standard**: Wählen Sie den Standard-Tastaturtyp, der in Ihrem Projekt verwendet werden soll, wenn durch das aufrufende Objekt bzw. die aufrufende Funktion kein Tastaturtyp festgelegt wird.

  – **Skalierung**: Mit dieser Option können Sie die Tastatur so verkleinern bzw. vergrößern, dass sie sich der Größe des Zielbildschirms anpasst. Ein Wert von 100% steht für die Standardgröße des jeweiligen Tastaturtyps.

  – **Zeige Hinweise**: Wenn diese Option angehakt ist, wird ein Tipp in der Titelleiste der Tastatur angezeigt. Für Objekte wird dieser Tipp im Dialog Objekteigenschaften auf Seite 178 festgelegt. Für andere Fälle tragen Sie eine Zeichenkette oder einen Zeichenketten-Tag in das Feld **Tipp anzeigen** ein, der dann als Standardtipp dient.

– **Min/Max-Felder aktivieren**: Wenn diese Option angehakt ist, werden die erlaubten Mindest- und Höchstwerte am unteren Rand der Tastatur angezeigt. Für Objekte werden diese Werte im Dialog Objekteigenschaften auf Seite 178 festgelegt. Für andere Fälle werden als Standard die Eigenschaften Min und Max auf Seite 153 des zugehörigen Tags verwendet.

☞ Die Felder **Min** und **Max** werden nur auf dem Tastaturtyp **Tastenfeld** angezeigt, und nur dann, wenn der zugeordnete Tag als Integer oder Real definiert ist. Wenn Min größer ist als Max, wird die Eingabe deaktiviert. Wenn das am Objekt eingestellte Paar Min/Max sich von den Eigenschaften Min/Max im Tag unterscheidet, versucht Ihr Projekt, die Größe der Eingabe entsprechend umzurechnen.

Einstellungen zur IP-Sicherheit

IP-Sicherheit: Klicken Sie auf diese Schaltflächen, um den Dialog IP-Sicherheit zu öffnen.

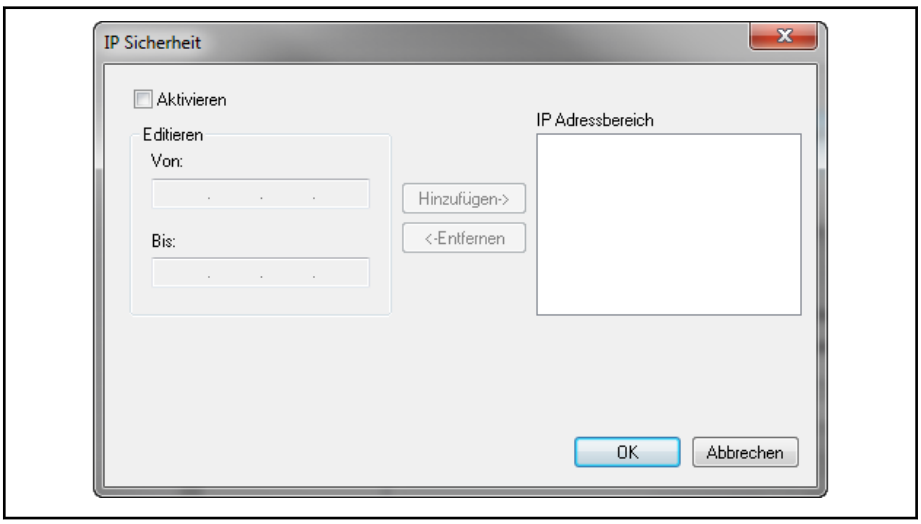


Abb.5-33: Dialog IP-Sicherheit

Mithilfe der Parameter dieses Dialogs geben Sie den Bereich von IP-Adressen für die Rechner an, die als Thin Clients auf Ihr Projekt zugreifen dürfen. Dies Option ist nützlich, wenn Sie die IP-Adressen der Thin-Client-Rechner, die sich mit dem Webserver verbinden dürfen, steuern können.

Klicken Sie das Kontrollkästchen **Aktivieren** an, und wenn die Parameter des Bereichs **Editieren** aktiv werden, geben Sie die IP-Adressen in die Felder **Von** und **Bis** ein, um den IP-Adress-Bereich zu benennen. Mit den Schaltflächen **Hinzufügen** und **Entfernen** verschieben Sie die IP-Adressen in die Liste **IP-Adressen-Bereich**. Wenn ein Thin Client sich mit dem Server zu verbinden versucht, prüft er, ob es eine IP-Adresse für die Thin-Client-Station in einem im Dialog IP-Sicherheit konfigurierten Bereich gibt. Wird keine gefunden, lehnt der Server den Verbindungsversuch der Thin-Client-Station ab.

Standardmäßig gilt die IP-Sicherheit nur für Thin Clients, die sich mit dem Datenserver verbinden. Jedoch können Sie IP-Sicherheit auch für eine Datenbanksynchronisation von Projekten, die auf verschiedenen Stationen laufen auf Seite 401 implementieren. Fügen Sie hierfür den folgenden Parameter in Ihre Projektdatei (**project_name.app**) ein:

[TCP]

UseWebIPSecurity=1

Weitere Einstellungen

Erweitert: Klicken Sie auf diese Schaltfläche, um einen Dialog zu öffnen, in dem die weiteren Web-Einstellungen bearbeiten werden können. In den meisten Fällen müssen diese Einstellungen nicht geändert werden. Abhängig von der in Ihrem Projekt verwendeten Architektur haben Sie jedoch die Flexibilität, weitere Einstellungen zu konfigurieren.

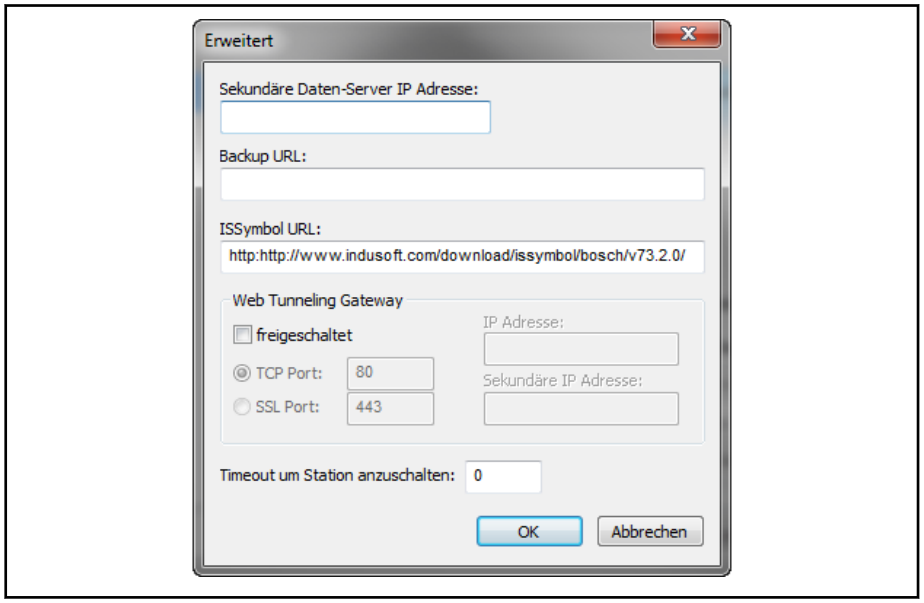


Abb.5-34: Dialog Weitere Einstellungen

- Feld **IP-Adresse des sekundären Datenservers**: Tragen Sie hier die IP-Adresse (bzw. den Host-Namen) Ihrer sekundären Datenserverstation ein. Die Datenserverstation ist der Rechner bzw. das Gerät, auf dem das TCP/IP-Server-Modul von "WinStudio" ausgeführt wird. Dieses Feld muss ausgefüllt werden, wenn Sie von Thin Clients ausgehend redundante Datenserver verwenden. Wenn der primäre Datenserver ausfällt, versucht der Thin Client automatisch, sich mit dem sekundären Datenserver zu verbinden.
- Feld **Backup-URL**: Tragen Sie die URL desjenigen Orts ein, an dem die Web-Dateien gespeichert sind (Dateien aus dem "Web"-Unterordner Ihres Projektordners). Dies URL wird eingesetzt, um die Dateien vom sekundären Webserver herunter zuladen, wenn der primäre Webserver nicht verfügbar ist.
- Feld **ISSymbol-URL**: Wenn sich der Thin Client mit dem Server verbindet, versucht er, das

  Control ISSymbol auf Seite 455 zuladen. Wenn ISSymbol auf dem lokalen Rechner (Thin Client) nicht registriert ist, versucht der Browser, es von der in diesem Feld spezifizierten URL herunter zuladen. Die Standard-URL ist eine Website, auf der "Bosch Rexroth" die aktuellste Version von ISSymbol zum Herunterladen vorhält. Möglicherweise müssen

Sie einen anderen Ort konfigurieren, vor allem wenn der Thin-Client-Rechner nicht mit dem Internet verbunden ist. Die Datei **ISSymbolVM.cab** (abgelegt im Unterverzeichnis **\BIN** von "WinStudio") muss bei der URL verfügbar sein, die in diesem Feld konfiguriert ist.

☞ Wenn die Thin-Client-Stationen keinen Zugang zum Internet haben, wird empfohlen, die Datei **ISSymbolVM.cab** auf der Webserverstation verfügbar zu machen und die URL hierfür in diesem Feld zu konfigurieren.

☞ Web-Browser, die unter dem Betriebssystem Windows CE laufen, sind nicht in der Lage, das Control ISSymbol (**ISSymbolCE.ocx**) automatisch von einer Remote-Stelle herunter zuladen. **ISSymbolCE.ocx** muss manuell in dem Windows-CE-Gerät registriert werden, das als Thin Client eingesetzt werden soll.

- **Web-Tunnel-Gateway**: Wählen Sie diese Option aus, um den Web-Tunnel-Gateway freizugeben. Je nach der Architektur auf Seite 458 Ihres Projekt müssen Sie unter Umständen die Verbindung der Thin-Client-Rechner zum Datenserver über ein Web-Tunnel-Gateway leiten.
- **TCP-Port**: Wählen Sie diese Option aus, wenn Sie HTTP mit dem Webserver von Microsoft IIS verwenden. Sie können hier den von Ihrem HTTP-Webserver verwendeten TCP-Port angeben ("80" ist der Standard-TCP-Port für das HTTP-Protokoll).
- **SSL-Port**: Wählen Sie diese Option, wenn Sie SSL (Secure Socket Layer) mit dem Webserver von Microsoft IIS verwenden. Sie können hier den von Ihrem HTTPS-Webserver verwendeten TCP-Port angeben ("443" ist der Standard-TCP-Port für das HTTPS-Protokoll).
- **IP-Adresse**: Die IP-Adresse des Webserver-Rechners, auf dem der Web-Tunnel-Gateway ausgeführt wird. Hierbei muss es sich um die IP-Adresse des Webservers handeln, der von der/den Thin-Client-Station(en) aus zugänglich ist.
- **Sekundäre IP-Adresse**: Die IP-Adresse des Webserver-Rechners, auf dem der sekundäre Web-Tunnel-Gateway ausgeführt wird. Hierbei muss es sich um die IP-Adresse des sekundären Webservers handeln, der von der/den Thin-Client-Station(en) aus zugänglich ist. Dieses Feld muss konfiguriert werden, wenn Sie redundante Webserver einsetzen.

☞ Die meisten Web-Einstellungen lassen sich dynamisch über die Funktion SetWebConfig auf Seite 709 ändern. Diese ist insbesondere dann nützlich, wenn Sie ein Projekt nur einmal anlegen und dann sicher gehen wollen, dass die Web-Einstellungen automatisch konfiguriert werden, wenn Sie Ihr Projekt auf verschiedenen Arbeitsstationen mit verschiedenen IP-Adressen laufen lassen.

5.5 Werkzeuge

WinStudio bietet verschiedene Werkzeuge an, die Sie bei der Erstellung, Konfiguration und Analyse Ihrer WinStudio-Anwendung unterstützen. Diese Werkzeuge finden Sie im Kontextmenü des Ordners **WinStudio** im Untermenü **Werkzeuge**.

Einige Funktionen sind nur in einem bestimmten Kontext verfügbar. Bitte beachten Sie die Hinweise zur Verwendung in den Beschreibungen der einzelnen Werkzeuge.

- **WinStudio-Bibliothek**: öffnet das Werkzeugfenster "WinStudio-Symbole". Hier finden Sie zahlreiche vorkonfigurierte Symbole, die Sie in Ihren WinStudio-Bildern verwenden können, siehe Symbole auf Seite 219
- **Variablen zurücksetzen**: setzt alle Variablen auf ihre ursprünglichen Werte zurück, siehe Tag-Datenbank zurücksetzen auf Seite 152
- **Variablenimport-Assistent**: öffnet den Dialog "Variablenimport-Assistent", mit dem Sie WinStudio-Variablen aus einem anderen Projekt in Ihre aktuelle Anwendung übernehmen können, siehe Importieren aus anderen Projektdatenbanken auf Seite 157
- **Datenbank-Spion**: öffnet das Werkzeugfenster "Datenbank-Spion". Dieses dient zum Anzeigen und Verändern von Variablen der Tag-Datenbank. siehe Datenbankspion auf Seite 93
- **LogWin**: öffnet das Ausgabefenster, in dem Sie Meldungen des WinStudio-Laufzeitmoduls der lokalen Station beobachten können, siehe Ausgabefenster LogWin auf Seite 93
- **Querverweise**: öffnet das Werkzeugfenster "Querverweise", in dem Sie nach der Verwendung von WinStudio-Variablen im aktuellen Projekt suchen können, siehe Querverweisliste auf Seite 96
- **Benutzer einfügen**: öffnet einen Konfigurationsdialog, mit dem Sie Benutzer erstellen können, siehe Erstellen und Konfigurieren von Benutzern auf Seite 428
- Diese Funktion ist nur beim Applikationstyp "WinStudio" verfügbar
- **Gruppe einfügen**: öffnet einen Konfigurationsdialog, mit dem Sie Benutzergruppen erstellen können, siehe Erstellen und Konfigurieren von Gruppen auf Seite 423

  Diese Funktion ist nur beim Applikationstyp "WinStudio" verfügbar.
- **Sicherheitseinstellungen**: öffnet den Dialog "Sicherheitseinstellungen". In diesem Dialog können Sie die WinStudio-Benutzerverwaltung aktivieren und konfigurieren, siehe Verwenden des Konfigurationsassistenten für das Sicherheitssystem auf Seite 410

  Diese Funktion ist nur beim Applikationstyp "WinStudio" verfügbar
- **Benutzer an-/abmelden**: öffnet den Anmeldedialog, siehe An- / Abmelden auf Seite 432
- **Remote-Datenbankspion**: öffnet den Eingabedialog, mit dem Sie sich mit einem anderen Gerät verbinden und dessen WinStudio-Variablen ansehen und verändern können, siehe Remote-Datenbankspion auf Seite 447
- **Remote LogWin**: öffnet den Eingabedialog, mit dem Sie sich mit einem anderen Gerät verbinden und dessen Meldungen der WinStudio-Laufzeitumgebung ansehen können, siehe Remote LogWin auf Seite 448
- **Status**: öffnet den Dialog "Ausführungs-Tasks". Dieser Dialog zeigt eine Übersicht aller Tasks, deren Startmodus und den aktuellen Ausführungsstatus
- **Ereignis-Logger**: öffnet den Dialog "Ereignis-Logger". In diesem Dialog können Sie die Protokollierung von WinStudio-Meldungen aktivieren und konfigurieren, siehe Ereignisse und Ereignisprotokolle auf Seite 285

- **Projekt laden**: öffnet den Windows-Dateibrowser. Sie können hier ein bereits vorhandenes WinStudio-Projekt auswählen und in das aktuelle Projekt übernehmen

HINWEIS	**Wenn Sie ein anderes WinStudio-Projekt laden, werden zunächst alle Dateien des aktuellen WinStudio-Projekts gelöscht.**

Wenn Sie sich nicht sicher sind, ob Sie tatsächlich alle vorhandenen Dateien löschen möchten, brechen Sie den Vorgang mit der Schaltfläche "Abbrechen" ab.

- **Web-Einstellungen**: öffnet den Konfigurationsdialog "Web-Einstellungen". In diesem Dialog können Sie den Web-Server konfigurieren, wenn Sie Ihr Projekt als Web-Anwendung ausführen möchten, siehe Registerkarte Web auf Seite 86

  Diese Funktion ist nur beim Applikationstyp "WinStudio" verfügbar.
- **Alle Dokumente als HTML speichern**: Erzeugt oder für alle WinStudio-Elemente eine HTML-Datei oder aktualisiert diese, siehe Erstellen eines einfachen Thin-Client-Programms auf Seite 451

5.6 Bild-/Arbeitsblatt-Editor

Erstellen und bearbeiten Sie mit Hilfe des Bild-Editors beliebige Bilder und Arbeitsblätter für Ihre Projekte. Sie können Informationen mit der Maus oder der Tastatur eingeben, Steuerungsdaten an Prozesse ausgeben und Bilder automatisch auf der Grundlage von in Ihren Prozessen eingegebenen Daten aktualisieren und animieren.

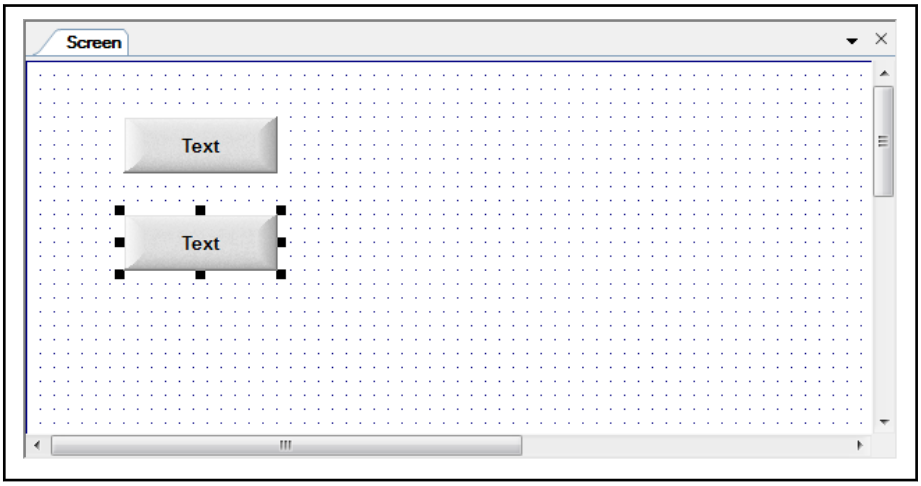


Abb.5-35: Bild-/Arbeitsblatt-Editor

Weitere Merkmale des Bild-Editors sind unter anderem:

- Point-and-Click- bzw. Drag-and-Drop-Unterstützung für eine einfache Bilderstellung
- Möglichkeit zur Gruppierung von Objekten, um deren Anordnung beizubehalten
- Bearbeitung von Gruppenelementen, ohne dass deren Gruppierung aufgelöst werden muss
- Handhaben von Bitmap-Objekten und Hintergrund-Bitmaps

5.7 Datenbankspion

Das Fenster "Datenbankspion" ist ein Fehleranalysewerkzeug für die Anzeige und Änderung von Tag-Werten sowie für Ausführung und Test von Funktionen und mathematischen Ausdrücken.



Abb.5-36: Beispiel für ein Datenbankspion-Fenster

Das Fenster zeigt die folgenden Informationen an:

- **Name**: Angeben einer Anwendungsvariablen, einer internen Variablen oder eines Ausdrucks, der überwacht werden soll.
- **Wert**: Anzeige des Werts, der vom Tag/Ausdruck zurückgegeben wird.
- **Qualität**: Anzeige der Qualität (GUT oder SCHLECHT) des Werts, der vom Tag/Ausdruck zurückgegeben wird.
- **Stetig**: Wählen Sie diese Option, wenn das Projekt den Tag/Ausdruck fortlaufend auswerten soll.

Um eine Variable zu überwachen, tragen Sie in der Spalte deren Namen ein oder klicken Sie in ein leeres Feld in der Spalte "Name" und öffnen Sie das Kontextmenü. Über die Funktion "Variable oder Ausdruck einfügen" können Sie alternativ den Browserdialog öffnen und eine Variable auswählen.



Der Datenbankspion ist andockfähig, d.h. er kann an eine andere Stelle in der Entwicklungsumgebung verschoben werden.

Klicken Sie auf die Titelleiste, und ziehen Sie das Fenster an eine neue Position.

Lassen Sie die Maustaste los, um das Fenster an seinem neuen Platz zu befestigen bzw. anzudocken.

5.8 Ausgabefenster LogWin

Das Fenster "WinStudio Log" können Sie verwenden, um Fehleranalysemeldungen anzuzeigen, die während der Projektlaufzeit erzeugt werden. Das Fenster zeigt OPC-, DDE- und TCP/IP-Transaktionen, Modulaktivierungen, Trace-Meldungen usw. an.

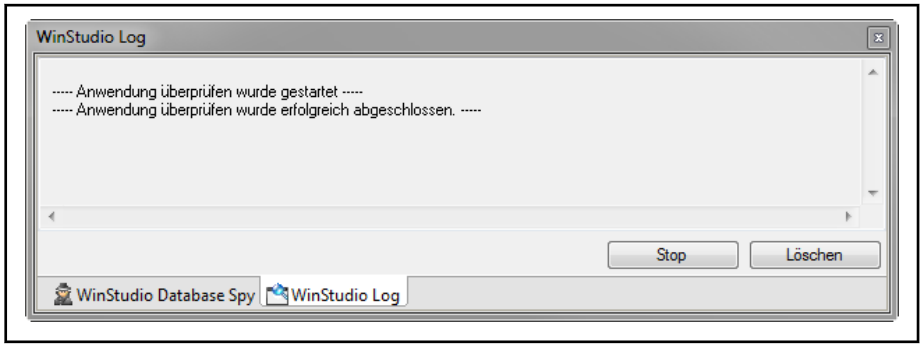


Abb.5-37: Beispiel-Fenster Ausgabe

Das Fenster "WinStudio Log" ist andockfähig, d.h. es kann an eine andere Stelle in der Entwicklungsumgebung verschoben werden. Klicken Sie dazu auf die Titelleiste und ziehen Sie das Fenster an eine neue Position. Lassen Sie die Maustaste los, um das Fenster an seinem neuen Platz zu befestigen bzw. "anzudocken".

Konfigurieren des Protokolls zur Anzeige zusätzlicher Informationen

Standardmäßig zeigt das Protokoll nur Fehleranalyse- und Fehlermeldungen an – also Meldungen, aus denen hervorgeht, dass Ihr Projekt nicht korrekt ausgeführt wird. Würde das Protokoll **alle** von "WinStudio" erzeugten Meldungen anzeigen, wäre die Ausgabe schnell unübersichtlich.

Um die im Protokoll angezeigten Meldungen zu konfigurieren gehen Sie folgendermaßen vor:

1. Führen Sie an einer beliebigen Stelle im Fenster "WinStudio Log" einen Rechtsklick aus und klicken Sie dann im Kontextmenü auf **Einstellungen**. Der Dialog "Logging-Eigenschaften" wird angezeigt:
2. In der Registerkarte "Log-Optionen" des Dialogs wählen Sie die Meldungsarten aus, die im Protokoll angezeigt werden sollen.

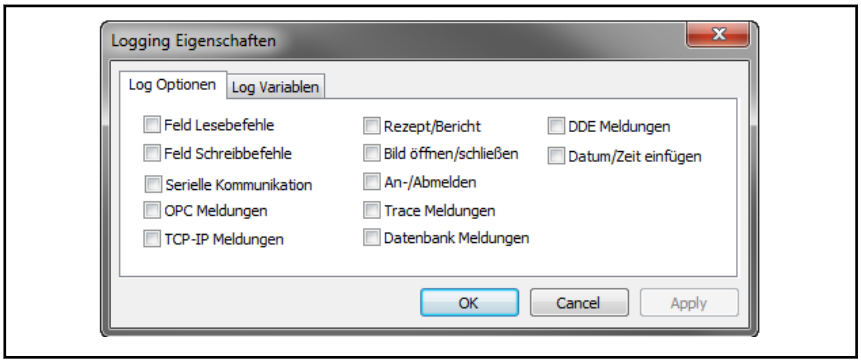


Abb.5-38: Protokoll-Einstellungen – Registerkarte Optionen

Optionen	Beschreibung
Feld-Lesebefehle	Zeigt alle Feld-Lesebefehle, die an angeschlossene Geräte geschickt werden.
Feld-Schreibbefehle	Zeigt alle Feld-Schreibbefehle, die an angeschlossene Geräte geschickt werden.
Serielle Kommunikation	Zeigt von konfigurierten Gerätetreibern auf Seite 379 erzeugte Meldungen an.
OPC-Meldungen	Zeigt Meldungen der OPC-Kommunikation auf Seite 390 an.
TCP/IP-Meldungen	Zeigt Meldungen der TCP/IP-Kommunikation auf Seite 401 an.
Rezept/Bericht	Zeigt Meldungen der Tasks für Rezepturen auf Seite 361 und Berichte auf Seite 363 an.

Optionen	Beschreibung
Öffnen/Schließen anzeigen	Zeigt detaillierte Informationen an, wenn ein Bild auf Seite 170 geöffnet oder geschlossen wird: **Festplatten-Ladezeit:** Die Zeit, um die Bilddatei von der Festplatte in den Speicher zu laden. **Öffnungszeit:** Zeit zum Laden des WinStudio-Bild, einschließlich der im Bild verwendeten Initialisierungs-Tags und dem Ausführen von ggf. vorhandenen „OnOpen“-Skripten oder -Funktionen. **Gesamt-Ladezeit:** Gesamtzeit zum Laden des Bilds (beinhaltet die obige Festplatten- und die Öffnungsladezeit). **"Erstes-Zeichnen"-Zeit:** Zeit zum erstmaligen Zeichnen der Bildobjekte. **"Erstes-Onwhile"-Zeit:** Zeit für das erstmalige Ausführen von ggf. vorhandenen „OnWhile“-Scripten oder -Funktionen. **Gesamt-Öffnungszeit:** Zeit insgesamt zum Öffnen des Bildes (beinhaltet die obige Erstes-Zeichnen- und Erstes-OnWhile-Zeit). **Schließzeit:** Zeit zum Schließen des Bildes, einschließlich der im Bild verwendeten Abschluss-Tags und dem Ausführen von ggf. vorhandenen „OnClose“-Scripten oder -Funktionen. **Gesamt-Schließzeit:** Gesamtzeit zum Schließen des Bildes, einschließlich der Zeit zum Schreiben der Bilddatei auf die Festplatte. Diese Informationen lassen sich zum Analysieren der Laufzeitleistung auf schwach ausgestatteten Endsystemen verwenden. Wenn ein bestimmter Schritt beim Öffnen oder Schließen ungewöhnlich lange braucht, kann er gefunden und neu konzipiert werden.
Anmelden/Abmelden	Anzeigen einer Meldung, wenn sich ein Benutzer an- oder abmeldet. (Weitere Informationen erhalten Sie in Sicherheit auf Seite 407.)
Trace-Meldungen	Zeigt Meldungen der Funktion **Trace()** an. Diese Funktion wird zum Erzeugen kundenspezifischer Meldungen aus Ihrem Projekt verwendet, um die Ausführung der Anwendung nachzuverfolgen.
Datenbankmeldungen	Zeigt Meldungen der ODBC- auf Seite 364 und ADO.NET- auf Seite 371 Datenbankschnittstellen an.
DDE-Meldungen	Zeigt Meldungen der DDE-Kommunikation auf Seite 403 an.
Datum/Uhrzeit einfügen	Jede Meldung mit einem Zeitstempel versehen.

Abb.5-1: Protokoll-Einstellungen – Registerkarte Optionen

3. Klicken Sie in der Registerkarte "Log Variablen" des Dialogs auf **Hinzufügen**, um nach Projekt-Tags auf Seite 116 zu suchen.

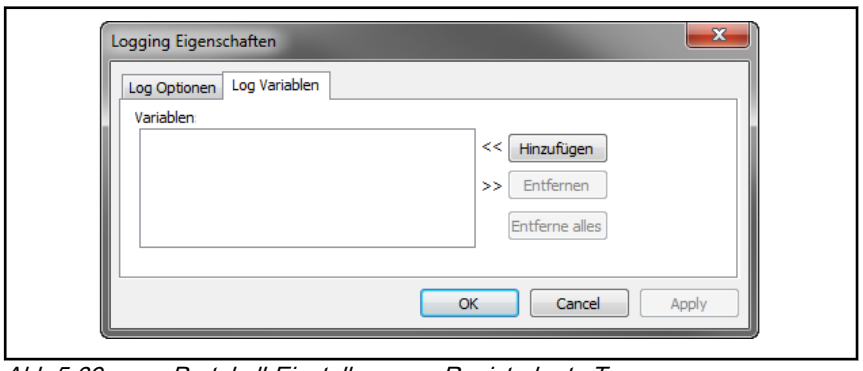


Abb.5-39: Protokoll-Einstellungen – Registerkarte Tags

Der Protokollbereich des LogWin-Fensters zeigt eine Meldung an, wenn sich der Wert eines ausgewählten Tags ändert.

4. Klicken Sie auf **OK**, um Ihre Einstellungen zu speichern und den Dialog "Logging-Eigenschaften" zu schließen.

5.9 Querverweisliste

Wenn Sie herausfinden möchten, in welchen Arbeitsblättern eine Variable verwendet wird, öffnen Sie das Werkzeugfenster **WinStudio ▸ Werkzeuge ▸ Querverweise**.

Um alle Arbeitsblätter zu durchsuchen geben Sie im Textfeld **Tag-Name** den Namen der zu suchenden Variablen ein. Alternativ können Sie den Variablenbrowser über die Schaltfläche  öffnen und eine Variable auswählen. Nach Bestätigung des Variablennamens mit der Eingabetaste startet der Suchvorgang. Im Protokollfenster werden anschließend alle Treffer aufgelistet (siehe auch Ausgabe auf Seite 93).

Wenn Sie auf einen Treffer mit der Maus einen Doppelklick ausführen, wird das zugehörige Arbeitsblatt im Editor geöffnet und das Element, das die gesuchte Variable verwendet, angewählt.

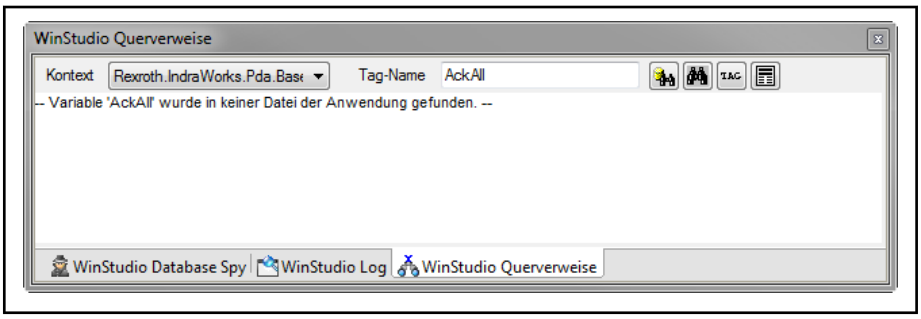


Abb.5-40: Ergebnisse Querverweissuche

Wenn Ihr Projekt mehrere Visualisierungsgeräte enthält, müssen Sie in der Auswahlliste "Kontext" auswählen, in welchem Visualisierungsgerät WinStudio suchen soll.

5.10 Statuszeile

Die Statuszeile befindet sich am unteren Rand der Entwicklungsumgebung. Sie zeigt Informationen über das aktive Bild z. B. die Anzahl der bereits verwendeten Tags an.

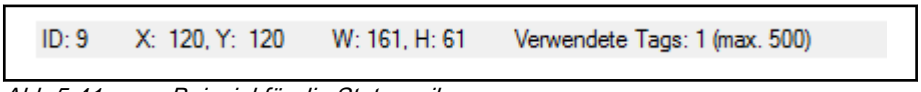


Abb.5-41: Beispiel für die Statuszeile

Die Felder der Statuszeile sind (von links nach rechts) in der nachfolgenden Tabelle beschrieben:

Feld	Beschreibung
Objekt-ID	Kennnummer eines ausgewählten Bildobjekts
Cursorposition	Position des Cursors innerhalb des aktiven Bildes bzw. des aktiven Arbeitsblatts. Wenn es sich um ein Bild handelt, wird die Lage des Mauszeigers als Koordinatenpaar X, Y angegeben, wobei X die Anzahl der Pixel vom linken Bildrand bis zum Cursor angibt und Y die Anzahl der Pixel vom oberen Bildrand zum Cursor. Wenn es sich um ein Arbeitsblatt handelt, dann wird die Lage des Text-Cursors als Zeile und Spalte angegeben.
Objektgröße	Größe (in Pixel) eines ausgewählten Bildobjekts, wobei W die Breite und H die Höhe ist
Verwendete Tags	Anzahl der bisher im Projekt verwendeten Tags und die für den aktuellen Projekttyp maximal mögliche Variablenanzahl

Abb.5-2: Felder der Statuszeile

6 WinStudio-Projekte

6.1 Anlegen eines neuen Projekts

6.1.1 Anlegen eines neuen Projekts, Allgemeines

Ein WinStudio-Projekt ist stets einem Visualisierungsgerät in einem IndraWorks-Projekt zugeordnet. Um ein neues Visualisierungsgerät zu einem geöffneten IndraWorks-Projekt hinzuzufügen gehen Sie folgendermaßen vor:

1. Fügen Sie ein Visualisierungsgerät zu Ihrem IndraWorks-Projekt hinzu. Wählen Sie dazu eine der folgenden Vorgehensweisen:
   - Öffnen Sie das Fenster **Bibliothek** (im Menü "Ansicht"), und wählen Sie im Register **Visualisierung** den passenden Visualisierungstyp aus, indem Sie diesen mit der Maus anklicken. Ziehen Sie anschließend den gewählten Visualisierungstyp mit gedrückter Maustaste auf den Projektordner im Projektexplorer und lassen Sie dort die Maustaste wieder los.
   - Öffnen Sie das Kontextmenü des Projektordners und wählen Sie im Untermenü **Hinzufügen** den zu Ihrem Visualisierungsgerät passenden Visualisierungstyp aus.

   In beiden Fällen öffnet sich der HMI-Gerätewizard.
2. Wählen Sie im HMI-Wizard in der Auswahlliste "Gerätetyp" den passenden Eintrag aus.
3. Geben Sie im Eingabefeld "Name" einen Text ein, mit dem das Gerät im Projektexplorer angezeigt werden soll.
4. Geben Sie im Feld "IP-Adresse" den Computernamen oder die IP-Adresse des Geräts an.
5. Wählen Sie aus der Auswahlliste "Applikationstyp" einen der folgenden Einträge aus:
   - **WinStudio-Anwendung**: Wählen Sie diesen Applikationstyp, wenn Sie eine eigenständige Applikation erstellen möchten, bei der keine weiteren Visualisierungsobjekte wie F-, M- und OP-Panels sowie verschiedene Standardanwendungen (Diagnose, Bedienbilder, etc) benötigt werden. Die Verwendung der WinStudio-internen Benutzerverwaltung sowie die Unterstützung von Web-Clients ist nur bei diesem Applikationstyp möglich.

     Bei diesem Applikationstyp wird das Projekt in der WinStudio-Laufzeitumgebung ausgeführt.
   - **HMI-Standardanwendung**: Wählen Sie diesen Applikationstyp, wenn Sie zusätzlich zu den WinStudio-Objekten auch Panels und Standardanwendungen verwenden möchten. WinStudio wird innerhalb der IndraWorks Runtime ausgeführt.
   - **MTX-Standardanwendung**: Wählen Sie diesen Applikationstyp, wenn Sie zusätzlich zu den WinStudio-Objekten auch Panels und Standardanwendungen in Verbindung mit einer MTX-Steuerung verwenden möchten. WinStudio wird innerhalb der IndraWorks Runtime ausgeführt.
6. Wählen Sie unter "Geräteversion" aus, welche Version die Laufzeitumgebung hat, die auf dem Visualisierungsgerät installiert ist.
7. Bestätigen Sie die Einstellungen mit der Schaltfläche "Fertigstellen". Der HMI-Wizard erstellt einen neuen Ordner für das Visualisierungsgerät im

Projektexplorer. Sie finden den WinStudio-Ordner unterhalb des Visualisierungsordner:

- Beim Typ **WinStudio-Anwendung** unterhalb des Visualisierungsordners.
- Beim Typ **HMI-Standardanwendung** und beim Typ **MTX-Standardanwendung** unterhalb des Ordners "Screen Application".

Beim Anlegen eines Visualisierungsgeräts werden folgende Einstellungen entsprechend des Geräte- und Applikationstyps angepasst:

- **Zielplattform**: Bei VEx-Geräten mit Windows-CE-Betriebssystem ist "WinStudio CE Lite", bei allen anderen "WinStudio Lite"ausgewählt
- **Auflösung**: Bei WinStudio-Anwendungen entspricht die Auflösung der Größe des Anwendungsfensters zur Laufzeit. Hierbei handelt es sich in der Regel um die Bildschirmauflösung des Zielgeräts (z.B. 1024x768). Bei HMI- und MTX-Applikationen weicht diese Einstellung von Bildschirmauflösung ab, da die Laufzeitumgebung zusätzlich Panels und Header darstellt. Der effektive Bereich für WinStudio-Bilder ist somit geringer.
- **Startmodus Viewer-Task**: Bei WinStudio-Anwendungen ist der Wert auf "Automatisch" gesetzt. Bei HMI- und MTX-Applikationen wird der Viewer-Task von der IndraWorks-Laufzeitumgebung gestartet, d.h. der Wert muss auf "Manuell" gesetzt sein.

6.1.2 Zielplattformen und Produkttypen

Durch die Zielplattform und den Produkttyp werden wichtige Aspekte des Projekts bestimmt, wie z.B. wie viele Tags vom Projekt unterstützt werden und welche Merkmale während der Laufzeit verwendet werden können.

Zielplattform

Die Zielplattform ist im Allgemeinen entweder Windows oder Windows Embedded.

Windows

Ein Rechner, auf dem eines der folgenden Betriebssysteme ausgeführt wird:

- Microsoft Windows XP mit Service Pack 3 oder höher
- Microsoft Windows Vista mit Service Pack 1 oder höher
- Microsoft Windows 7, alle Versionen
- Microsoft Windows Server 2003 mit Service Pack 2 oder höher
- Microsoft Windows Server 2008, alle Versionen
- Microsoft Windows XP Embedded Service Pack 3
- Microsoft Windows Embedded Standard 7 (2009)

Auf dem Rechner muss die Software "Rexroth WinStudio" vollständig installiert sein, selbst wenn sie nicht zur Projektentwicklung eingesetzt wird, weil die vollständige Software die zum Beherbergen des Projekt-Laufzeitobjekts auf Windows erforderlichen Module beinhaltet. Ebenso muss der Rechner mit einem passenden Laufzeit-Lizenzschlüssel ausgestattet sein. Weitere Informationen erhalten Sie in Installation auf Seite 39.

Alle Projektmerkmale stehen zur Verfügung, wenn **Windows** die Zielplattform ist.

Windows Embedded

Ein Embedded-Gerät, auf dem eines der folgenden Betriebssysteme ausgeführt wird:

- Microsoft Windows XP Embedded Service Pack 3

- Microsoft Windows Embedded Standard 7 (2009)
- Microsoft Windows Embedded Compact (früher Windows CE bzw. Windows Mobile genannt), Version 5.0 oder höher

Auf dem Gerät muss entweder EmbeddedView (bei Windows XP Embedded und Windows Embedded Standard) oder CE-View (bei Windows Embedded Compact) installiert sein. Ebenso muss das Gerät mit einem passenden Laufzeit-Lizenzschlüssel ausgestattet sein. Weitere Informationen erhalten Sie in Installation auf Seite 39.

Bitte beachten Sie: Wenn **Windows Embedded** die Zielplattform ist, stehen bestimmte Projektfunktionen nicht zur Verfügung:

- alle DDE-Produktfunktionen, insbesondere der DDE-Server-Task, der DDE-Client-Task und -Arbeitsblätter.
- ODBC-Laufzeit-Task, -Arbeitsblätter und -Funktionen. (Die ADO-Datenbankschnittstelle ist jedoch verfügbar.)
- OPC-HDA-Server-Task.
- Weniger bedeutende Grafikfunktionen, z.B. die Unterstützung anderer Grafikformate als BMP und JPG und Fülleffekte für andere Formen als Rechtecke.
- Diverse andere eingebaute Funktionen, wie in der Dokumentation der internen Skriptsprache beschrieben.



In den meisten Fällen gilt: Wenn Sie möchten, dass Ihr Host-Projekt-Laufzeitobjekt auf einem Gerät mit XP Embedded oder Windows Embedded Standard beherbergt wird und wenn Sie nicht vorhaben, auf diesem Gerät Projektentwicklung zu betreiben, dann sollten Sie auf diesem Gerät nur EmbeddedView installieren und als Zielplattform **WinStudio Embedded** auswählen. EmbeddedView hat einen kleineren Ressourcenbedarf als die vollständige Software "Rexroth WinStudio" und ist für Remote-Installation und -Verwaltung geeignet.

Wenn Sie jedoch Funktionen verwenden möchten, die mit der Zielplattform **WinStudio Embedded** nicht verfügbar sind, müssen Sie auf dem Gerät die vollständige Software "Rexroth WinStudio" installieren und als Zielplattform **WinStudio** auswählen.

Die vollständige Software "Rexroth WinStudio" lässt sich auf einem Gerät mit Windows Embedded Compact unter keinen Umständen installieren.

Produkttyp Der Produkttyp bestimmt hauptsächlich, wie viele Tags Sie im Projekt anlegen können, einschließlich gemeinsamer Tags und aus anderen Systemen importierter Tags. Der Projekttyp "WinStudio" kann wesentlich mehr Tags unterstützen als "WinStudio Embedded".

6.1.3 Zielsystem eines bestehenden Projekts ändern

Das Zielsystem wird beim Hinzufügen eines Visualisierungsgeräts entsprechend des Gerätetyp vorkonfiguriert. In folgenden Fällen ist es notwendig, die Standardeinstellung zu ändern:

- Sie benötigen für Ihre Anwendung mehr als 500 Tags.
- Sie möchten eine oder mehrere der folgenden Funktionen nutzen: Trends, Rezepturen, Reporte, Externe Datenbanken.

- Sie verwenden den Applikationstyp "WinStudio-Anwendung" und möchten die WinStudio-interne Benutzerverwaltung oder die Web-Funktionalität nutzen.

Falls dies für Ihr Projekt zutrifft, müssen Sie das Zielsystem auf WinStudio 1.5k (für 1500 Variablen) oder höher ändern. Alle Zielsysteme ab "WinStudio 1.5k" ermöglichen die Nutzung der oben genannten Funktionen.

Um das Zielsystem "WinStudio 1.5k" oder höher auswählen zu können benötigen Sie eine entsprechende WinStudio-Engineering-Lizenz. Andernfalls steht die Auswahl des Zielsystems nicht zur Verfügung.

Wenn Sie das Zielsystem ändern möchten und über eine ausreichende WinStudio-Engineering-Lizenz verfügen, gehen Sie folgendermaßen vor:

1. Öffnen Sie den WinStudio-Eigenschaftendialog.
2. Klicken Sie auf das Register "Optionen"
3. Öffnen Sie die Auswahlliste im Bereich **Zielsystem**. Eine Liste verfügbarer Produkttypen wird angezeigt. Wählen Sie den neuen Produkttyp Ihres Projekts aus.

   Nähere Informationen entnehmen Sie bitte Zielplattformen und Produktarten auf Seite 100.
4. Schließen Sie den Dialog mit der Schaltfläche "OK".

Das Projekt wird in den ausgewählten Produkttyp umgewandelt.



Wenn Sie von einer WinStudio-Zielplattform auf eine WinStudio-Embedded-Zielplattform gewechselt haben, stehen einige Produktfunktionen nicht mehr zur Verfügung.

6.2 E-Mail-Einstellungen konfigurieren

Über einige Merkmale, wie z.B. Alarme und bestimmte Funktionen, können E-Mails an ausgewählte Empfänger versendet werden. Um diese Funktionen einsetzen zu können, müssen Sie die E-Mail-Einstellungen Ihres Projekts konfigurieren.

Die E-Mail-Einstellungen können jederzeit während der Laufzeit durch Aufruf der Funktion "CnfEmail" konfiguriert werden. Sie können aber auch Standardeinstellungen konfigurieren, die beim erstmaligen Ausführen des Projekts automatisch verwendet werden und bei Bedarf während der Laufzeit wiederhergestellt werden, wodurch alle über einen Aufruf der Funktion "CnfEmail" erfolgten Änderungen überschrieben werden.

1. Öffnen Sie das Kontextmenü des Ordners "WinStudio" und wählen Sie den Eintrag **Werkzeuge ▸ E-Mail-/FTP-Einstellungen**. Der Dialog "E-Mail- und FTP-Konfiguration" wird angezeigt.
2. Klicken Sie auf die Registerkarte **E-Mail-Einstellungen**.

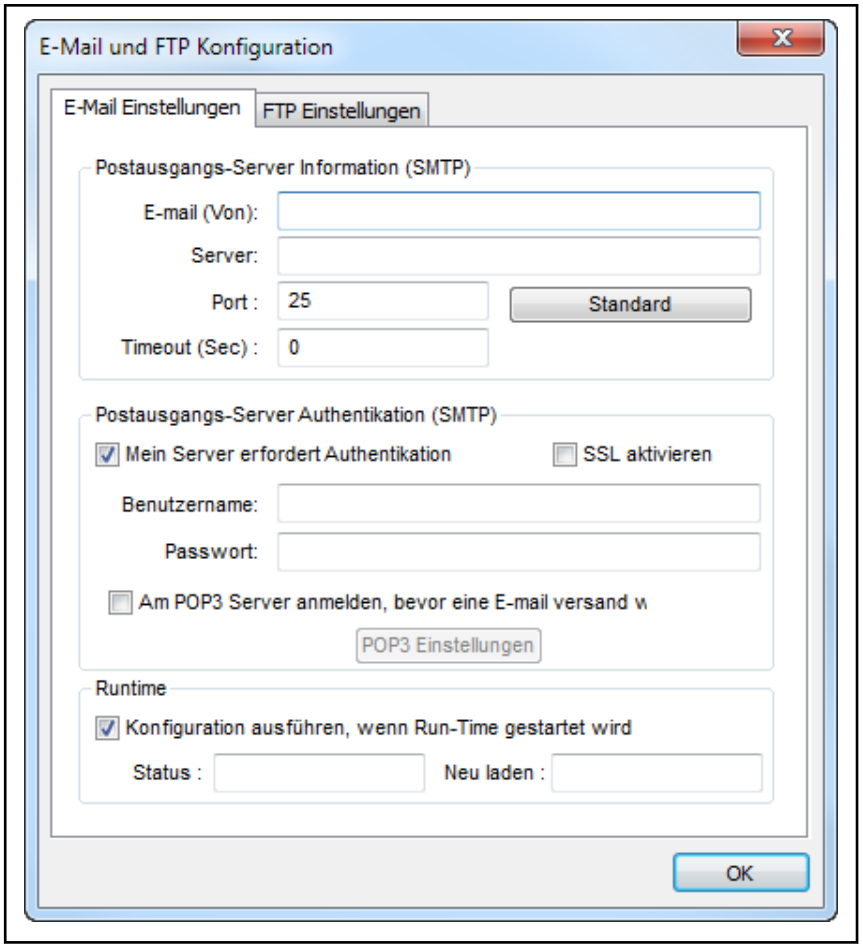


Abb.6-1: Registerkarte E-Mail-Einstellungen

3. Tragen Sie Ihre E-Mail-Adresse in das Feld **E-Mail (Absender)** ein.
4. In den Feldern **Server** und **Port** tragen Sie die Server-Adresse und Port-Nummer für Ihren ausgehenden Mail-Server ein. Der Standard-Port für SMTP ist 25, aber dies hängt von der Konfiguration Ihres Servers und Netzwerks ab. Bitte wenden Sie sich hierzu an Ihren E-Mail-Provider.
5. Wenn Ihr Ausgangs-Mailserver eine Authentifizierung erfordert, wählen Sie **Mein Server erfordert eine Authentifizierung** aus. Wenn die Authentifizierung zudem verschlüsselt sein muss, wählen Sie **SSL aktivieren** aus. Tragen Sie dann Ihre Zugangsdaten in die Felder **Benutzername** und **Passwort** ein.

   Die meisten Ausgangsserver erfordern Authentifizierung, um Spam und anderen Missbrauch durch unbekannte Benutzer abzuwehren.

☞ Verschlüsselung mittels SSL ist nicht für Projekte verfügbar, die auf Windows-CE-Zielsystemen laufen.

6. Im Feld **Status** geben Sie den Namen eines Tags (vom Typ Integer) ein, der Status-Codes erhalten soll, wenn das Projekt E-Mail verschickt.
7. In das Feld **Neu laden** tragen Sie einen Tag/Ausdruck ein. Wenn sich der Wert dieses Tags/Ausdrucks ändert, lädt das Projekt diese Standard-E-Mail-Einstellungen neu.
8. Klicken Sie auf **OK**, um Ihre Einstellungen zu speichern und den Dialog zu schließen.

6.3 FTP-Einstellungen konfigurieren

Mit Hilfe von internen WinStudio-Funktionen lassen sich Dateien zwischen Rechnern über FTP übertragen. Um diese Funktionen einsetzen zu können, müssen Sie die FTP-Einstellungen Ihres Projekts konfigurieren.

Die FTP-Einstellungen können jederzeit während der Laufzeit durch Aufruf der Funktion CnfFTP auf Seite 767 konfiguriert werden. Sie können aber auch Standardeinstellungen konfigurieren, die beim erstmaligen Ausführen des Projekts automatisch verwendet werden und bei Bedarf während der Laufzeit wiederhergestellt werden, wodurch alle über einen Aufruf der Funktion **CnfFTP** erfolgten Änderungen überschrieben werden.

1. Öffnen Sie das Kontextmenü des Ordners "WinStudio" und wählen Sie den Eintrag **Werkzeuge** ▸ **E-Mail-/FTP-Einstellungen**. Der Dialog "E-Mail- und FTP-Konfiguration" wird angezeigt.
2. Klicken Sie auf die Registerkarte **FTP-Einstellungen**.

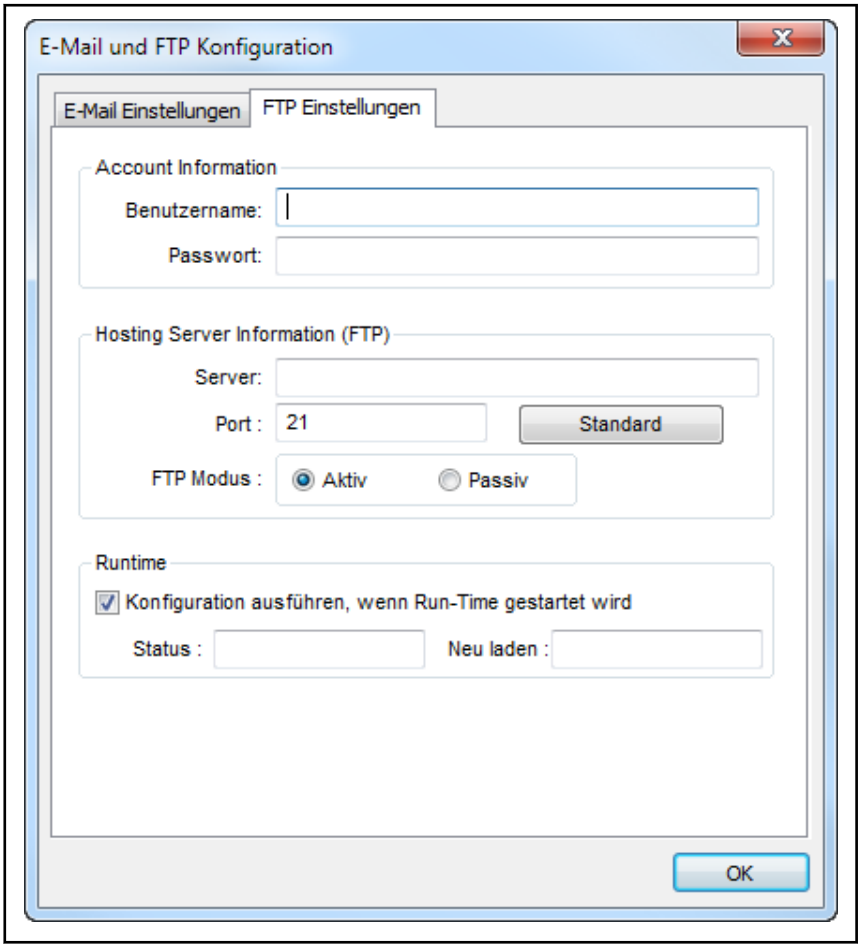


Abb.6-2: Registerkarte FTP

3. Tragen Sie Ihre Zugangsdaten für den FTP-Server in die Felder **Benutzername** und **Passwort** ein.
4. In den Feldern **Server** und **Port** tragen Sie die Server-Adresse und Port-Nummer ein. Der Standard-Port für FTP ist 21, aber dies hängt von der Konfiguration Ihres Servers und Netzes ab. Bitte wenden Sie sich hierzu an Ihren Server-Administrator.
5. Je nach der Konfiguration Ihres Servers wählen Sie den Modus **Aktiv** oder **Passiv**. Mit dem passiven FTP-Modus können einige Netzwerk-Fi-

rewalls umgangen werden. Bitte konsultieren Sie auch hierzu Ihren Server-Administrator.

6. Im Feld **Status** geben Sie den Namen eines Tags (vom Typ Integer) ein, der Status-Codes erhalten soll, wenn das Projekt eine Datei überträgt.
7. In das Feld **Neu laden** tragen Sie einen Tag/Ausdruck ein. Wenn sich der Wert dieses Tags/Ausdrucks ändert, lädt das Projekt diese Standard-FTP-Einstellungen neu.
8. Klicken Sie auf **OK**, um Ihre Einstellungen zu speichern und den Dialog zu schließen.

6.4 Projektausführung unter Windows-Systemdiensten

Ihr "WinStudio"-Projekt lässt sich so konfigurieren, dass es als "Windows-Systemdienst" ausgeführt wird. Mit Microsoft-Windows-Systemdiensten – früher als NT-Dienste bekannt – können Sie lang laufende Programme erstellen, die in ihren eigenen Windows-Sitzungen ausgeführt werden. Diese Prozesse lassen sich automatisch beim Hochfahren des Rechners starten, sie lassen sich anhalten und wieder starten und sie erscheinen auf keiner Benutzeroberfläche. Aufgrund dieser Merkmale sind Dienste ideal für den Einsatz auf einem Server oder immer dann, wenn Sie die Funktionalität mit einer langen Laufzeit benötigen und die Prozesse mit anderen Benutzern, die auf dem gleichen Rechner arbeiten, nicht in Konflikt kommen sollen. Ebenso können Sie Dienste im sicheren Umfeld eines bestimmten Benutzerkontos ausführen lassen, das sich vom Konto des angemeldeten Benutzers bzw. vom Standard-Benutzerkonto unterscheidet. Mehr Informationen zu Diensten finden Sie in der Microsoft Developer Network (MSDN) Bibliothek.

Warum sollte ein Projekt unter Windows-Systemdiensten ausgeführt werden?

- um sicherzugehen, dass Ihr Projekt mit den jeweils benötigten Systemrechten ausgeführt wird, ungeachtet der Rechte des aktuell bei Windows angemeldeten Benutzers
- um zu verhindern, dass der Benutzer in Ihr Projekt eingreifen kann, während es ausgeführt wird oder
- um Ihr Projekt weiter auszuführen, wenn kein Benutzer angemeldet ist

Anlegen und Konfigurieren des Windows-Systemdienstes



Alle unten beschriebenen Verfahren wurden unter Verwendung von Microsoft Windows XP Service Pack 3 getestet. Um die Aktionen auszuführen, müssen Sie zudem als Benutzer mit Administratorrechten angemeldet sein und Sie sollten wissen, wie man die "Computerverwaltungs-Konsole" verwendet. (Zugriff auf diese Konsole durch Rechtsklick auf das Symbol **Arbeitsplatz** und Auswahl von **Verwalten** aus dem Kontextmenü.)

Es gibt zwei Möglichkeiten, den Windows-Systemdienst für Ihr Projekt anzulegen und zu konfigurieren: Sie können das Werkzeug" Dienst-Konfiguration" in WinStudio selbst verwenden oder Sie können das zusammen mit installierte Befehlszeilen-Hilfsprogramm verwenden.

Werkzeug Dienste-Konfiguration

Sie können einen neuen Dienst aus der Entwicklungsumgebung heraus konfigurieren und ausführen, indem Sie im Kontextmenü von WinStudio den Eintrag **Dienste-Konfiguration** wählen. Dieser öffnet den Dialog "Dienst-Konfiguration":

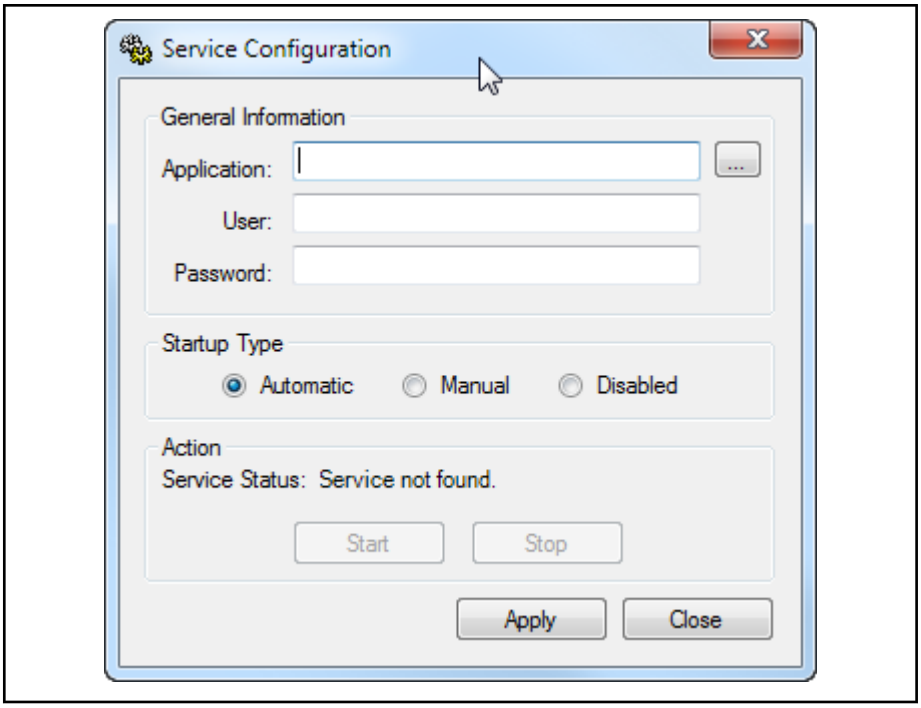


Abb.6-3: Dialog Dienst-Konfiguration

Feld Projekt

Der Speicherort der Projektdatei (**projektname.app**), die der Dienst laden und ausführen soll, wenn er gestartet wird. Es muss ein vollständiger Pfad eingegeben werden.

Feld Benutzer

Das Windows-Benutzerkonto, unter dem der Dienst ausgeführt werden soll. Diese Einstellung ist optional; wenn sie nicht verwendet wird, wird der Dienst unter Lokales System ausgeführt.

Vermeiden Sie möglichst, den Dienst unter Lokales System laufen zu lassen. Dieses Konto verfügt über zu viele Rechte für das Dateisystem und über zu wenig Rechte, um die OPC-Client -und -Server-Module auf Seite 390 ordnungsgemäß ausführen zu lassen. Die beste Alternative ist, einen Benutzer speziell dafür anzulegen, "WinStudio" auszuführen, und dessen Rechte dann entsprechend den Bedürfnissen Ihres Projekts zu konfigurieren. Nähere Informationen hierzu finden Sie untenstehend unter „Benutzerrechte konfigurieren“.

Feld Passwort

Das Passwort für das angegebene Benutzerkonto. Diese Einstellung ist optional; sie wird nicht benötigt, wenn kein Benutzer angegeben ist oder wenn der angegebene Benutzer kein Passwort hat.

Feld Start-Typ

Wie der Windows-Dienst starten soll. Die folgenden Optionen sind verfügbar:

- **Automatisch**: Der Dienst startet automatisch beim Hochfahren des Rechners.
- **Manuell**: Der Dienst kann manuell in der "Computerverwaltungs-Konsole" oder durch Anklicken von **Start** wie unten beschrieben gestartet werden.
- **Deaktiviert**: Der Dienst wird angelegt und dann deaktiviert. Er kann erst ausgeführt werden, wenn ein Administrator-Benutzer ihn in der "Computerverwaltung"-Konsole aktiviert hat.

Bereich Aktion

Starten oder Anhalten des Dienstes. Bitte beachten Sie: Diese Schaltflächen sind erst aktiv, wenn der Dienst tatsächlich angelegt ist.

Anlegen eines neuen Dienstes:

1. Neben dem Feld **Projekt** klicken Sie auf **…**, um einen normalen Windows-Dateibrowser zu öffnen. Wählen Sie mit Hilfe des Browsers Ihre Projektdatei aus
2. Tragen Sie in den Feldern **Benutzer** und **Passwort** jeweils den Benutzer und das Passwort (falls vorhanden) für das Windows-Benutzerkonto ein, unter dem der Dienst ausgeführt werden soll
3. Wählen Sie einen **Start-Typ** aus
4. Klicken Sie auf **Anwenden**. Der Dienst wird mit den angegebenen Einstellungen angelegt

Nach dem Anlegen wird der Dienst in der Konsole "Dienste"(**Start ▸ Systemsteuerung ▸ Verwaltung ▸ Dienste**) angezeigt und zwar unter dem Namen "Rexroth WinStudio". Mit dieser Konsole kann der Dienst schnell angehalten und wieder ausgeführt werden, wenn Sie die "WinStudio"-Entwicklungsumgebung nicht extra aufrufen möchten

Befehlszeilen-Hilfsprogramm

Der Dienst kann auch mit Hilfe des Befehlszeilen-Hilfsprogramms **StdSvcInst.exe** konfiguriert werden. Dies bietet einige Optionen mehr als das oben beschriebene Werkzeug Dienst-Konfiguration an – wie z.B. das Angeben eines Namens und einer Beschreibung für den Dienst – und es lässt sich einsetzen, ohne dass die "WinStudio"-Entwicklungsumgebung gestartet werden muss. Das Hilfsprogramm findet sich im Ordner **\Bin** Ihres "WinStudio"-Programmverzeichnisses. Um das Hilfsprogramm auszuführen, öffnen Sie ein Eingabeaufforderungsfenster, navigieren zum Verzeichnis **\Bin** und geben das Programm als Befehl mit den gewünschten Optionen ein.

Das Hilfsprogramm hat die folgende Befehls-Syntax:

StdSvcInst { -create -app filepath -startup { auto | manual | disabled } -user username **-password** password **-name** displayname **-descr** description **| -start | -stop | -delete }**

-create

Erstellt den Windows-Dienst.

-app filepath

Legt fest, welche Projektdatei (**projektname.app**) der Dienst laden und ausführen soll, wenn er gestartet wird. (Dies entspricht dem Feld **Projekt** im Dialog "Dienstkonfiguration".) Sie müssen den vollständigen Dateipfad angeben und er muss in Anführungszeichen eingeschlossen sein.

Dieser Befehlsparameter ist beim Anlegen eines neuen Dienstes "vorgeschrieben".

-startup { auto | manual | disabled }

Legt fest, wie der Dienst starten soll. (Dies entspricht dem Feld **Start-Typ** im Dialog "Dienstkonfiguration".) Dieser Befehlsparameter ist optional; wenn er nicht verwendet wird, ist das Standardverhalten für einen neuen Dienst "manual" = manuell.

-user username

Gibt das Windows-Benutzerkonto an, unter dem der Dienst ausgeführt werden soll. (Dies entspricht dem Feld **Benutzer** im Dialog "Dienstkonfiguration".) Dieser Befehlsparameter ist optional; wenn sie nicht verwendet wird, wird der Dienst unter Lokales System ausgeführt.

-password password

Gibt das Passwort für das angegebene Benutzerkonto an. (Dies entspricht dem Feld **Passwort** im zuvor beschriebenen Dialog "Dienstkonfiguration".) Dieser Parameter ist optional; er wird nicht benötigt, wenn kein Benutzer angegeben ist oder wenn der angegebene Benutzer kein Passwort hat.

-name displayname

Legt den Dienstnamen fest, der auf der "Computerverwaltungs-Konsole" angezeigt wird. Der Name muss in Anführungszeichen eingeschlossen sein. Dieser Befehlsparameter ist optional; der Standardname lautet „WinStudio“.

-descr description

Legt die Dienstbeschreibung fest, die auf der "Computerverwaltungs-Konsole" angezeigt wird. Die Beschreibung muss in Anführungszeichen eingeschlossen sein. Dieser Befehlsparameter ist optional.

-start

Startet den Dienst. Dies ist das Gleiche wie das Starten des Dienstes mit Hilfe der "Computerverwaltungs-Konsole" oder das Anklicken von **Start** im Dialog "Dienstkonfiguration".

-stop

Hält den Dienst an. Dies ist das Gleiche wie das Anhalten des Dienstes mit Hilfe der "Computerverwaltungs-Konsole" oder das Anklicken von **Stopp** im Dialog "Dienstkonfiguration".

-delete

Löscht den Dienst.

Beispiel: Anlegen des Dienstes

In diesem Beispiel möchten wir einen neuen Windows-Dienst mit folgenden Optionen anlegen:

"WinStudio"-Projektdatei	**C:\Users\benutzername\Eigene Dokumente\WinStudio v7.3 Projects\projekt_name \projekt_name.app**
Start-Modus	automatisch
Benutzer	"WinStudio"
Passwort	"WinStudio"
Dienstname	"Rexroth WinStudio"
Dienstbeschreibung	"Startet ein WinStudio-Projekt"



Beachten Sie: Das System muss bereits ein Benutzerkonto „WinStudio“ mit dem Passwort „WinStudio“ enthalten.

Zum Erstellen dieses Diensts mit den gewünschten Optionen:

1. Stellen Sie sicher, dass Sie als Benutzer mit Administratorrechten angemeldet sind
2. Öffnen Sie ein Eingabeaufforderungs-Fenster (**Start ▸ Alle Programme ▸ Zubehör ▸ Eingabeaufforderung**)
3. Navigieren Sie zum Ordner Bin: **cd Bin**
4. Geben Sie den folgenden Befehl ein:

   StdSvcInst –create –app "C:\Users\benutzername\Eigene Dokumente \WinStudio v7.3 Projects\ projekt_name/projekt_name.app" -startup auto –user WinStudio -password WinStudio –name "WinStudio" –descr "Startet ein WinStudio-Projekt"

Wenn der Vorgang erfolgreich ist, zeigt das System die Meldung "Service erstellt" an. Anderenfalls zeigt es eine Fehlermeldung an.

Beispiel: Ändern der Projektdatei

Nach dem Anlegen des Dienstes wollen Sie ggf. die "WinStudio"-Projektdatei ändern, die er ausführt. Dies können Sie mit Hilfe des Befehlsparameters **-app** erreichen:

1. Stellen Sie sicher, dass Sie als Benutzer mit Administratorrechten angemeldet sind.
2. Halten Sie den Dienst an, während er noch ausgeführt wird.
3. Öffnen Sie ein Eingabeaufforderungs-Fenster.
4. Navigieren Sie zum Verzeichnis Bin.
5. Geben Sie den folgenden Befehl ein – um z.B. NTDemo als Projektdatei festzulegen:

   StdSvcInst –app "C:\Programme (x86)\Rexroth\WinStudio\Demos\ NTDemo\NTDemo.app"

Beispiel: Löschen des Dienstes

So löschen Sie den Dienst:

1. Stellen Sie sicher, dass Sie als Benutzer mit Administratorrechten angemeldet sind.
2. Halten Sie den Dienst an, während er noch ausgeführt wird.

3. Öffnen Sie ein Eingabeaufforderungs-Fenster.
4. Navigieren Sie zum Verzeichnis Bin.
5. Geben Sie den folgenden Befehl ein:

   StdSvcInst –delete

Konfigurieren von Benutzerrechten

Der Dienst wird unter den Rechten des Benutzerkontos ausgeführt, das im Feld **Benutzer** des Hilfsprogramms "Dienstkonfiguration" (oder durch den Befehlsparameter **-user** des Befehlszeilen-Hilfsprogramms) angegeben ist. Wenn "WinStudio" eine System-Ressource benötigt, zu der diesem Konto die Berechtigung fehlt, scheitert es. Dieses Konto muss folglich so konfiguriert werden, dass es die nötigen Rechte hat.

☞ Das Folgende kann nur von einem Benutzer mit vollen Administratorrechten durchgeführt werden.

Benutzerkonto für die Anmeldung als Dienst freigeben

Zunächst muss das angegebene Benutzerkonto aktiviert werden, um sich als Dienst am Rechner anzumelden. Aktivierung des Kontos:

1. Öffnen Sie die Konsole "Lokale Sicherheitseinstellungen" (**Start** ▸ **Systemsteuerung** ▸ **Verwaltung** ▸ **Lokale Sicherheitsrichtlinie**).
2. Im Konsolenfenster wählen Sie den Ordner „Sicherheitseinstellungen \Lokale Richtlinien\Zuweisen von Benutzerrechten“.
3. In der Liste verfügbarer Richtlinien führen Sie einen Doppelklick auf **Anmelden als Dienst** aus.

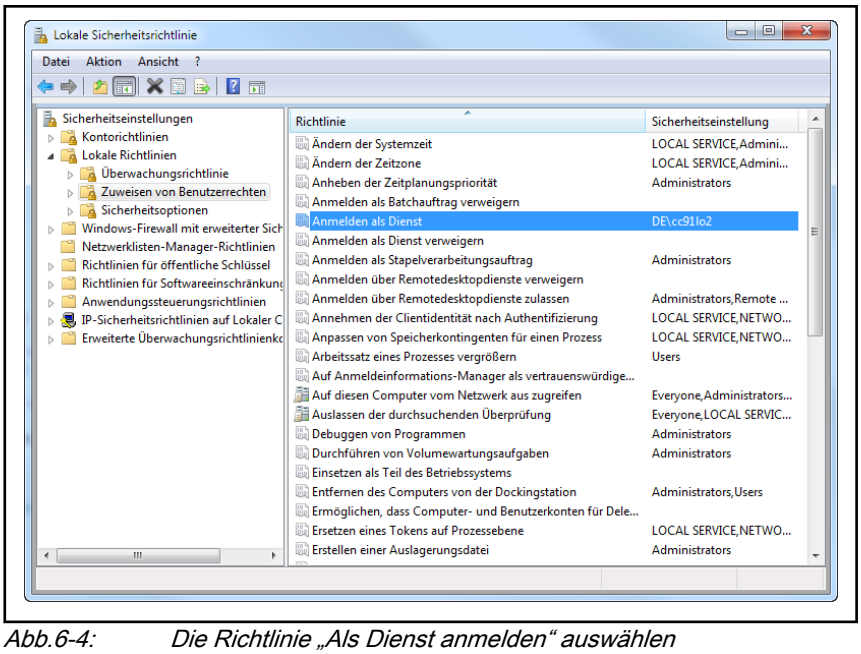


Abb.6-4: Die Richtlinie „Als Dienst anmelden“ auswählen

Die Eigenschaftenseite "Als Dienst anmelden" wird angezeigt.

4. Klicken Sie auf **Benutzer oder Gruppe hinzufügen ...**. Der Dialog "Benutzer oder Gruppen wählen" wird angezeigt.
5. Tragen Sie den Namen des Benutzerkontos ein, unter dem der Dienst ausgeführt werden soll.
6. Klicken Sie auf **OK**.

Benutzer vollständige Kontrolle über das Programmverzeichnis geben

Damit Ihr "WinStudio"-Projekt ordnungsgemäß ausgeführt wird, muss das angegebene Benutzerkonto vollständige Kontrolle über das Programmverzeichnis und alle darin enthaltenen Dateien haben. So geben Sie dem Konto diese Rechte:

1. Öffnen Sie den "Windows-Explorer" (**Start** ▸ **Alle Programme** ▸ **Zubehör** ▸ **Windows Explorer**).
2. Wählen Sie **Extras** ▸ **Ordneroptionen**. Der Dialog "Ordneroptionen" wird angezeigt.
3. Klicken Sie auf die Registerkarte **Ansicht**.
4. Stellen Sie sicher, dass die Option **Freigabe-Assistent verwenden** ausgewählt ist.

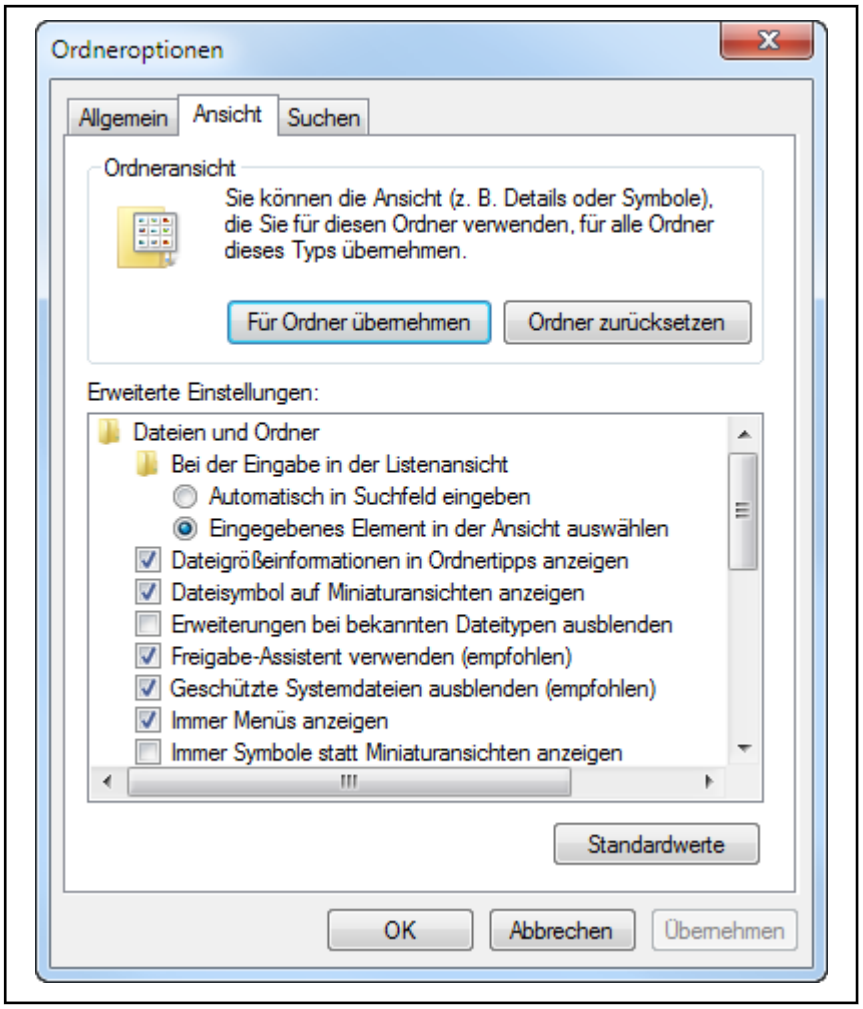


Abb.6-5: Stellen Sie sicher, dass "Freigabe-Assistent verwenden" aktiviert ist

5. Schließen Sie den Dialog "Ordneroptionen", indem Sie auf **OK** klicken.
6. Bleiben sie im "Windows-Explorer", suchen Sie Ihr "WinStudio"-Programmverzeichnis – d.h. den Ordner, der die Datei **projektname.APP** enthält – und wählen Sie es aus.

   In diesem Beispiel ist der Ordner "NTDemo".
7. Führen Sie einen Rechtsklick auf den Ordner aus und wählen Sie **Eigenschaften** aus dem Kontextmenü. Die Eigenschaftenaufstellung des Verzeichnisses wird angezeigt.

8. Klicken Sie auf die Registerkarte **Sicherheit**.

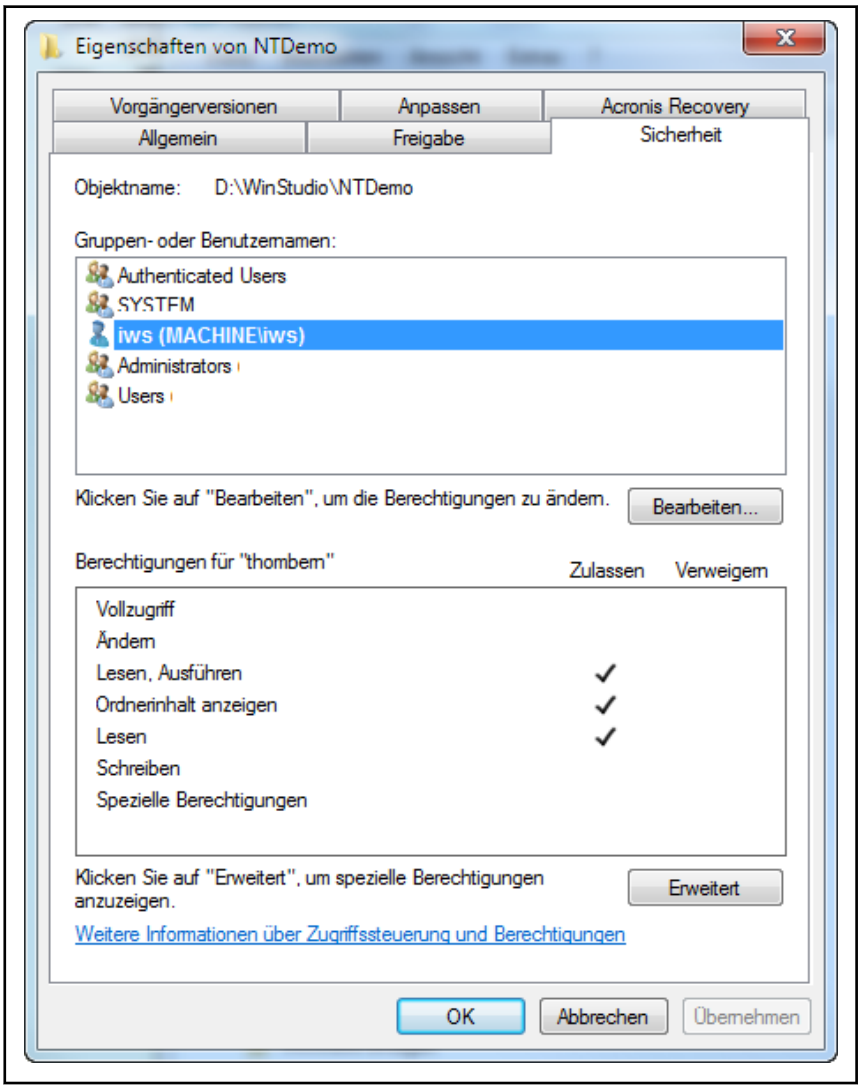


Abb.6-6: Eigenschaftendialog für das Programmverzeichnis

9. Klicken Sie auf **Hinzufügen** und fügen Sie dann das Benutzerkonto hinzu, das Sie beim Anlegen des Dienstes angegeben haben. In diesem Beispiel ist "iws" dieser Benutzer.
10. Wählen Sie diesen Benutzer aus und setzen Sie seine Berechtigung „Vollzugriff“ auf **Erlauben**.
11. Klicken Sie auf **OK**, um Ihre Einstellungen zu speichern und den Dialog zu schließen.
12. Schließen Sie den "Windows-Explorer".

Dem Benutzerkonto erlauben, das OPC-Client/Server-Modul auszuführen

Wie bereits erwähnt, haben normale Benutzer zu wenig Rechte, um das OPC-Client/Server-Modul ordnungsgemäß auszuführen. Dieses Konto muss folglich so konfiguriert werden, dass es diese Rechte hat:

1. Öffnen Sie die Konsole "Komponentendienste" (**Start ▸ Systemsteuerung ▸ Verwaltung ▸ Lokale Komponentendienste**).
2. Im Konsolenfenster wählen Sie den Ordner **Konsolenstamm\Komponentendienste\Computer\Arbeitsplatz\DCOM-Konfiguration**.

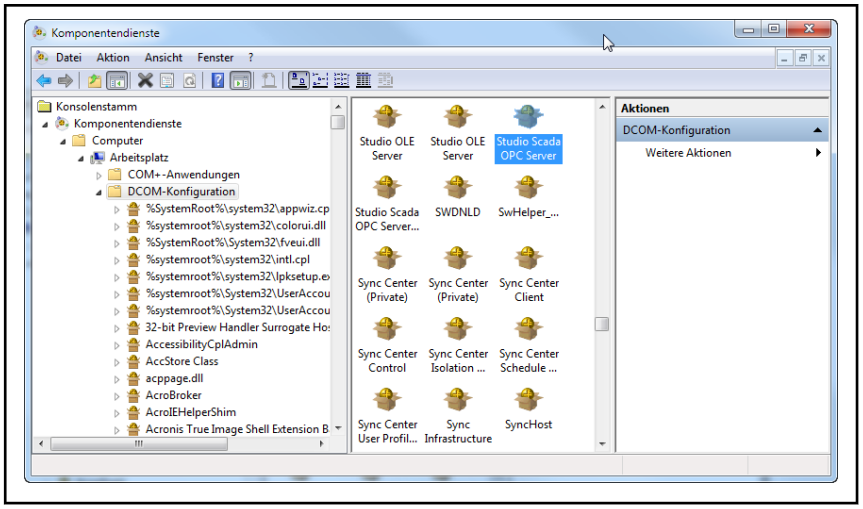


Abb.6-7: Das OPC-Server-Modul in der Komponentendienste-Konsole auswählen

3. Im Bereich DCOM-Konfiguration einen Rechtsklick auf das Symbol **WinStudio Scada OPC Server** ausführen und dann aus dem Kontextmenü **Eigenschaften** auswählen. Das Eigenschaftenblatt "WinStudio SCADA OPC-Server" wird angezeigt.
4. Klicken Sie auf die Registerkarte **Identität**.
5. Klicken Sie **Dieser Benutzer** an und füllen Sie die Felder dann mit demselben Benutzer und demselben Passwort, dass Sie beim Anlegen des Dienstes angegeben haben.

   In diesem Fall ist "iws" dieser Benutzer und "iws" ist auch das Passwort.

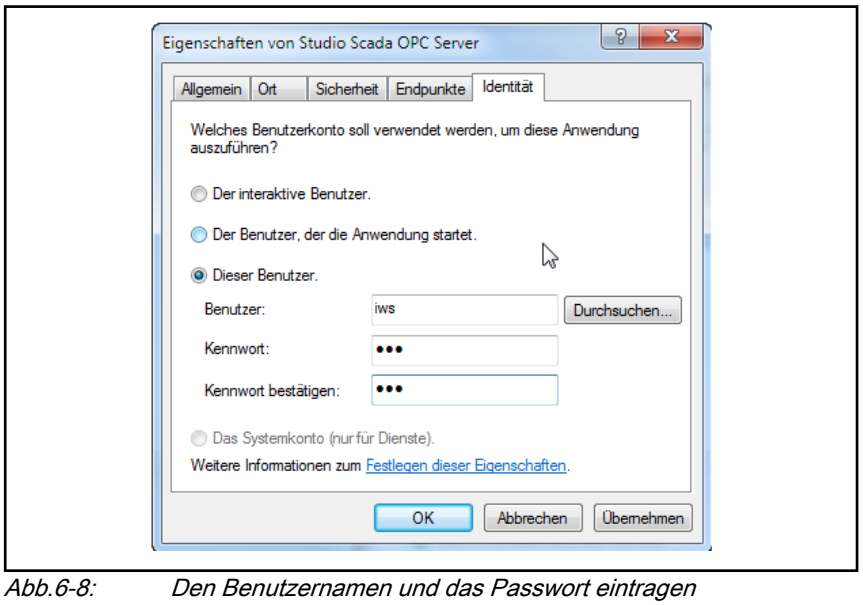


Abb.6-8: Den Benutzernamen und das Passwort eintragen

6. Klicken Sie auf **OK**, um Ihre Einstellungen zu speichern und das Eigenschaftenblatt zu schließen.
7. Schließen Sie die Konsole "Komponentendienste".

Fehlerbeseitigung

Wenn "WinStudio" unter Windows-Diensten ausgeführt wird, hat es keine Benutzeroberfläche. Wenn ein Fehler auftritt, wird er daher nur als Windows-Anwendungsereignis protokolliert. Diese Meldungen können mit Hilfe der "Ereignisanzeige"-Konsole (**Start** ▸ **Systemsteuerung** ▸ **System und Sicherheit** ▸ **Verwaltung** ▸ **Ereignisanzeige**) überprüft werden.

WinStudio-Projekte

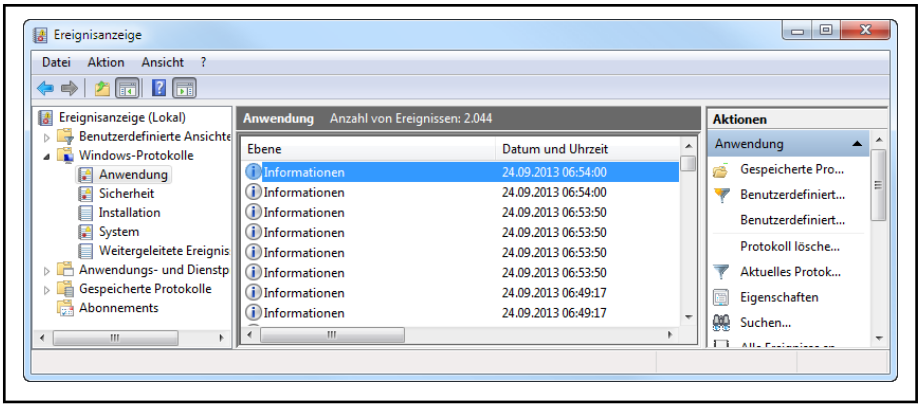


Abb.6-9: WinStudio-Fehlermeldungen in der Ereignisanzeige

7 Arbeiten mit Tags

7.1 Tags und die Projektdatenbank

7.1.1 Tags und die Projektdatenbank, Allgemeines

Tags sind ein Kernbestandteil jedes "WinStudio"-Projekts. Einfach ausgedrückt sind Tags Variablen, die von "WinStudio" verwendet werden zum Empfangen und Speichern von Daten aus der Kommunikation mit Anlagengeräten, aus den Ergebnissen von Berechnungen und Funktionen und aus Anwendereingaben. Tags können wiederum verwendet werden zum Anzeigen von Informationen auf Bildern (und Webseiten), zum Manipulieren von Bildobjekten und zur Steuerung von Laufzeit-Tasks auf Seite 357.

Aber Tags sind mehr als nur einfache Variablen. "WinStudio" enthält eine Echtzeit-Datenbankverwaltung mit einer Reihe von ausgereiften Funktionen wie Zeitstempel für alle Wertänderungen, Prüfung von Tag-Werten gegen Minimal- und Maximallaufzeitwerte, Vergleich von Tag-Werten mit Alarmgrenzen usw. Ein "WinStudio"-Tag verfügt sowohl über einen Wert als auch über verschiedene Eigenschaften, auf die zugegriffen werden kann – auf manche bei der Entwicklung und auf andere nur während der Laufzeit.

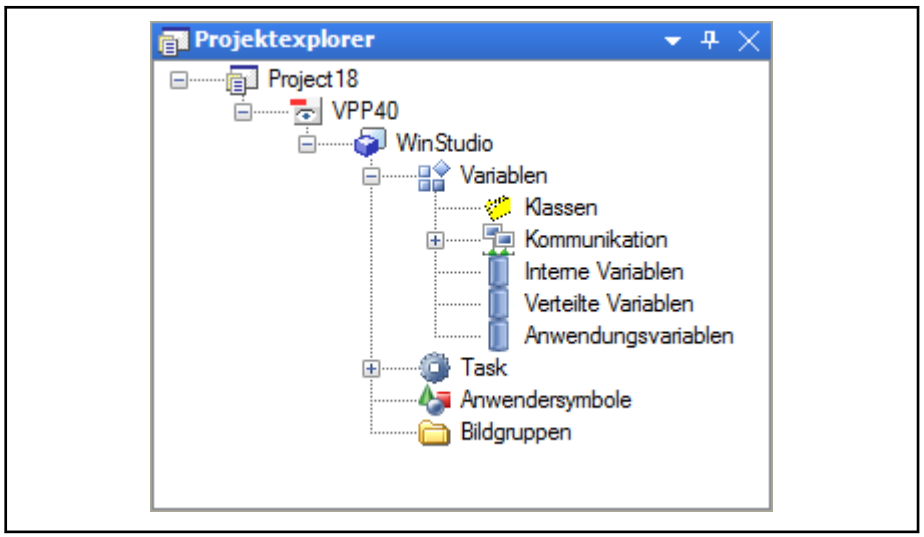


Abb.7-1: WinStudio-Tags im Projektexplorer

Alle WinStudio-Tags sind in eine der folgenden Kategorien eingeordnet, die jeweils als Element im Ordner "Variablen" dargestellt sind:

- Anwendungsvariablen auf Seite 116: sind Tags, die während der Projektierung erstellt werden. Anwendungsvariablen werden u.a. an den folgenden Stellen verwendet:
  - Bild-Tags
  - Tags, die Werten von Geräten lesen oder in ein Gerät schreiben
  - Steuerungs-Tags
  - Hilfs-Tags zur Verwendung bei mathematischen Berechnungen
- Verteilte Variablen auf Seite 120: Tags werden in einem PC-basierten Steuerprogramm angelegt und dann in die Tag-Datenbank von "WinStudio" importiert.

  Gemeinsame Tags können nicht in "WinStudio" verändert werden – die Tags müssen im PC-basierten Original-Steuerprogramm geändert und dann wieder in die Tag-Datenbank importiert werden.

- Interne Variablen auf Seite 121: sind vordefinierte Tags mit festgelegten Funktionen, die für Überwachungstasks in "WinStudio" verwendet werden. Zum Beispiel
  - enthalten Datums-Tags das aktuelle Datum im Zeichenkettenformat
  - enthalten Zeit-Tags die aktuell Uhrzeit im Zeichenkettenformat

  Die meisten internen Variablen sind "schreibgeschützt", d.h. diese Tags können nicht zur Datenbank hinzugefügt, geändert oder aus der Datenbank entfernt werden.

  Zur Anzeige einer Liste der internen Variablen öffnen Sie den Editor "Interne Variablen" im Ordner "Variablen". Die obige Abbildung zeigt eine Teilliste der internen Variablen.

Nach dem Erstellen des Tags kann dieser überall im Projekt verwendet werden und der gleiche Tag kann für mehr als ein Objekt oder Merkmal verwendet werden.

7.1.2 Anwendungsvariablen

Anwendungsvariablen, Allgemeines

Der Editor "Anwendungsvariablen" zeigt alle im aktuellen Projekt erstellten Tags an. Anwendungsvariablen können zum Anzeigen von Werten verwendet werden, zum Lesen und Schreiben an Außengeräte, zur Steuerung, zum Durchführen mathematischer Berechnungen usw.

☞ Vor dem Löschen eines Tags wird dringend die Verwendung des Werkzeuges**Querverweissuche** empfohlen, um zu überprüfen, ob der Tag nicht in einem anderen Teil des Projekts verwendet wird (Bilder, Mathematikblätter usw.). Wenn ein in einem anderen Teil des Projekts verwendeter Tag aus der Projektdatenbank gelöscht wird, entsteht ein Kompilierungsfehler und das Projekt funktioniert mangelhaft.

Zum Erstellen eines neuen Tags führen Sie einen Rechtsklick auf das Element "Anwendungsvariablen" aus und wählen **Variable einfügen** aus dem Kontextmenü.

Der Dialog "Neue Variable" wird angezeigt, wie in der folgenden Abbildung dargestellt:

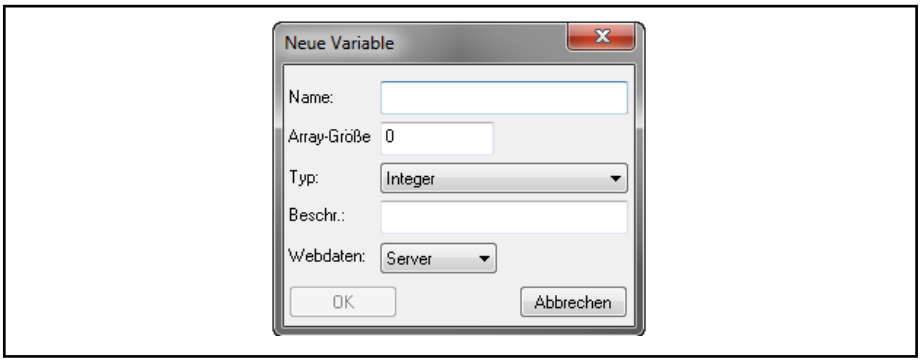


Abb.7-2: Dialog Neue Variable

Dieser Dialog wird zur Festlegung der folgenden Parameter verwendet.

- **Name**: Geben Sie einen Namen für den neuen Tag ein. Das erste Zeichen muss ein Buchstabe sein und der Name kann bis zu 255 Zeichen enthalten.

- **Array-Größe**: Geben Sie einen Wert zur Festlegung der Tag-Größe ein. Jeder Wert größer Null gibt an, dass der Tag ein Array auf Seite 125 ist.
- **Typ**: Wählen Sie einen Standard Tag-Typ auf Seite 124 aus der Liste (**Boolean**, **Integer**, **Real** oder **String**). Es können auch neue Typen, so genannte Klassen auf Seite 118 aus den Standardtypen definiert werden.
- **Beschreibung**: Geben Sie eine Tag-Beschreibung zu Dokumentationszwecken ein.
- **Webdaten**: Wählen Sie durch Klicken eine der folgenden Optionen aus:
  - **Lokal**: Die Informationen in diesem Tag sollen nicht über das Internet gemeinsam genutzt werden.
  - **Server**: Die Informationen in diesem Tag sollen über das Internet gemeinsam genutzt werden.

Diese Optionen haben keine Auswirkung auf Projekte ohne Webfähigkeiten. Wenn die Option **Webdaten** für ein Projekt mit Webfähigkeiten ausgewählt wird, dann funktionieren alle Objekteigenschaften, die den Typ "Lokal" verwenden, nicht ordnungsgemäß über das Internet.

Es müssen eindeutige Tag-Namen erstellt werden. Es kann kein Tag mit dem Namen eines bestehenden Tags erstellt werden.

Die Tag-Eigenschaften können aus einem der folgenden Dialoge eingesehen oder bearbeitet werden.

- Dialog "Tag-Eigenschaften": Öffnen Sie das Werkzeugfenster "Querverweise", geben Sie den Tag-Namen ein und drücken Sie die Schaltfläche **Tag-Eigenschaften**.
- Editor "Anwendungsvariablen": Klicken Sie im Ordner **Variablen** auf das Element **Anwendungsvariablen** .

Das Datenblatt "Anwendungsvariablen" enthält fünf Spalten (siehe folgenden Beispieldialog).

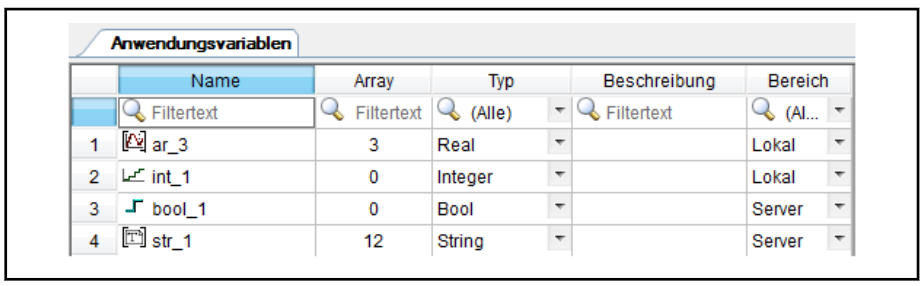


Abb.7-3: Editor Anwendungsvariablen

Dieser Editor dient zum Erstellen, Ändern oder Löschen von Tag-Eigenschaften. Sie können einen Rechtsklick auf eine Tag-Eigenschaft ausführen und Standard-Windows-Befehle zum Ausschneiden (<Strg>+<Z>X), Kopieren (<Strg>+<C>) oder Einfügen (<Strg>+<V>) eines Tags und seiner Eigenschaften verwenden. Außerdem können die letzen Änderungen in einem Feld rückgängig gemacht werden (<Strg>+<Z>).

Durch einen Rechtsklick und Auswählen der entsprechenden Option aus dem Kontextmenü können Daten im Datenblatt "Anwendungsvariablen" sortiert und/oder zusätzliche Spalten zu/aus dem Blatt hinzugefügt/entfernt werden.

Erweitern des Datenblatts Anwendungsvariablen

Das Arbeitsblatt Anwendungsvariablen kann bei Bedarf auf bis zu 65.488 Zeilen erweitert werden.

Das Datenblatt ist normalerweise auf eine Höchstgrenze von 32.721 Zeilen begrenzt. (Dies ist getrennt von der Maximalgröße der Projektdatenbank als Ganzes sowie der Laufzeitgrenze, die bei der Auswahl einer Zielplattform für ein neues Projekt festgelegt wird.)

Zum Erweitern des Arbeitsblatts wird die Projektdatei (**projekt_name.app**) durch Hinzufügen des folgenden Eintrags bearbeitet:

[Optionen]

EnableExtendedTagCount=1

Dies führt allerdings zu den folgenden Einschränkungen:

- Projekt-Tags in Zeilen 32.722-65.488 des Arbeitsblatts können nicht als Zeiger in Ausdrücken verwendet werden. Das heißt, ein Ausdruck wie **Abs(numArray[indexTag]), indexTag** kann sich nicht in diesen Zeilen befinden. (Diese Einschränkung gilt nicht für die VBScript-Schnittstelle).
- In einem Klassen-Arbeitsblatt können nur die ersten 32 Klassen-Member Alarme haben. Für alle Klassen-Member nach den ersten 32 funktionieren keine Alarme.

Allgemein gesagt bringt das Erweitern des Datenblatts "Anwendungsvariablen" die Leistungsfähigkeit von "WinStudio" an ihre Grenzen und sollte nur vorgenommen werden, wenn es absolut notwendig ist.

7.1.3 Ordner Klassen

Der Ordner "Klassen" enthält alle "Klassen" des Projekts und deren entsprechende Klassenelemente. Klassen sind zusammengesetzte Strukturen, die aus den Standarddatentypen Tag-Typen auf Seite 124 ("Integer", "Real", "Boolean" und "String") bestehen. Klassen ermöglichen Datenkapselung auf hoher Ebene in der Projekt-Datenbank. Ein Tag vom Typ Klasse bietet eine Wertegruppe für seine Klassenelemente.

Zum Definieren einer Klasse müssen die "Klassenelemente." und deren Typen definiert werden. Klassenelemente. sind Variablen, die Werte für ein Projekt mit bestimmten Merkmalen enthalten. Daher kann das Definieren einer Klasse sehr nützlich für Projekte mit einer wiederkehrenden Gruppe von Variablen sein.



Klassenvariablen werden im Editor **Anwendungsvariablen** im Ordner Projekt-Tags auf Seite 116 mit dem Klassensymbol angezeigt.

Zum Zugreifen auf die Elemente einer Klasse wird die folgende Syntax mit einem Punkt (".") als Trennzeichen verwendet: **TagName.MemberName**. Zum Beispiel: **tk.STD** oder **tk.TMP**.

Wenn der Tag "tk" ein Array auf Seite 125 ist, wird die folgende Syntax verwendet: **ArrayTagName[ArrayIndex].MemberName**

Zum Beispiel: **tk[1].STD** oder **tk[n].TMP**

Ein Tag vom Typ Klasse stellt eine mit der Klasse verbundene Wertegruppe (anstatt eines Einzelwerts) dar. Tags vom Typ Klasse werden durch Gruppieren einfacher Tags erstellt, die zu Klassenelementen werden. Die Höchstzahl an Klassenelementen für jede Klasse hängt von der Produktspezifikation ab. Klassenelementen können Standardtypen "Integer", "Real", "Boolean" und "String" enthalten, wie bereits erwähnt.

Zum Erstellen einer neuen Klasse öffnen Sie das Kontextmenü des Ordners "Klassen " und wählen den Eintrag"Klasse einfügen" aus.

Wenn der Dialog "Klasse einfügen" angezeigt wird, geben Sie einen Namen in das Feld **Name** ein und bestätigen Sie die Eingabe mit **OK**.

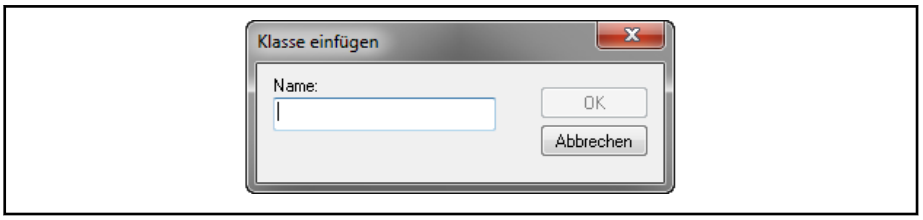


Abb.7-4: Dialog Klasse einfügen

Der Klassenname muss eindeutig sein, d. h. es können keine zwei Klassen mit dem selben Namen erstellt werden.

"Verteilte Variablen" und "Interne Variablen" können nicht als Klassen konfiguriert werden.

Wenn Sie das Datenblatt einer "Klasse" im Editor öffnen, können Sie Klassenelemente neu erstellen, ändern oder löschen.

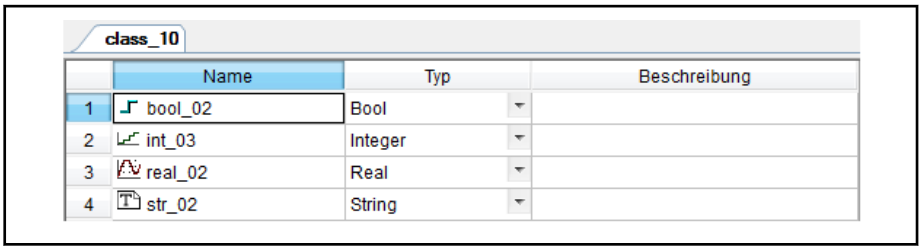


Abb.7-5: Datenblatt Klasse

Die maximale Anzahl der Elemente für jede Klasse ist 512, so lange dies nicht die von dieser für das Projekt ausgewählten Lizenz (Produktart) unterstützte Höchstzahl überschreitet.

Wenn ein Tag aus einem Klassentyp erstellt wird, zählt jedes Element der Klasse als ein Tag für Lizenzbeschränkungen, da jedes Klassenelement einen unabhängigen Wert hat.

Um ein Klassenelement oder eine Eigenschaft zu bearbeiten, führen Sie einen Rechtsklick auf das Element aus und verwenden Standard-Windows-Befehle zum Ausschneiden (<Strg>+<X>), Kopieren (<Strg>+<C>), oder Einfügen (<Strg>+<V>). Außerdem können die letzen Änderungen in einem Feld rückgängig gemacht werden (<Strg>+<Z>).

Klassenelemente können ebenfalls wie folgt bearbeitet werden:

- **Name**: geben Sie den Namen für das Klassenelement oder die Elementeigenschaft ein. Das erste Zeichen muss ein Buchstabe sein und der Name kann bis zu 255 Zeichen enthalten.
- **Typ**: wählen Sie einen Datentyp ("Boolean", "Integer", "Real" oder "String") aus.
- **Beschreibung**: geben Sie zu Dokumentationszwecken eine Beschreibung für das Klassenelement ein.

- Elemente einer Klasse können nicht zu einem anderen Klassentyp gehören
- der Klassenname muss eindeutig sein, d. h. der Name einer bestehenden Klasse kann nicht wiederverwendet werden
- aber: Elemente mit dem gleichen Namen können in verschiedenen Klassen erstellt werden

Zum Löschen einer Klasse und ihrer sämtlichen Klassenelemente führen Sie einen Rechtsklick auf einen Klassenordner aus und wählen Löschen.

"WinStudio" deaktiviert die Funktion "Löschen", wenn Laufzeit-Tasks ausgeführt werden.

Eine Klasse kann nicht gelöscht werden, solange eine Variable von der zu löschenden Klasse existiert.

7.1.4 Verteilte Variablen

Der Ordner "Verteilte Variablen" enthält alle von "WinStudio" und dem ausgewählten PC-basierten Steuerungsgerät gemeinsam verwendeten Tags. Diese Tags müssen in der PC-basierten Steuerungssoftware erstellt und geändert werden, dann werden sie automatisch unter den folgenden Bedingungen in "WinStudio" importiert:

- beim Start von "WinStudio" oder
- bei einem Rechtsklick auf den Ordner "Verteilte Tags" zum Aktualisieren (Update) der Datenbank.

Jedes PC-basierte Steuerungssoftwarepaket hat seine eigenen, individuellen Schnittstellenmerkmale und -bedingungen für den Tag-Import durch "WinStudio". So kann es zum Beispiel notwendig sein, dass die PC-basierte Steuerungssoftwareanwendung ausgeführt wird, um Datenbank-Tags nach "WinStudio" zu importieren.

Wenn keine PC-basierten Steuerungssoftwareprodukte mit der Anwendung verbunden sind, sind die Unterordner "Ansicht Datenblatt" und "Tag-Liste" (im Ordner "Verteilte Variablen") leer.

Verteilte Variablen können in der "WinStudio"-Umgebung nicht bearbeitet werden, sie können aber in der PC-basierten Steuerungssoftware verändert werden.

Verteilte Variablen können aber wie alle anderen Tags in allen "WinStudio"-Tasks konfiguriert werden.

Verteilte Variablen sind schreibgeschützt und sichtbar im Dialog "Tag-Eigenschaften" sowie dem Datenblatt "Verteilte Variablen".

Mit einem Rechtsklick auf den Ordner "Verteilte Variablen" (oder einen Klick auf **Ansicht Datenblatt**) und Auswahl der Option **Aktualisieren** wird die letzte "Version" der Tag-Datenbank der PC-basierten Steuerungssoftware aktualisiert.

Zum Ändern der Datenbank Gemeinsame Tags (Erstellen eines neuen Tags, Löschen von Tags oder Ändern von Tag-Eigenschaften) muss dieser Befehl zur Aktualisierung der gemeinsamen Datenbank von "WinStudio" aktiviert werden.

Zur Ansicht des Dialogs "Tag-Eigenschaften":

- Klicken Sie die Schaltfläche **Tag-Eigenschaften** im Werkzeugfenster "Querverweise" an (der Tag-Name muss im Feld **Tag-Name** sein) oder
- Doppelklicken Sie auf den Tag-Namen im Editor Projekt-Tags auf Seite 116)

Wenn das Datenblatt "Verteilte Variablen" angezeigt wird, enthält es vier Spalten ("Name", "Größe", "Typ" und "Beschreibung"). Dieses Datenblatt ist schreibgeschützt, es kann nur zur Ansicht von verteilten Variablen verwendet werden.

7.1.5 Interne Variablen

Der Ordner "Interne Variablen" enthält vordefinierte Tags mit bestimmten Funktionen (Uhrzeit, Datum, Alarmbestätigung, Speichern des protokollierten Benutzers usw.). Diese Tags können nicht bearbeitet oder gelöscht werden, ihre Werte können aber aus jedem "WinStudio"-Task heraus erreicht, kopiert und an anderer Stelle verwendet werden.



Die Datenbank der internen Variablen wird mit einem Rechtsklick auf das Element "Interne Variablen" aktualisiert. Eine Liste der verfügbaren internen Variablen einschließlich ihrer Eigenschaften und Beschreibungen finden Sie unter Liste der System-Tags auf Seite 821.

Die Eigenschaften einer internen Variablen können im Datenblatt "Interne Variablen" eingesehen werden, das vier Spalten enthält ("Name", "Größe", "Typ" und "Beschreibung").



Die meisten internen Variablen sind schreibgeschützt.

Um z.B. die Uhrzeit zu ändern, muss die mathematische Funktion SetSystemTime auf Seite 599 verwendet und nicht die interne Variable beschrieben werden.

7.2 Erstellen von Tags

7.2.1 Namenssyntax für Tags

Beim Benennen von Tags müssen die folgenden Richtlinien beachtet werden:

- Ein Tag-Name muss **eindeutig** sein, d. h. derselbe Name darf nicht für zwei verschiedene Tags (oder Funktionen) verwendet werden.

  Wird ein bereits existierender Tag-Name eingegeben, erkennt "WinStudio" dies und erstellt keinen neuen Tag.
- Ein Tag-Name muss mit einem "Buchstaben" beginnen. Zusätzlich können Buchstaben, Zahlen und der Unterstrich ("_") im Tag-Namen verwendet werden.
- Die folgenden Sonderzeichen dürfen **nicht** im Tag-Namen verwendet werden:

  ` ~ ! @ # $ % ^ & * ( ) - = / + \ [ ] { } < > ?
- Ein Tag-Name oder Klassenelementname darf maximal 255 Zeichen lang sein.

  Es können Groß- und Kleinbuchstaben verwendet werden. Tag-Namen sind **unabhängig** von Groß- und Kleinschreibung.



Da "WinStudio" nicht zwischen Groß- und Kleinbuchstaben unterscheiden kann, können beide verwendet werden, um Tag-Namen besser lesbar zu machen:

"FuellungTank" anstatt "fuellungtank"

- Tag-Namen müssen sich von Namen der internen Variablen und mathematischen Funktionen unterscheiden.



Verwenden Sie das Sonderzeichen "@" am Anfang eines Tag-Namens, um anzugeben, dass der Tag im Projekt als indirekter Tag auf Seite 131 verwendet wird.

Einige Beispiele für gültige Tag-Namen:

- Temperatur
- druck1
- Anzahl
- x

7.2.2 Auswahl des Tag-Typs

Mit "WinStudio" können die folgenden Tag-Typen erstellt werden:

- **Basis-Tags** enthalten einen einfachen Wert
- **Array-Tags** sind Tag-Gruppen, die den gleichen Namen mit eindeutigen Indizes verwenden
- **Klassen-Tags** sind eine Gruppe von zusammengesetzten Tags, die aus benutzerdefinierten Datentypen (Boolean, Integer, Real oder String) oder Datentypen-Strukturen bestehen
- **Indirekte Tags** sind Zeiger, die indirekten Zugriff auf einen anderen Tag-Typ erlauben, einschließlich Klassen-Tags

Im Folgenden werden diese Tag-Typen näher betrachtet.

Basis-Tags

Ein "Basis"-Tag erhält einen Einzelwert. Typischerweise sind die meisten für ein Projekt definierten Tags Basis-Tags.

Einige Beispiele für Basis-Tags:

- "TankID" (zur Kennzeichnung verschiedener Tanks in Ihrem Projekt)
- "Temperatur" (zur Angabe der aktuellen Temperatur eines Objekts)
- "Status" (zur Angabe, ob ein Objekt offen oder geschlossen ist)

Array-Tags

Ein "Array"-Tag besteht aus einer Reihe von Tags, die den gleichen Namen, aber eindeutige Indizes (eine Matrix aus "n" Zeilen und einer Spalte) zur Unterscheidung zwischen den einzelnen Tags haben. Ein "Array-Index" kann ein feststehender Wert, ein anderer Tag oder ein Ausdruck sein. Die maximale Arraygröße wird durch Produktspezifikationen festgelegt.

Array-Tags können verwendet werden für:

- Einfache Konfigurationen
- Ermöglichen von Multiplexverfahren in Bildern, Rezepturen und Kommunikationsschnittstellen
- Einsparung von Entwicklungszeit bei der Tag-Deklarierung

Array-Tags können in zwei Formaten angegeben werden:

- Für einen einfachen Array-Tag geben Sie ein:

```
ArrayTagName[ArrayIndex]
```

- Für einen zusammengesetzten Array-Tag (wenn der Array-Index ein Ausdruck aus einem Tag und einer Rechenoperation ist) geben Sie ein:

  ```
  ArrayTagName[ArrayIndex + c]
  ```

  Hierbei gilt:

  – **ArrayTagName** ist der Tag-Name

  – **[ArrayIndex]** ist der eindeutige Index (festgelegter Wert oder anderer Tag)

  – "+" ist eine Rechenoperation und

  – **c** ist eine numerische Konstante

☞

- Sie müssen für jeden Array-Tag einen Maximalindex angeben durch Eingabe eines Werts ("n") in der Spalte Arraygröße des "Anwendungsvariablen"-Datenblatts oder im Feld Array-Größe eines Dialogs "Neuer Tag" (siehe Anwendungsvariablen auf Seite 116).

  Wenn Sie einen Array-Tags mit "n" Positionen erstellen, erstellt "WinStudio" tatsächlich **"n"+1** Positionen (von 0 bis "n"). Wenn Sie zum Beispiel "ArrayTag[15]" erstellen, hat das Array 16 Elemente, wobei 0 die Anfangs- und 15 die Endposition ist.

- In einem Array-Tags dürfen keine Leerzeichen verwendet werden.

  Wenn "WinStudio" einen Tag liest, beginnt es mit dem ersten Zeichen und fährt fort, bis es das erste Leerzeichen oder die erste Null findet. Auf diese Weise erkennt das System keine Zeichen nach einem Leerzeichen als Teil eines Array-Tags.

  Wenn zum Beispiel **a[Sekunde + 1]** eingegeben wird, erkennt "WinStudio" **a[Sekunde** als Tag und betrachtet diesen als ungültig, weil "WinStudio" die schließende Klammer nicht findet (erkennt). Wenn aber **a[Sekunde+1]** eingegeben wird, ist dies ein gültiger Array-Tag.

Es kann überall dort ein Array-Tag spezifiziert werden, wo ein Variablennamen verwendet würde. Da Array-Tags außerdem die Konfiguration erheblich vereinfachen und Entwicklungszeit sparen, empfehlen wir, sie wo immer möglich zu verwenden.

Nehmen wir zum Beispiel an, dass die Temperatur von vier Tanks überwacht werden soll. Die herkömmliche Konfigurationsmethode ist:

- Temperatur1 – hohe Temperatur an Tank 1
- Temperatur2 – hohe Temperatur an Tank 2
- Temperatur3 – hohe Temperatur an Tank 3
- Temperatur4 – hohe Temperatur an Tank 4

Array-Tag können wie folgt zur Vereinfachung dieses Anwendungsfalls verwendet werden (wobei **[n]** für die Tanknummer steht):

- Temperatur[n] – hohe Temperatur an Tank "n"

Die folgende Tabelle enthält einige zusätzliche Beispiele für Array-Tags:

Arbeiten mit Tags

Beispiele für Array-Tags

Beispiel für Array-Tags	Beschreibung
Tank[1], **Tank[2]**, **Tank[500]**	Einfache Arrays, bei denen die Array-Indizes ("1", "2" und "500") numerische Konstanten sind. Zum Beispiel Tanknummern.
Tank[tk]	Ein einfaches Array, bei dem der Index ("tk") ein Tag ist. Zum Beispiel ein Tag, der für eine Tanknummer steht.
Tank[tk+1]	Ein Array, bei dem der Index ("tk+1") ein Ausdruck ist. Zum Beispiel der Wert von "tk" (Tanknummer) plus 1.

☞ Wenn bei der Verwendung eines anderen Tags als Bezug auf den Index eines Arrays der Wert des Tags außerhalb der Array-Grenze liegt, werden die folgenden Ergebnisse ausgegeben:

- Wenn **IndexTag** größer als die Größe des Arrays ist, weist **MeinArray[IndexTag]** auf die Endposition des Arrays und
- Wenn **IndexTag** kleiner als 0 ist, weist **MeinArray[IndexTag]** auf die Anfangsposition des Arrays hin (d.h., **MeinArray[0]**).

Indirekte Tags

Indirekte Tags „zeigen“ auf andere Datenbank-Tags (einschließlich Tags vom Typ Klasse). Die Verwendung von indirekten Tags kann Entwicklungszeit sparen, denn diese ersparen das Erstellen von doppelten Tags (und der darin enthaltenen Logik).

Ein indirekter Tag kann aus jedem Tag vom Typ String erstellt werden durch einfache Eingabe des Sonderzeichens "@" vor dem Tag-Namen **@TagName**.

- Zum Referenzieren auf einen einfachen Tag wird angenommen, dass der Tag **strX** (ein String-Tag) den Wert „Tank“ enthält, der der Name eines anderen Tags ist. Dann bietet das Lesen oder Schreiben auf **@strX** Zugriff auf den Wert des Tags **Tank**.
- Zum Referenzieren eines Tags und Members vom Typ Klasse wird einfach ein String-Tag erstellt, der auf den Klasse-Tag und das Member zeigt.

  Wenn zum Beispiel ein Tag **strX** (ein String-Tag) den Wert „Tank.Fuellstand“ enthält, der der Name des Klassen-Tags ist, dann bietet das Lesen oder Schreiben auf **@strX** Zugriff auf den Wert des Tags **Tank.Fuellstand**.
- Es kann auch direkt auf ein Member eines Tags vom Typ Klasse gezeigt werden durch Kennzeichnung eines Klasse-Typs, der auf ein Klasse-Member zeigt.

  Zum Beispiel: zum Zugriff auf das Member **Tank.Fuellstand** der Klasse muss der Wert „Tank“ im Tag **strX** gespeichert und die Syntax **@strX.Fuellstand** verwendet werden.

7.2.3 Auswahl des Tag-Datentyps

Außerdem sollte beim Entwurf eines Tags berücksichtigt werden, welche Datentypen der Tag erhalten wird. "WinStudio" erkennt die folgenden Standard-Tag-"Datentypen":

- **Boolean** ("ein Bit"): Einfacher Boolescher Wert mit den möglichen Werten 0 (falsch) und 1 (wahr). Gleichbedeutend mit dem Datentyp "bool" in C++. Üblicherweise zum Ein- und Ausschalten von Objekten und zum Schließen und Öffnen von Objekten verwendet.

- **Integer** ("vier Bytes"): Ganze Zahl (positiv, negativ oder Null), intern als signiertes 32-Bit gespeichert.

  Gleichbedeutend mit dem Datentyp "signed long int" in C++. Üblicherweise zum Zählen ganzer Zahlen oder zum Setzen von Ganzzahlwerten verwendet.

  Beispiele: **0, 5, −200**.
- **Real** ("Gleitkomma, acht Bytes"): Reelle Zahl, die intern als signiertes 64-Bit gespeichert wird.

  Gleichbedeutend mit dem Datentyp "double" in C++. Üblicherweise für Messwerte oder für Dezimal- oder Bruchwerte verwendet.
- **String** ("alphanumerische Daten, bis zu 1024 Zeichen"): Zeichenkette bis zu 1024 Zeichen, enthält Buchstaben, Zahlen oder Sonderzeichen.

  Unterstützt sowohl ASCII- als auch UNICODE-Zeichen.

  Beispiele: **Rezeptur Produkt X123, 01/01/90, *** Ein *****.

Ein Tag kann durch Zuweisung einer Klasse auf Seite 128 auch in einen zusammengesetzten Tag verwandelt werden.

Eine Klasse ist eine Vorlage, die aus zwei oder mehr Tag-Definitionen besteht, jede mit ihren eigenem Datentyp.

Klassen können in Projekten mit Elementen (z.B. Flüssigkeitstanks) mit mehrfachen, zu überwachenden oder zu steuernden Merkmalen (z.B. Füllstand, Temperatur, Druck) verwendet werden.

Weitere Informationen unter: Tag-Eigenschaften und -Parameter auf Seite 136

Ändern, wie Boolean-Tags numerische Werte erhalten

Wenn ein numerischer Wert ungleich 0 (d.h. ≠0) in einen Boolean-Tag geschrieben wird, nimmt der Tag standardmäßig den Wert 1 an.

Dieses Verhalten kann bei Bedarf durch Bearbeiten der Datei **project_name.app** zum Verändern der folgenden Einstellung geändert werden:

```
[Optionen]
BooleanTrueAboveZero=value
```

Wird **BooleanTrueAboveZero** auf den Standardwert "0" gesetzt, verhält sich das Projekt wie oben beschrieben.

Wird **BooleanTrueAboveZero** auf den Standardwert "1" gesetzt, verhält sich das Projekt wie folgt:

- Wird ein numerischer Wert kleiner oder gleich 0 (d.h. ≤0) in einen Boolean-Tag geschrieben, nimmt der Tag den Wert 0 (falsch) an.
- Wird ein numerischer Wert größer 0 (d.h. >0) in einen Boolean-Tag geschrieben, nimmt der Tag den Wert 1 (wahr) an.

HINWEIS	**Dies ist eine globale Laufzeiteinstellung.**

Wenn Sie nur ändern wollen, wie bestimmte Tags gehandhabt werden, sollten Sie diese Einstellung nicht ändern.

7.2.4 Verwenden von Array-Tags

"WinStudio"-Tags können aus einem Einzelwert oder einem Array bestehen.

Die maximale Arrays-Größe ist 16384, so lange dies nicht die von dieser für das Projekt ausgewählten Lizenz (Produktart) unterstützte Höchstzahl überschreitet.

Jede Array-Position (einschließlich der Position 0) zählt als ein Tag für Lizenzbeschränkungen, da jede Position einen unabhängigen Wert hat.

Ein Array-Tag ist eine Tag-Gruppe mit dem gleichen Namen, der durch Indizes (eine Matrix aus **n** Zeilen und 1 Spalte) gekennzeichnet wird. Die maximale Array-Größe ist abhängig von der Produktspezifikation.

Die folgende Syntax kann für den Zugriff auf einen Array-Tag verwendet werden:

```
ArrayTagName[ArrayIndex]
```

Zum Beispiel: **tank[0]**, **tank[1]**, **tank[2]** und **tank[500]**.

Für jeden Array-Tag muss in der Spalte **Größe** jedes Datenblatts ein Höchstindex angegeben werden.

Es kann **n** angegeben werden, um anzuzeigen, dass der Array-Tag Positionen von 0 bis **n** hat.

Wenn zum Beispiel TagA die Größe 3 hat, könnten die Tag-Elemente **TagA[0]**, **TagA[1]**, **TagA[2]** und **TagA[3]** sein.



Verwenden Sie den Array-Tag wo immer möglich, da er die Speicherausnutzung optimiert und die Konfiguration vereinfacht.

Wenn Sie zum Beispiel eine Anzeige zur Überwachung jedes Tanks möchten, könnten Sie Array-Tags zum Konfigurieren einer einzelnen Anzeige verwenden, die Tags mit Verbindungen zu allen Tanks enthält.

Zum Beispiel (Verwendung des Tags **tk** als Index mit der Nummer des Tanks): **Druck[tk]**, **Temperatur[tk]** und **Temperatur[tk+1]**.

Ein Array-Tag kann ein Tag, ein numerischer Wert oder ein Ausdruck mit dem Rechenzeichen "+" sein.

Wenn Sie einen Bezug zu einem Array mit einem Index mit dem Rechenzeichen "+" herstellen, müssen die folgende Syntax verwenden:

```
ArrayTagName[NumWert1+NumWert2]
```

Wobei **NumWert1** und **NumWert2** ein Integer-Tag oder eine numerische Konstante sein können.

Zum Beispiel: **Temperatur[tk+2]** oder **Temperatur[tk+6]**.

Die Verwendung von Array-Tags in "WinStudio" kann eine bedeutende Zeitersparnis bei der Projektentwicklung darstellen.

Wenn zum Beispiel Tag-Punkte mit Bezug auf die Temperatur von vier Tanks benötigt werden. Die herkömmliche Konfigurationsmethode lautet wie folgt:

- **Temperatur1**: hohe Temperatur an Tank 1
- **Temperatur2**: hohe Temperatur an Tank 2
- **Temperatur3**: hohe Temperatur an Tank 3
- **Temperatur4**: hohe Temperatur an Tank 4

Die Verwendung von Array-Tag vereinfacht diese Aufgabe wie folgt:

- **Temperatur[j]**: hohe Temperatur an Tank {j}

Wenn Sie einen Array-Tag mit vier Positionen erstellen, erstellt das System fünf Positionen (von 0 bis 4).

Zum Beispiel:

```
tag_example[15] //start position=0, end position=15
```

Daher hat das Array **tag_beispiel[15]** 16 Elemente.

Wenn bei der Verwendung eines anderen Tags als Bezug auf den Index eines Arrays der Wert des Tags außerhalb der Array-Größe liegt, werden die folgenden Ergebnisse ausgegeben:

- Wenn **IndexTag** größer als die Größe des Arrays ist, weist **MeinArray[IndexTag]** auf die Endposition des Arrays und
- wenn **IndexTag** kleiner als 0 ist, weist **MeinArray[IndexTag]** auf die Anfangsposition des Array hin.

Array-Tags

Ein "Array"-Tag besteht aus einer Reihe von Tags, die den gleichen Namen, aber eindeutige Indizes (eine Matrix aus **n** Zeilen und einer Spalte) zur Unterscheidung zwischen den einzelnen Tags haben. Ein "Array-Index" kann ein feststehender Wert, ein anderer Tag oder ein Ausdruck sein. Die maximale Arraygröße wird durch Produktspezifikationen festgelegt.

Array-Tags können verwendet werden für:

- Einfache Konfigurationen
- Ermöglichen von Multiplexverfahren in Bildern, Rezepturen und Kommunikationsschnittstellen
- Einsparen von Entwicklungszeit bei der Tag-Deklarierung

Array-Tags können in einem von zwei Formaten angegeben werden:

- Für einen einfachen Array-Tag geben Sie ein:

  `ArrayTagName[ArrayIndex]`
- Für einen zusammengesetzten Array-Tag (wenn der Array-Index ein Ausdruck aus einem Tag und einer Rechenoperation ist) geben Sie ein:

  `ArrayTagName[ArrayIndex + c]`

  Hierbei gilt:
  - **ArrayTagName** ist der Tag-Name
  - **[ArrayIndex]** ist der eindeutige Index (festgelegter Wert oder anderer Tag)
  - "+" ist eine Rechenoperation und
  - **c** ist eine numerische Konstante
- Sie müssen für jeden Array-Tag einen Maximalindex angeben durch Eingabe eines Werts (**n**) in der Spalte Array-Größe des "Anwendungsvariablen"-Datenblatts oder im Feld Array-Größe eines Dialogs "Neuer Tag".

  (siehe „Erstellen von Projektdatenbank-Tags auf Seite 116“).

  Wenn Sie einen Array-Tag mit **n** Positionen erstellen, erstellt "WinStudio" tatsächlich **n+1** Positionen (von **0** bis **n**).

  Wenn Sie zum Beispiel **ArrayTag[15]** erstellen, hat das Array 16 Elemente, wobei 0 die Anfangs- und 15 die Endposition ist.
- In einem Array-Tag dürfen keine Leerzeichen verwendet werden. Wenn "WinStudio" einen Tag liest, beginnt es mit dem ersten Zeichen und fährt fort, bis es das erste Leerzeichen oder die erste Null findet. Dementsprechend erkennt das System keine Zeichen nach einem Leerzeichen als Teil eines Array-Tag

  Wenn Sie zum Beispiel **a[Sekunde + 1]** eingeben, erkennt "WinStudio" **a[Sekunde** als Tag und betrachtet diesen als ungültig, weil "WinStudio" die schließende Klammer nicht findet (erkennt).

  Wenn Sie aber **a[Sekunde+1]** eingeben, ist dies ein gültiger Array-Tags

Sie können überall dort einen Array-Tags spezifizieren, wo Sie einen Variablennamen verwenden würden. Da Array-Tags außerdem die Konfiguration erheblich vereinfachen und Entwicklungszeit sparen können, empfehlen wir, sie wo immer möglich zu verwenden.

Nehmen wir zum Beispiel an, dass die Temperatur von vier Tanks überwacht werden soll. Die herkömmliche Konfigurationsmethode ist:

- **Temperatur1** – hohe Temperatur an Tank 1
- **Temperatur2** – hohe Temperatur an Tank 2
- **Temperatur3** – hohe Temperatur an Tank 3
- **Temperatur4** – hohe Temperatur an Tank 4

Sie können Array-Tags wie folgt zur Vereinfachung dieses Tasks verwenden (wobei **[n]** für die Tanknummer steht):

- **Temperatur[n]** – hohe Temperatur an Tank **[n]**

Die folgende Tabelle enthält einige zusätzliche Beispiele für Array-Tags:

Beispiele für Array-Tags

Beispiel für Array-Tags	Beschreibung
Tank[1], Tank[2], Tank[500]	Einfache Arrays, bei denen die Indizes (1,2 und 500) numerische Konstanten sind. Zum Beispiel Tanknummern.
Tank[tk]	Einfaches Array, bei dem der Array-Index (tk) ein Tag ist, z. B. ein Tag, der für eine Tanknummer steht.
Tank[tk+1]	Verbund, bei dem der Array-Index (tk+1) ein Ausdruck ist, z. B. der Wert von tk (Tanknummer) plus 1.

Wenn bei der Verwendung eines anderen Tags als Bezug auf den Index eines Arrays der Wert des Tags außerhalb der Array-Größe liegt, werden die folgenden Ergebnisse ausgegeben:

- Wenn **IndexTag** größer als die Größe des Arrays ist, weist **MeinArray[IndexTag]** auf die Endposition des Arrays und
- Wenn **IndexTag** kleiner als 0 ist, weist **MeinArray[IndexTag]** auf die Anfangsposition des Arrays hin (d.h., **MeinArray[0]**).

7.2.5 Verwenden von Klassen

"Klassen"-Tags sind zusammengesetzte Tags, die einen hohen Grad an Datenkapselung innerhalb der Tag-Datenbank ermöglichen. Wo Basis-Tags einen Einzelwert erhalten, werden "Klassen" zum Erhalten von Mehrfachwerten entworfen.

Ein Tag vom Typ Klasse kann durch Gruppieren von Basis- oder Array-Tags, die dann zu Klasse-"Members" werden, erstellt werden. Die Höchstzahl an Members für jede Klasse hängt von der Produktspezifikation ab.

Klasse-Tags können in zwei Formaten angegeben werden:

- Die Syntax für einfache Klasse-Tags ist

  `TagName.KlasseMemberName`

  (Wobei der Punkt als Trennzeichen verwendet wird.)

  Wenn z.B. mehrere verschiedene Bedingungen (z.B. "Temperatur", "Füllstand" und "Druck") in einem Tank überwacht werden sollen, könnte ein Klassen-Tag wie folgt erstellt werden:

  - **Tank.Temperatur**
  - **Tank.Fuellstand**

  - **Tank.Druck**

- Die Syntax zum Erstellen eines zusammengesetzten Klassen-Tags (Verwendung eines Array-Tags) ist

  `ArrayTagName[ArrayIndex].KlasseMemberName`

  (Wobei wiederum der Punkt als Trennzeichen verwendet wird.)

  Wenn die Bedingungen "Temperatur", "Füllstand" und "Druck" in mehreren Tanks überwacht werden sollen, könnte ein Klassen-Tag wie folgt erstellt werden:

  - **Tank[tk].Temperatur**
  - **Tank[tk].Fuellstand**
  - **Tank[tk].Druck**

  Wobei "tk" ein Array-Index ist, der für die Tanknummer steht.

Ordner Klassen

Der Ordner **Klassen** enthält alle Klassen des Projekts.

Klassen sind zusammengesetzte Tags, die aus benutzerdefinierten Datentyp-Strukturen oder Tag-Typen auf Seite 124 ("Integer", "Real", "Boolean" und "String") bestehen.



Klassen ermöglichen Datenkapselung auf hoher Ebene in der Projekt-Datenbank.

Ein Tag vom Typ Klasse bietet eine Wertegruppe für seine Klassenelemente.

Zum Definieren einer Klasse müssen die "Klassenelemente" und deren Typen definiert werden.

Klassenelemente sind Variablen, die Werte für ein Projekt mit bestimmten Merkmalen enthalten. Daher kann das Definieren einer Klasse sehr nützlich für Projekte mit einer wiederkehrenden Gruppe von Variablen sein.

Klassen im Ordner "Klassen" sowie Tags vom Datentyp "Klasse" im Ordner Anwendungsvariablen auf Seite 116werden mit dem Symbol angezeigt.

Zum Zugreifen auf die Elemente einer Klasse wird die folgende Syntax mit einem Punkt (".") als Trennzeichen verwendet:

`TagName.MemberName`

Zum Beispiel: **tk.STD** oder **tk.TMP**.

Wenn der Tag "tk" ein Array auf Seite 125 ist, wird die folgende Syntax verwendet:

Programm:

```
ArrayTagName[ArrayIndex].MemberName
```

Zum Beispiel: **tk[1].STD** oder **tk[n].TMP**

Ein Tag vom Typ Klasse enthält eine mit der Klasse verbundene Wertegruppe (anstatt eines Einzelwerts).

Tags vom Typ Klasse werden durch Gruppieren einfacher Tags erstellt, die zu Klassenelementen werden. Die Höchstzahl an Elementen für jede Klasse hängt von der Produktspezifikation ab. Klassenelemente können Standardwerte "Integer", "Real", "Boolean" und "String" enthalten.

Zum Erstellen einer neuen Klasse wird eine der folgenden Methoden zum Öffnen des Dialogs "Klasse einfügen" verwendet:

- Anwahl des Eintrags "Neue Klasse" im Kontextmenü des Ordners "Klassen" oder
- Erstellen eines neuen Klassen-Tags im Ordner Projekt-Tags auf Seite 116.

Wenn der Dialog "Klasse einfügen" angezeigt wird, geben Sie einen Namen in das Feld **Name** ein und klicken dann auf **OK** zum Schließen des Dialogs.

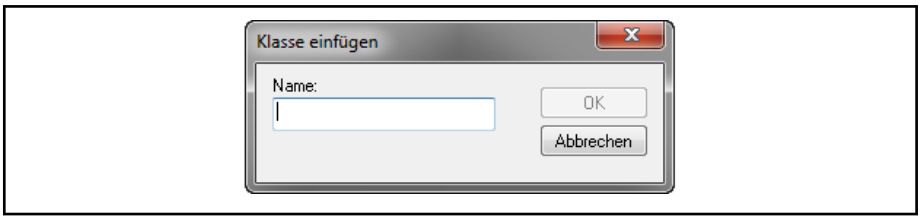


Abb.7-6: Dialog Klasse einfügen

Es muss ein eindeutiger Klassenname eingegeben werden. Es können keine zwei Klassen mit dem selben Namen erstellt werden. Außerdem können gemeinsame Tags und Systemtags nicht als Klassen konfiguriert werden.

Wenn das Datenblatt "Klasse" angezeigt wird, kann dies zum Erstellen, Ändern oder Löschen von Klassenelementen und deren sichtbaren Eigenschaften verwendet werden. (Klassen können nicht aus dem Dialog "Tag-Eigenschaften" bearbeitet werden.)

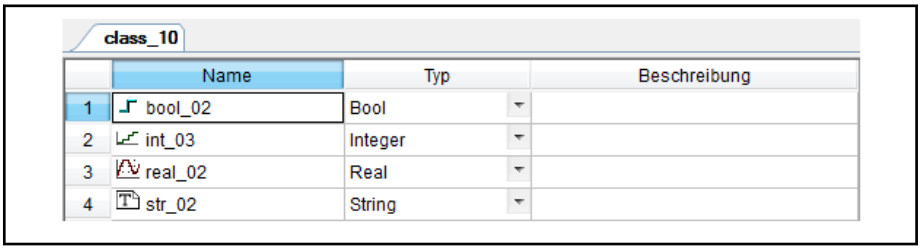


Abb.7-7: Datenblatt Klasse

Die maximale Anzahl der Klassenelemente für jede Klasse ist 512, so lange dies nicht die von dieser für das Projekt ausgewählten Lizenz (Produktart) unterstützte Höchstzahl überschreitet. Wenn ein Tag aus einem Klassentyp erstellt wird, zählt jedes Element der Klasse als ein Tag für Lizenzbeschränkungen, da jedes Klassenelement einen unabhängigen Wert hat.

Um ein Klassenelement oder eine Eigenschaft zu bearbeiten, führen Sie einen Rechtsklick auf das Element aus und verwenden Standard-Windows-Befehle zum Ausschneiden <Strg>+<X>, Kopieren <Strg>+<C> oder Einfügen <Strg>+<V>. Außerdem können die letzen Änderungen in einem Feld rückgängig gemacht werden <Strg>+<Z>.

Elementeigenschaften können ebenfalls wie folgt bearbeitet werden:

- Feld **Name**: Geben Sie den Namen für das Klassenelement oder die Elementeigenschaft ein. Das erste Zeichen muss ein Buchstabe sein und der Name kann bis zu 255 Zeichen enthalten
- Combobox **Typ**: Wählen Sie einen Datentyp aus ("Boolean", "Integer", "Real" oder "String")
- Feld **Beschreibung**: Geben Sie zu Dokumentationszwecken eine Beschreibung für das Klassenelement ein

Elemente einer Klasse können nicht zu einem anderen Klassentyp gehören. Außerdem muss ein eindeutiger Klassenname erstellt werden. Der Name einer bestehenden Klasse kann nicht wiederverwendet werden. Es können aber Elemente mit dem gleichen Namen in verschiedenen Klassen erstellt werden.

Zum Löschen einer Klasse und ihrer sämtlichen Elemente führen Sie einen Rechtsklick auf einen Klassenordner aus und wählen Löschen.

"WinStudio" deaktiviert die Funktion Löschen, wenn Laufzeit-Tasks ausgeführt werden.

Eine Klasse kann nicht gelöscht werden, wenn sie mit einem noch existierenden Tag verbunden ist.

7.2.6 Verwendung von indirekten Tags (Zeiger)

"WinStudio" unterstützt einen indirekten Zugriff auf Tags in der Datenbank.

Beispiel: einen Tag **X** vom typ String.

Dieser Tag kann den Namen eines beliebigen anderen Tags in der Datenbank enthalten (d.h. er kann einen Zeiger auf einen beliebigen anderen Tag-Typ bieten, einschließlich des Typs Klasse).

Syntax für einen indirekten Tag:

`@IndirekterTagName`

Ein Tag mit dem Namen **X** enthält einen String "TEMP".

Lesen und/oder Schreiben nach **@X** gibt Zugriff auf den Wert der Variablen "TEMP".

Jeder als String-Tag erstellter Tag ist potentiell ein indirekter Tag (Zeiger).

Um auf einen Tag vom Typ Klasse zu verweisen, kann ein Tag vom Typ String deklariert werden, der auf einen Klasse-Tag zeigt:

Klasse	"TANK" mit dem Element "Fuellstand"
Tag	"TK" mit Klasse "TANK"
Tag	"XCLASS" vom Typ String

Zum Zugreifen auf den Wert **TK.Fuellstand** müssen der Wert **TK.Fuellstand** innerhalb des Tags "XCLASS" gespeichert und die Syntax **@XCLASS** verwendet werden.

Es kann auch direkt auf ein Element eines Tags vom Typ Klasse Bezug genommen werden durch Kennzeichnung eines Klasse-Typs, der auf ein Klassenelement zeigt.

Zum Beispiel:

Klasse	"TANK" mit dem Element "Fuellstand"
Tag	"TK" mit Klasse "TANK"
Tag	"XCLASS" mit Klasse "TANK"

Zum Zugreifen auf den Wert **TK.Fuellstand** muss der Wert „"TK"“ innerhalb des Tags "XCLASS" gespeichert werden und die Syntax **@XCLASS.Fuellstand** verwenden.

Beim Erstellen von Tags zur indirekten Verwendung muss ein **X** in die Tag-Spalte geschrieben werden, anstatt diese als Strings zu erstellen. Für den Type den gewünschten Tag-Typ, für den ein Bezug erstellt wird, eingeben.

Gemäß dem Beispiel für "XCLASS": **@Z Integer**, **@X Klasse:TANK**.

Indirekte Tags Indirekte Tags „zeigen“ auf andere Datenbank-Tags.



Die Verwendung von indirekten Tags kann Entwicklungszeit sparen, denn diese indirekten Tags ersparen das Erstellen von doppelten Tags (und der darin enthaltenen Logik).

Ein indirekter Tag kann aus jedem Tag vom Typ String erstellt werden durch einfache Eingabe des Sonderzeichens "@" vor dem Tag-Namen **@TagName**.

- Zum Referenzieren auf einen einfachen Tag wird angenommen, dass der Tag "strX" (ein String-Tag) den Wert „Tank“ enthält, der der Name eines anderen Tags ist, dann bietet das Lesen oder Schreiben auf **@strX** Zugriff auf den Wert des Tags "Tank".
- Zum Referenzieren eines Tags und Elementen vom Typ Klasse wird einfach ein String-Tag erstellt, der auf den Klasse-Tag und das Element zeigt.

  Wenn zum Beispiel ein Tag "strX" (ein String-Tag) den Wert "„Tank-Fuellstand“" enthält, der der Name des Klasse-Tags ist, dann biet das Lesen oder Schreiben auf **@strX** Zugriff auf den Wert des Tags **Tank.Fuellstand**.
- Es kann auch direkt ein Element eines Klassen-Tags referenziert werden durch Kennzeichnung eines Klassen-Typs, der auf ein Klassenelement zeigt.

  Zum Beispiel: zum Zugriff auf das Element **Tank.Fuellstand** der Klasse müssen der Wert "Tank" im Tag "strX" gespeichert und die Syntax **@strX.Fuellstand** verwendet werden.

7.2.7 Auswahl des Tag-Gültigkeitsbereichs

Bei "WinStudio" kann entschieden werden, ob ein Tag nur eine einzige Instanz auf einem Projektserver oder Kopien auf jeder lokalen Station anlegt.

- **Server** (Standard): Der Tag wird auf dem Projektserver angelegt und von allen verbundenen Clients (z.B. Thin Client, Secure Viewer) gemeinsam verwendet. Eine Änderung des Tag-Werts beeinflusst das gesamte Projekt.
- **Lokal**: Eine virtuelle Kopie des Tags wird getrennt auf jeder lokalen Station (Server + Clients) vorgehalten, eine Änderung des Tag-Werts beeinflusst nur die Station, auf der die Änderung durchgeführt wurde.

7.3 Erstellen von Datenbank-Tags

7.3.1 Erstellen von Tags im Datenblatt

Mit den folgenden Schritten werden Tags im Datenblatt "Anwendungsvariablen" erstellt:

1. Öffnen Sie den Ordner **Variablen**
2. Öffnen Sie das Datenblatt "Anwendungsvariablen" durch einen Doppelklick auf das Element im Projektexplorer:

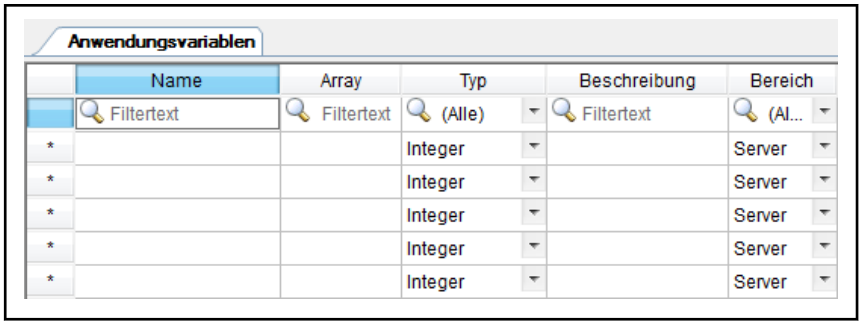


Abb.7-8: Datenblatt Anwendungsvariablen

3. Suchen Sie eine leere Zeile im Datenblatt und konfigurieren Sie die folgenden Felder.



Die Tab-Taste auf der Tastatur kann zum Springen in die nächste Spalte verwendet werden.

- Spalte **Name**: Geben Sie einen Namen gemäß der korrekten Syntax ein.

  (Weitere Informationen finden Sie unter Namenssyntax für Tags auf Seite 121)
- Spalte **Arraygröße**:
  - Für einen Array-Tag geben Sie einen Wert ein, um den Maximalindex des Arrays festzulegen.
  - Für jeden anderen Tag-Typ geben Sie Null (0) ein.
- Combobox **Typ**: Klicken Sie auf den Pfeil, um einen Datentyp (Boolean, Integer, Real oder String) für den Tag aus der Liste auszuwählen.

  (Weitere Informationen finden Sie unter Auswahl eines Tag-Datentyps auf Seite 124)
- Spalte **Beschreibung** (freigestellt): Geben Sie zu Dokumentationszwecken eine Beschreibung ein.
- Combobox **Gültigkeitsbereich**: Klicken Sie auf den Pfeil, um festzulegen, ob der Tag-Wert mit Thin-Client-Stationen gemeinsam genutzt (darauf angezeigt) wird.

  (Weitere Informationen unter Auswahl des Tag-Gültigkeitsbereichs auf Seite 132)

4. Klicken Sie in eine neue Zeile, um einen weiteren Tag zu erstellen.

   Sollen keine weiteren Tags erstellt werden, schließen Sie das Datenblatt "Anwendungsvariablen".

Das folgende Beispiel zeigt eine Reihe von Tags, die in einem Datenblatt "Anwendungsvariablen" konfiguriert wurden.

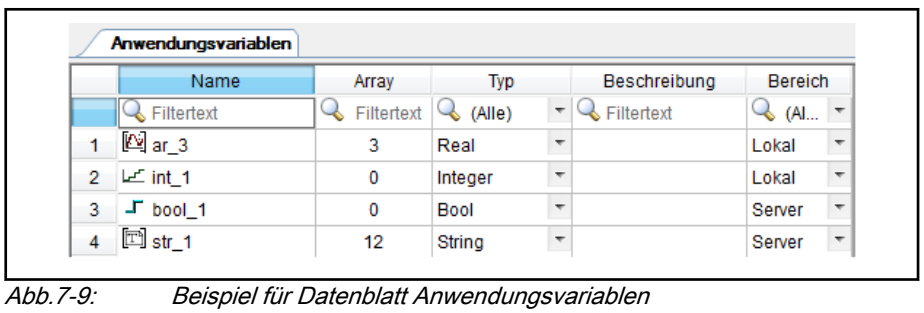


Abb.7-9: Beispiel für Datenblatt Anwendungsvariablen

7.3.2 Erstellen von Tags "On the Fly"

Anstatt zum Erstellen eines neuen Tags jedes Mal das Datenblatt Anwendungsvariablen zu öffnen, können einzelne Tags "On the Fly", also ohne andere Operationen zu unterbrechen, durch Ausführen der folgenden Schritte erstellt werden:

- Wählen Sie im "Projektexplorer" aus dem Kontextmenü des Elements "Anwendungsvariablen" den Eintrag "Variable einfügen" oder
- geben Sie einen neuen Tag-Namen in ein beliebiges Textfeld **Tag/ Ausdruck** ein (verfügbar in Dialog "Objekteigenschaften", Arbeitsblättern usw.). Wenn der "Abfrage"-Dialog fragt, ob eine neuer Tag erstellt werden soll, klicken Sie auf **Ja**.

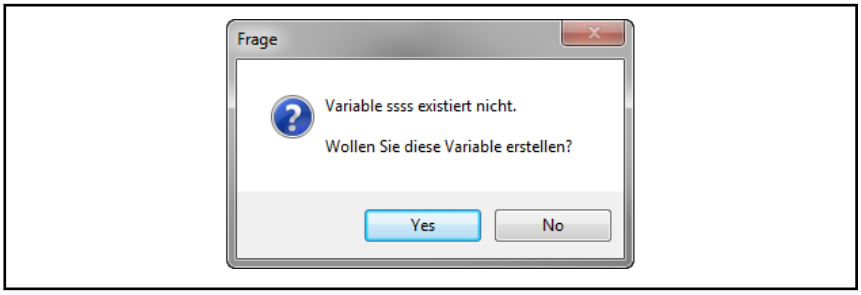


Abb.7-10: Erstellen eines neuen Tags

Jeder dieser Schritte öffnet einen Dialog "Neuer Tag", der je nach Bedarf ausgefüllt werden kann.

(Weitere Informationen unter Anwendungsvariablen, Allgemeines auf Seite 116)

7.3.3 Bearbeiten von Tags

Die Eigenschaften eines Tags können jederzeit während der Entwicklung oder der Laufzeit geändert werden.

Änderung im Datenblatt Anwendungsvariablen

Mit den folgenden Schritten werden Tags aus dem Datenblatt "Anwendungsvariablen" bearbeitet:

1. Öffnen Sie das Datenblatt "Anwendungsvariablen" und suchen Sie den gewünschten Tag.
2. Führen Sie einen Doppelklick in die Spalte mit den zu ändernden Daten aus und geben die neuen Informationen in das Datenblatt ein.
3. Nach dem Bearbeiten schließen Sie das Datenblatt. Ihre Änderungen werden sofort in die Tag-Datenbank übernommen.

3.1 Durch einen Rechtsklick und Auswählen der entsprechenden Option aus dem Kontextmenü können Daten im Datenblatt "Anwendungsvariablen" sortiert und/oder zusätzliche Spalten zu/aus dem Blatt hinzugefügt/entfernt werden

3.2 Sie können einen Rechtsklick auf eine Tag-Eigenschaft ausführen und Standard-Windows-Befehle zum Ausschneiden (<Strg>+<Z>X), Kopieren (<Strg>+<C>) oder Einfügen (<Strg>+<V>) eines Tags und seiner Eigenschaften verwenden

3.3 Sie können die letzten Änderungen in einem Feld rückgängig machen (<Strg>+<Z>)

7.4 Erstellen von Klassen

Zum Erstellen eines neuen Klassen-Tags:

1. Öffnen Sie das Kontextmenü des Elements "Klassen" und wählen Sie den Eintrag **Klasse einfügen** aus
2. Wenn der Dialog "Klasse einfügen" angezeigt wird, geben Sie im Textfeld **Name** einen Namen ein

   (Details siehe Namenssyntax für Tags auf Seite 121 und Auswahl des Tag-Typs auf Seite 122)

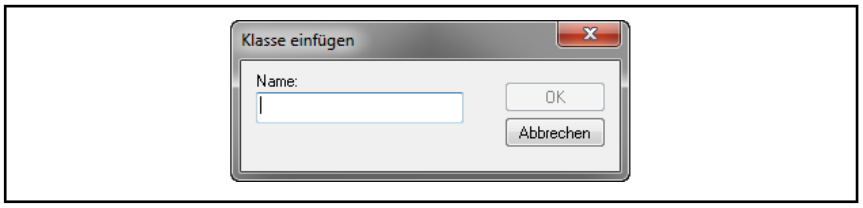


Abb.7-11: Dialog Klasse einfügen

3. Schließen Sie den Dialog "Klasse einfügen", indem Sie auf **OK** klicken. Es wird ein neues Element im Projektexplorer angelegt. Öffnen Sie dieses durch einen Doppelklick.

   Die Spalten im Arbeitsblatt werden wie folgt konfiguriert:

   - **Name**: Name für Klassenelemente eingeben.
   - Auswahlliste **Typ**: Klicken Sie auf den Pfeil, um den Datentyp (Boolean, Integer, Real oder String) des Klassenelements aus der Liste auszuwählen.
   - **Beschreibung** (optional): Geben Sie zu Dokumentationszwecken eine Beschreibung für das Klassenelement ein.

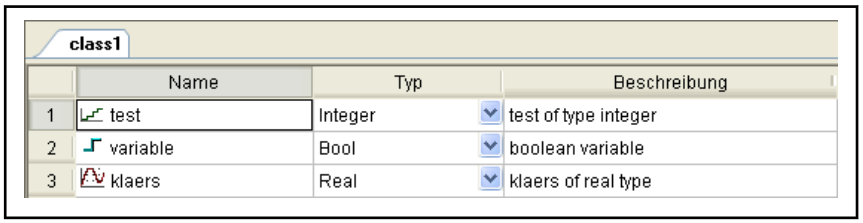


Abb.7-12: Beispiel für Arbeitsblatt "Klasse"

4. Klicken Sie auf die nächste leere Zeile und geben Sie die Daten für das nächste Klassenelement ein, das in dieser Klasse enthalten sein soll. Wenn alle Elemente hinzugefügt wurden, kann das Arbeitsblatt "Klasse" geschlossen werden.
5. Anschließend verwenden Sie die Anleitung unter Erstellen von Tags im Datenblatt auf Seite 132 zum Erstellen und Verbinden eines Tags mit der neuen Klasse.

   Bitte beachten Sie, dass beim Anklicken der Pfeil-Schaltfläche zur Ansicht der Typ-Liste der neue Klassenname ("CTank") enthalten ist (siehe Zeile 5 in der folgenden Abbildung). Wählen Sie den Klassennamen aus dieser Liste aus.

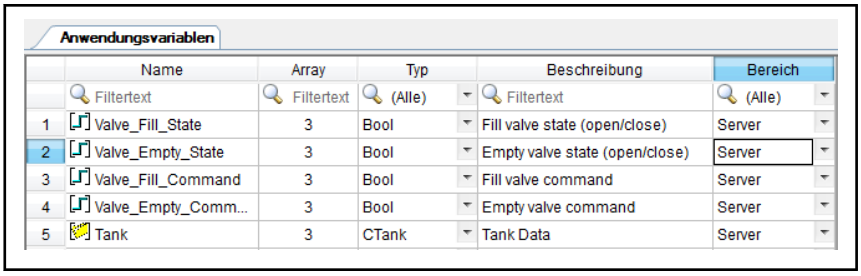


Abb.7-13: Erstellen des Klasse-Tags Tank

6. Zum Abschluss speichern Sie Ihre Arbeit und schließen das Arbeitsblatt.

7.5 Festlegen von Tag-Eigenschaften

7.5.1 Tag-Eigenschaften und -Parameter

Ein Tag-Typ besitzt folgende Eigenschaften:

- **Parameter**
- **Alarm-Eigenschaften**
- **Historie-Eigenschaften**

Zum Konfigurieren dieser Eigenschaften eines Tags öffnen Sie das Werkzeugfenster "Querverweise", geben Sie im Textfeld den Tag-Namen ein und klicken Sie auf die Schaltfläche **Tag-Eigenschaften**.

Der Dialog "Variablen-Eigenschaften" wird angezeigt:

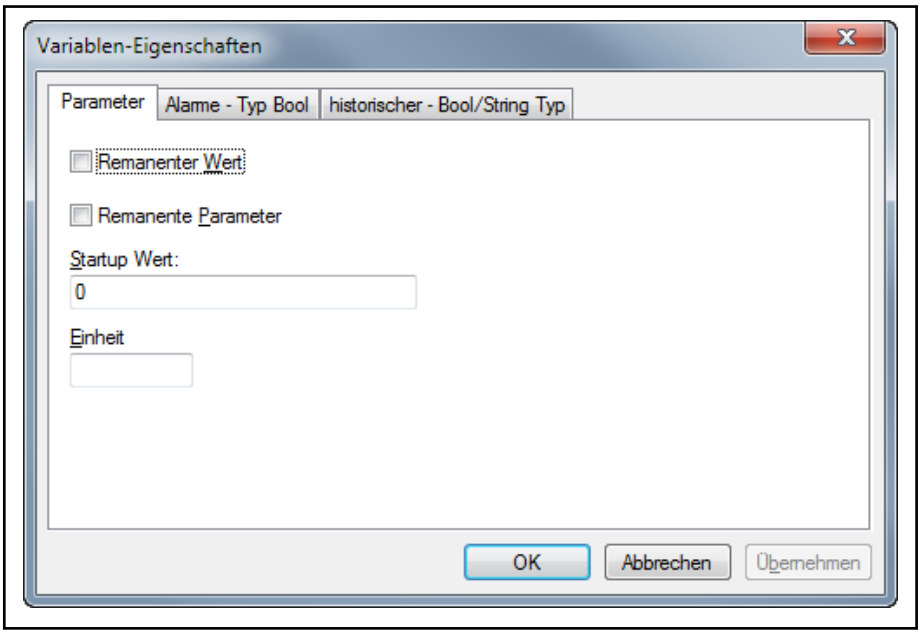


Abb.7-14: Dialog Tag-Eigenschaften für einen Boolean-Tag

Verwenden Sie die Parameter in diesem Dialog zum Konfigurieren der verschiedenen Tag-Eigenschaften.

Die Anleitungen zum Konfigurieren folgen typabhängig:

Tag-Parameter vom Typ String

Legen Sie die Parameter vom Typ String im Dialog "Tag-Eigenschaften" wie folgt fest:

- **Remanenter Wert**: Aktivieren Sie das Kästchen, um den Tag-Wert für den Fall eines unerwarteten Systemabsturzes dauerhaft zu speichern.

So kann bei einem Neustart der zuletzt gespeicherte Wert verwendet werden.

- **Remanente Parameter**: Aktivieren (klicken) Sie das Kästchen, um die Laufzeitänderungen in den Tag-Eigenschaften zu speichern. Nur bestimmte Eigenschaften können gespeichert werden.

  Eine vollständige Liste dazu finden Sie in Liste der Tag-Eigenschaften auf Seite 142.

HINWEIS	**Das Aktivieren einer der speichernden Optionen für Tags mit sich häufig ändernden Werten kann zu einem hohen Aufkommen an Festplattenzugriffen und damit zu verlangsamter Leistung führen.**

- **Startwert**: Geben Sie einen Tag-Wert für den Start der Anwendung ein.

  Der Tag nimmt diesen Wert an, wenn die Option **Remanenter Wert** deaktiviert wird.

- **Einheit**: Geben Sie als Kurzbeschreibung oder Bezug eine beliebige Zeichenfolge (bis zu 9 Zeichen) ein.

  Auf diesen Tag kann während der Laufzeit zugegriffen werden.

☞ "WinStudio" akzeptiert keine schreibenden Werte, die außerhalb des in den Feldern **Min** und **Max** festgelegten Arrays liegen.

Zusätzlich erzeugt "WinStudio" eine Meldung im Ausgabefenster LogWin auf Seite 93, um anzuzeigen, dass das System versucht hat, einen Wert außerhalb des festgelegten Arrays zu schreiben

Tag-Parameter von Typ Integer und Real

☞ Beschreibungen von **Remanenter Wert**, **Remanente Parameter**, **Startwert** und Parameter **Einheit** siehe "Tag-Parameter vom Typ String" auf Seite 136

Legen Sie die Parameter vom Typ Integer und Real im Dialog "Tag-Eigenschaften" wie folgt fest:

- *Bereich Maßeinheiten*
  - **Min**: Geben Sie einen Mindestwert für den Tag in Maßeinheiten ein. Auf diesen Tag kann während der Laufzeit zugegriffen werden.
  - **Max**: Geben Sie einen Höchstwert für den Tag in Maßeinheiten ein. Auf diesen Tag kann während der Laufzeit zugegriffen werden.
  - **Einheit**: Geben Sie als Kurzbeschreibung oder Bezug für den Tag eine beliebige Zeichenfolge (bis zu 9 Zeichen) ein. Auf diesen Tag kann während der Laufzeit zugegriffen werden.
- *Bereich Signalzustände*
  - **Unempfindlichkeitsbereich (Totband)**: Aktivieren (Anklicken) des Kästchens zum Einfügen des Totband-Werts für einen Tag.

    Der Totband-Wert ist eine Abweichung um einen Zentralwert für den Tag, für die kein Alarm ausgelöst wird.
  - **Glätten**: Aktivieren (klicken) Sie das Kästchen, um die Änderungsrate für die Tag-Eigenschaften zu speichern.

    Es können nur Integer- und Real-Tags verwendet werden.

Wenn zum Beispiel die Option **Glätten** für den Tag **FUELLSTAND1** mit dem Wert = 50 gewählt wurde, speichert das System in der nächsten Suche – wenn **FUELLSTAND1** sich auf 60 ändert – den Mittelwert von 50 + 60 in der Datenbank, also den neuen Wert = 55.

Tag-Parameter vom Typ Boolean



Beschreibungen von **Remanenter Wert**, **Remanente Parameter**, **Startwert** und Parameter **Einheit** siehe "Tag-Parameter vom Typ String" auf Seite 136

Siehe auch:

- Verwenden von Tag-Eigenschaften: Alarme auf Seite 138
- Verwenden von Tag-Eigenschaften: Historie auf Seite 141

7.5.2 Verwenden von Tag-Eigenschaften: Alarme

Verwenden Sie den Dialog Tag-Eigenschaften auf Seite 153 zur Ansicht der konfigurierten Alarme für einen gewählten Tag. "WinStudio" deaktiviert diesen Befehl, wenn Alarm-Arbeitsblätter geöffnet sind.

Vor der Verwendung dieser Dialoge sollten die Alarmgruppen bereits erstellt sein.

ALARMTYPEN:

- **HiHi**: ein sehr hoher Alarm liegt vor
- **Hi**: ein hoher Alarm liegt vor
- **Lo**: ein niedriger Alarm liegt vor
- **LoLo**: ein sehr niedriger Alarm liegt vor
- **Rate**: ein Alarm aufgrund der Änderungsrate liegt vor
- **Abweichung**: Ein Alarm aufgrund einer Abweichung von einem gegebenen Sollwert liegt vor

Beispiel:

Wenn SollWert = 50, Abweichung + = 5, Abweichung − = 5 und Abweichung Tote Zone = 0,5;

erzeugt "WinStudio" einen Alarm, wenn temp ≥ 55 oder temp ≤ 45 und

kehrt zum Normalzustand zurück, wenn temp ≤ 54,5 oder temp ≥ 45,5.

ALARMGRENZWERTE:

- **HiHiLimit**: Beim Erstellen von sehr hohen Alarmen im Dialog "Tag-Eigenschaften" wird dieses Feld zum Angeben der Grenzwerte verwendet.

  Auf dieses Feld kann während der Laufzeit für „fliegende“ Änderungen zugegriffen werden.
- **HiLimit**: Beim Erstellen von hohen Alarmen im Dialog "Tag-Eigenschaften" wird dieses Feld zum Angeben der Grenzwerte verwendet.

  Auf dieses Feld kann während der Laufzeit für „fliegende“ Änderungen zugegriffen werden.
- **LoLimit**: Beim Erstellen von niedrigen Alarmen im Dialog "Tag-Eigenschaften" wird dieses Feld zum Angeben der Grenzwerte verwendet.

  Auf dieses Feld kann während der Laufzeit für „fliegende“ Änderungen zugegriffen werden.

- **LoLoLimit**: Beim Erstellen von sehr niedrigen Alarmen im Dialog "Tag-Eigenschaften" wird dieses Feld zum Angeben der Grenzwerte verwendet.

  Auf dieses Feld kann während der Laufzeit für „fliegende“ Änderungen zugegriffen werden.
- **AbwSollwert**: Bezugspunkt für eine Tag-Wertabweichung, die einen Alarm auslöst. Definition der Alarmmeldung im Dialog "Tag-Eigenschaften" oder in einem "Alarm"-Arbeitsblatt.

  Auf dieses Feld kann während der Laufzeit zugegriffen werden.
- **Abw+Limit**: Grenzwertabweichung zu einem höheren Wert als dem **AbwSollwert** in einer Tag-Eigenschaft, der einen Alarm auslöst. Definition der Alarmmeldung im Dialog "Tag-Eigenschaften" oder in einem "Alarm"-Arbeitsblatt.

  Auf dieses Feld kann während der Laufzeit zugegriffen werden.
- **Abw+Limit**: Grenzwertabweichung zu einem niedrigeren Wert als dem **AbwSollwert** in einer Tag-Eigenschaft, der einen Alarm auslöst. Definition der Alarmmeldung im Dialog "Tag-Eigenschaften" oder in einem "Alarm"-Arbeitsblatt.

  Auf dieses Feld kann während der Laufzeit zugegriffen werden.
- **RateGrenzwert**: Grenzwert einer Veränderung der Rate in einem Tag-Wert, der einen Alarm auslöst. Definition der Alarmmeldung im Dialog "Tag-Eigenschaften" oder in einem "Alarm"-Arbeitsblatt.

  Auf dieses Feld kann während der Laufzeit zugegriffen werden.

Alarme für Tags vom Typ Real oder Integer

Legen Sie im Dialog "Tag-Eigenschaften" einen Alarm für Tags vom Typ Integer und Real wie folgt fest:

- **Alarme freigegeben**: Kästchen klicken (aktivieren), um gemäß Konfiguration freizugeben
- **Fern Best Tag**: Eingabe eines Tags zur Bestätigung des Alarms
- **Tote Zone**: Eingabe eines Werts für den Filter für die Alarmnormalisierung.

Beispiel:

Wenn der Tag TEMP1 mit einem Alarm mit Grenzwert Hi = 90 und Tote Zone = 5 konfiguriert ist,

erzeugt "WinStudio" einen Alarm, wenn TEMP1 ≥ 90 ist und

kehrt zum Normalzustand zurück, wenn TEMP1 ≤ 85 ist.

- **Übersetzung freigegeben**: Kästchen anklicken (aktivieren), um das Übersetzen von Meldungen freizugeben, das auf der Registerkarte "Optionen" im Dialog Projekteinstellungen auf Seite 67 definiert ist.

  Weitere Informationen zu Übersetzungen unter Übersetzungswerkzeuge auf Seite 437.

☞ "WinStudio" speichert Alarmmeldungen mit aktiviertem Attribut **ÜbersetzungFreigabe** in einer Datei mit dem Namen "Alarm.TXT" im Verzeichnis "\DATABASE\" Ihres Projekts.

- **HiHi (HiHiLimit)**: Kästchen anklicken (aktivieren), um anzuzeigen, dass ein sehr hoher Alarm vorliegt. Zugriff während der Laufzeit möglich.

- **Hi (HiLimit)**: Kästchen anklicken (aktivieren), um anzuzeigen, dass ein hoher Alarm vorliegt. Zugriff während der Laufzeit möglich.
- **Lo (LoLimit)**: Kästchen anklicken (aktivieren), um anzuzeigen, dass ein niedriger Alarm vorliegt. Zugriff während der Laufzeit möglich.
- **LoLo (LoLoLimit)**: Kästchen anklicken (aktivieren), um anzuzeigen, dass ein sehr niedriger Alarm vorliegt. Zugriff während der Laufzeit möglich.
- **Rate (RateLimit)**: Kästchen anklicken (aktivieren), um anzuzeigen, dass ein Rate-Alarm vorliegt. Zugriff während der Laufzeit möglich.
- **Abweichung+**: Kästchen anklicken (aktivieren), um anzuzeigen, dass ein Abweichungs-Alarm vorliegt. Zugriff während der Laufzeit möglich.
- **Abweichung-**: Kästchen anklicken (aktivieren), um anzuzeigen, dass ein Abweichungs-Alarm vorliegt. Zugriff während der Laufzeit möglich.
- **Abweichung Sollwert**: Eingabe eines Bezugspunkts für die Abweichung. Zugriff während der Laufzeit möglich.
- **Abweichung Tote Zone**: Eingabe eines Bezugswerts für die Abweichung.

Alarme für Tags vom Typ Boolean

Verwenden Sie die Registerkarte "Alarme Typ Bool" im Dialog "Tag-Eigenschaften" zur Festlegung der Alarmeigenschaften eines Tags vom Typ Boolean.

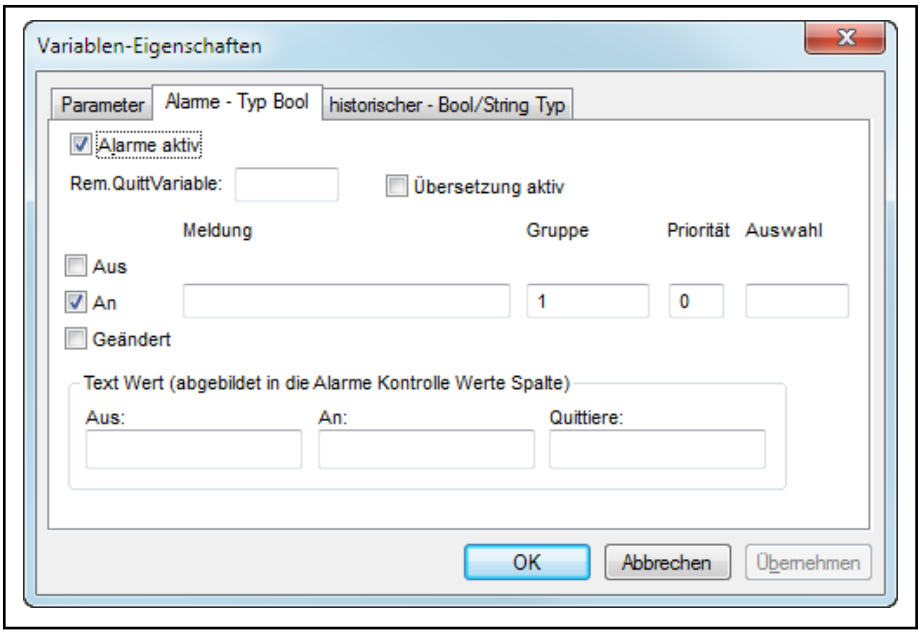


Abb.7-15: Registerkarte Alarme Typ BOOL

Konfigurieren Sie die Parameter wie folgt:

- **Alarme aktivieren**: Kästchen klicken (aktivieren), um gemäß Konfiguration freizugeben.
- **Fern-Bestätigungsvariable**: Geben Sie einen Tag ein, um die Fernbestätigung für Alarme freizugeben, die bei Änderung des Tag-Werts auftritt.
- **Übersetzung aktivieren**: Kästchen anklicken (aktivieren), um das Übersetzen von Meldungen freizugeben, das auf der Registerkarte "Optionen" im Dialog Projekteinstellungen auf Seite 67 definiert ist.

  Weitere Informationen zu Übersetzungen unter Übersetzungswerkzeuge auf Seite 437.



"WinStudio" speichert Alarmmeldungen mit aktiviertem Attribut **ÜbersetzungFreigabe** in einer Datei mit dem Namen **Alarm.TXT** im Verzeichnis **\DATABASE** Ihres Projekts:

- **Aus**: Kästchen anklicken (aktivieren), um immer eine Alarmmeldung zu erzeugen, wenn der Tag-Wert gleich Null ist.
- **Ein**: Kästchen anklicken (aktivieren), um immer eine Alarmmeldung zu erzeugen, wenn der Tag-Wert gleich Eins ist.
- **Geändert**: Kästchen anklicken (aktivieren), um immer eine Alarmmeldung zu erzeugen, wenn der Tag-Wert sich ändert.
- **Text-Wert**: Verwenden Sie die Felder **Aus**, **Ein** und **Best** zum Konfigurieren von Mnemonics (zum Beispiel "Geschlossen" oder "Offen") für die Alarmzustände **Aus**, **Ein** und **Best** (nur für Boolean-Tags).

  Während der Laufzeit zeigt das Alarmsteuerungsobjekt diesen Text in der Spalte "Wert" der Alarmmeldung an.

  Auf diesen Text kann auch mit Hilfe der Tag-Felder auf Seite 142 zugegriffen werden.



Wenn keine Mnemonics konfiguriert werden, zeigt das Alarmsteuerungsobjekt in der Spalte "Wert" den Tag-Wert ("0" oder "1") an.

7.5.3 Verwenden von Tag-Eigenschaften: Historie

Verwenden Sie den Dialog Tag-Eigenschaften auf Seite 153 zur Ansicht der Historie für einen gewählten Tag. "WinStudio" deaktiviert diesen Befehl, wenn Trend-Arbeitsblätter geöffnet sind.

Vor der Verwendung dieser Dialoge sollten die Trendgruppen bereits erstellt sein.



Die Historie unterstützt keine Tags vom Typ String.

Siehe Ordner Rezepturen auf Seite 361 für weitere Informationen zum Speichern von String-Werten.

Historie vom Typ Integer und Real

Geben Sie im Dialog "Tag-Eigenschaften" die Historie für Tags vom Typ Integer und Real wie folgt frei:

- **Historie aktivieren**: Kästchen anklicken (aktivieren) zur Freigabe der Speicherung der Beispielwerte für den gewählten Tag.
- **Gruppennummer**: Festlegung der Gruppennummer, mit der dieser Tag verbunden ist.
- **Log Totband**: Legen Sie einen Beispielwert fest, der aufgenommen wird, wenn der Veränderungswert größer oder gleich dem Wert **Log Totband** ist.

Historie vom Typ Boolean

Geben Sie im Dialog "Tag-Eigenschaften" die Historie für Tags vom Typ Boolean wie folgt frei:

- **Historie aktivieren**: Kästchen anklicken (aktivieren) zur Freigabe der Speicherung der Beispielwerte für den gewählten Tag.
- **Gruppennummer**: Festlegung der Gruppennummer, mit der dieser Tag verbunden ist.

7.5.4 Liste der Tag-Eigenschaften

Tag-Eigenschaften (auch als „Tag-Felder“ bekannt) sind Metadaten, die mit jedem Tag in der Datenbank verbunden sind. Die meisten dieser Eigenschaften können im Dialog "Tag-Eigenschaften" festgelegt werden, der durch Klicken auf die Schaltfläche **Tag-Eigenschaften** im Werkzeugfenster "Querverweise" geöffnet werden kann.

Um während der Laufzeit auf eine Tag-Eigenschaft zuzugreifen, wird die folgende Syntax (ohne Leerzeichen) überall dort verwendet, wo normalerweise ein Tag angegeben würde:

```
tag_name->eigenschaft_name
```

Auf die folgenden Tag-Eigenschaften können während der Laufzeit zugegriffen werden:

Tag-Eigenschaft	Beschreibung	S oder S/L	Datentyp	Verfügbar auf Datentyp...				Speichernd
				Bool	Int	Real	Str	
Name	Der Name des Tags wie in der Projekt-Tag-Datenbank konfiguriert.	R	String, bis zu 32 Zeichen	J	J	J	J	n/a
MemberName	Der Name des Klassenelement in einer korrekt konfigurierten Klasse auf Seite 118. **HINWEIS:** Die Syntax muss lauten: **Klasse.Member->MemberName** Beispiel: **Tank.Fst->MemberName = Fst**	L	String, bis zu 32 Zeichen	J	J	J	J	n/a
Size	Arraygröße, wenn der Tag kein Array-Tag ist, gibt er den Wert 0 zurück	L	Integer	J	J	J	J	n/a
Index	Die Indexnummer eines Elements in einem Array auf Seite 125. (Ein Array ist jeder Tag mit größer 0.) **HINWEIS:**Die Syntax muss lauten: **Tag[Index]->Index** Beispiel: **Tag[1]->Index = 1**	L	Integer	J	J	J	J	n/a
Description	Die Beschreibung des Tags, konfiguriert im Tag-Datenblatt.	L	String	J	J	J	J	J
Quality	Tag-Qualität (192=GOOD; 0=BAD). Das Projekt aktualisiert dieses Feld jedes Mal, wenn der Tag das Ergebnis eines Ausdrucks oder einen Wert von einem Kommunikations-Task erhält (z.B. einem Treiber oder OPC). Wenn der Ausdruck ungültig ist (z.B. Division durch Null) oder wenn es im Zusammenhang mit dem Tag einen Kommunikationsfehler gibt, setzt das Projekt die Qualität auf BAD.	L	Integer	J	J	J	J	N
TimeStamp	Uhrzeit und Datum, wann der Wert des Tags zuletzt geändert wurde.	L	String	J	J	J	J	N

Tag-Eigenschaft	Beschreibung	S oder S/L	Datentyp	Verfügbar auf Datentyp...				Speichernd
				Bool	Int	Real	Str	
Blocked	Diese Eigenschaft kann zwei Werte haben: Das ist hilfreich, wenn man alle Aktionen im Zusammenhang mit einem bestimmten Tag dynamisch deaktivieren will. Selbst wenn ein Tag blockiert ist, zählt er für Lizenzzwecke trotzdem noch für die Gesamtzahl der Tags. • 0: Der Tag ist blockiert und wird von allen Laufzeit-Tasks ignoriert. Er wird effektiv aus der Projekt-Datenbank entfernt. • 1: Der Tag ist deblockiert und alle Laufzeit-Tasks können normal darauf zugreifen.	L/S	Boolean	J	J	J	J	N
Unit	Eine kurze Beschreibung (bis zu 9 Zeichen) der Maßeinheit für den Tag-Wert. Zum Beispiel **Kg**, **kWh**, **Pa**.	L/S	String, bis zu 9 Zeichen	J	J	J	J	J
Max	Der Höchstwert, der während der Laufzeit in den Tag geschrieben werden kann.	L/S	Real	N	J	J	N	J
Min	Der Mindestwert, der während der Laufzeit in den Tag geschrieben werden kann.	L/S	Real	N	J	J	N	J
B0 … B31	Wert (0 oder 1) eines beliebigen der 32 Bits (b0, b1, b2, ... b31) eines Integer-Tags. (B0: LSB B31: MSB)	L/S	Boolean	N	J	N	N	N
DisplayValue	Ein umgerechneter Tag-Wert, der nur auf dem Bildschirm angezeigt wird. **DisplayValue = (Value / UnitDiv) + UnitAdd** Wird verwendet, wenn die tatsächlichen Tag-Werte eine Maßeinheit haben (siehe **Unit** oben), aber auf dem Bildschirm in einer anderen Maßeinheit angezeigt werden soll (siehe **DisplayUnit** unten) Zum Beispiel Grad Celsius und Grad Fahrenheit. Wenn der Anwender **DisplayValue** während der Laufzeit verändert, dann wird die Umrechnung umgekehrt, bevor die Änderung tatsächlich in den Tag geschrieben wird. **Value = (DisplayValue − UnitAdd) * UnitDiv**	L/S	Real	N	J	J	N	n/a
DisplayUnit	Eine kurze Beschreibung (bis zu 9 Zeichen) der Maßeinheit für **DisplayValue**. **HINWEIS:** Diese Eigenschaft kann nur durch Verwendung der Funktionen SetDisplayUnit auf Seite 682 und SetTagDisplayUnit auf Seite 683 festgelegt werden.	L	String, bis zu 9 Zeichen	N	J	J	N	N

Arbeiten mit Tags

Tag-Eigenschaft	Beschreibung	S oder S/L	Datentyp	Verfügbar auf Datentyp...				Speichernd
				Bool	Int	Real	Str	
UnitDiv	Zahl, durch die der Tag-Wert dividiert wird, um **DisplayValue** zu erhalten. Um keine Division durchzuführen, sollte **UnitDiv** 1 sein. **HINWEIS:** Diese Eigenschaft kann nur durch Verwendung der Funktionen SetDisplayUnit auf Seite 682 und SetTagDisplayUnit auf Seite 683 festgelegt werden.	L	Real	N	J	J	N	N
UnitAdd	Zahl, die zum Tag-Wert addiert wird, um **AnzeigeWert** zu erhalten. Um keine Addition durchzuführen, sollte **EinheitAdd** 0 sein. **HINWEIS:** Diese Eigenschaft kann nur durch Verwendung der Funktionen SetDisplayUnit auf Seite 682 und SetTagDisplayUnit auf Seite 683 festgelegt werden.	L	Real	N	J	J	N	N
DisplayMax	Der Höchstwert, der während der Laufzeit in **DisplayValue** eingegeben werden kann. **DisplayMax = (Max / UnitDiv) + UnitAdd** Wenn **DisplayMax** während der Laufzeit verändert wird, ändert sich **Max** ebenfalls wie folgt: **Max = (DisplayMax − UnitAdd) * UnitDiv**	L/S	Real	N	J	J	N	N
DisplayMin	Der Mindestwert, der während der Laufzeit in **DisplayValue** eingegeben werden kann. **DisplayMin = (Min / UnitDiv) + UnitAdd** Wenn **DisplayMin** während der Laufzeit verändert wird, ändert sich **Min** ebenfalls wie folgt: **Min = (DisplayMin − UnitAdd) * UnitDiv**	L/S	Real	N	J	J	N	N
HiHiLimit	Grenzwert für den HiHi-Alarm	L/S	Real	N	J	J	N	J
HiLimit	Grenzwert für den Hi-Alarm	L/S	Real	N	J	J	N	J
LoLimit	Grenzwert für den Lo-Alarm	L/S	Real	N	J	J	N	J
LoLoLimit	Grenzwert für den LoLo-Alarm	L/S	Real	N	J	J	N	J
RateLimit	Grenzwert für den Rate-Alarm	L/S	Real	N	J	J	N	J
AbwSollwert	Sollwert für Abweichungsalarme.	L/S	Real	N	J	J	N	n/a
AbwPLimit	Grenzwert für den Alarm Abweichung+	L/S	Real	N	J	J	N	J
AbwMLimit	Grenzwert für den Alarm Abweichung-	L/S	Real	N	J	J	N	J
HiHi	Bei 0 ist der HiHi-Alarm nicht aktiv. Bei 1 ist der HiHi-Alarm aktiv.	L	Boolean	J	J	J	N	n/a
Hi	Bei 0 ist der Hi-Alarm nicht aktiv. Bei 1 ist der Hi-Alarm aktiv.	L	Boolean	J	J	J	N	n/a
Lo	Bei 0 ist der Lo-Alarm nicht aktiv. Bei 1 ist der Lo-Alarm aktiv.	L	Boolean	J	J	J	N	n/a
LoLo	Wenn 0 ist der Lo-Alarm nicht aktiv. Bei 1 ist der LoLo-Alarm aktiv.	L	Boolean	J	J	J	N	n/a

Tag-Eigenschaft	Beschreibung	S oder S/L	Datentyp	Verfügbar auf Datentyp...				Speichernd
				Bool	Int	Real	Str	
Rate	Bei 0 ist der Rate-Alarm nicht aktiv. Bei 1 ist der Rate-Alarm aktiv.	L	Boolean	J	J	J	N	n/a
AbwP	Bei 0 ist der Alarm Abweichung+ nicht aktiv. Bei 1 ist der AbwP-Alarm aktiv.	L	Boolean	N	J	J	N	n/a
AbwM	Bei 0 ist der Alarm Abweichung- nicht aktiv. Bei 1 ist der AbwM-Alarm aktiv.	L	Boolean	N	J	J	N	n/a
AlrStatus	Ganzzahlwert mit dem Status des aktuell aktiven Alarms in Zusammenhang mit dem Tag. Jedes Bit dieses Ganzzahlwerts gibt einen bestimmten Status an: • Bit 0 (LSB): HiHi-Alarm aktiv • Bit 1: Hi-Alarm aktiv • Bit 2: Lo-Alarm aktiv • Bit 3: LoLo-Alarm aktiv • Bit 4: Rate-Alarm aktiv • Bit 5: Alarm Abweichung+ aktiv • Bit 6: Alarm Abweichung- aktiv Beispiele: Wenn der Tag->AlrStatus den Wert 2 zurückgibt, bedeutet das, dass der Hi-Alarm aktiv ist. Wenn der Wert 3 zurückgegeben wird, bedeutet das, dass der HiHi- und der Hi-Alarm gleichzeitig aktiv sind. Gibt diese Eigenschaft den Wert 0 zurück, bedeutet das, dass es in Verbindung mit diesem Tag keine aktiven Alarme gibt. Für Boolean-Tags können nur die Werte 1 (Bit 1), 4 (Bit 2) oder 16 (Bit 4) zurückgegeben werden.	L	Integer	J	J	J	N	N
Ack	Diese Eigenschaft kann zwei Werte haben: Dies funktioniert als globale Bestätigung für den Tag und geht nur dann auf 0, wenn alle Alarme für den Tag bestätigt wurden. • 0: Für diesen Tag stehen keine Alarme an, die bestätigt werden müssen. • 1: Für diesen Tag gibt es mindestens einen Alarm, der bestätigt werden muss.	L	Boolean	J	J	J	N	N
UnAck	Diese Eigenschaft kann zwei Werte haben: Wenn dieser Wert manuell auf 1 gesetzt wird, werden ggf. aktive Alarme bestätigt. Der Wert dieser Eigenschaft ist immer das Gegenteil der Eigenschaft **Ack**. • 0: Für diesen Tag gibt es mindestens einen Alarm, der bestätigt werden muss. • 1: Für diesen Tag stehen keine Alarme an, die bestätigt werden müssen.	L/S	Boolean	J	J	J	N	N

Arbeiten mit Tags

Tag-Eigenschaft	Beschreibung	S oder S/L	Datentyp	Verfügbar auf Datentyp...				Speichernd
				Bool	Int	Real	Str	
AlrAckValue	Text zum Status **Bestätigt** eines Boolean-Tag. Dieser Text wird in der Wert-Spalte einer Alarm-/Ereignis-Steuerung auf Seite 290 angezeigt. Dieser Text kann auch im Dialog Tag-Eigenschaften auf Seite 138 (Alarme – Typ Bool) bearbeitet werden.	L/S	String, bis zu 32 Zeichen	J	N	N	N	J
AlrOffValue	Text zum Status **Normalisiert** eines Boolean-Tag. Dieser Text wird in der Wert-Spalte einer Alarm-/Ereignis-Steuerung auf Seite 290 angezeigt. Dieser Text kann auch im Dialog Tag-Eigenschaften auf Seite 138 (Alarme – Typ Bool) bearbeitet werden.	L/S	String, bis zu 32 Zeichen	J	N	N	N	J
AlrOnValue	Text zum Status **Aktiv** eines Boolean-Tag. Dieser Text wird in der Wert-Spalte einer Alarm-/Ereignis-Steuerung auf Seite 290 angezeigt. Dieser Text kann auch im Dialog Tag-Eigenschaften auf Seite 138 (Alarme – Typ Bool) bearbeitet werden.	L/S	String, bis zu 32 Zeichen	J	N	N	N	J
AlrDisable	Diese Eigenschaft kann zwei Werte haben: • 0: Die Alarme zu diesem Tag sind aktiviert. Das bedeutet, dass der Alarm bei Auftreten einer Alarmbedingung aktiv wird. • 1: Die Alarme zu diesem Tag sind deaktiviert. Das bedeutet, dass der Alarm bei Auftreten einer Alarmbedingung nicht aktiv wird.	L/S	Boolean	J	J	J	N	N

- Wenn eine Eigenschaft unter „speichernd“ als „n/a“ gekennzeichnet ist, ist diese in der Tag-Definition (z.B. Name, Größe) impliziert oder der Wert der Eigenschaft wird während der Laufzeit kontinuierlich abgeleitet (z.B. Alarmaktivierung, AnzeigeWert). Um Speichern für einen Tag freizugeben, wird die Option **Remanente Parameter** im Dialog Tag-Eigenschaften auf Seite 136 ausgewählt.
- Wenn das Projekt versucht, einen Wert außerhalb des in den Eigenschaften **Min** und **Max** festgelegten Arrays zu schreiben, akzeptiert die Tag-Datenbank den neuen Wert nicht und schreibt eine Warnmeldung in das Fenster Ausgabe auf Seite 93. Wenn beide Eigenschaften **Min** und **Max** mit dem Wert "0" (Null) konfiguriert sind, wird jeder auf den Tag-Typen angewendete Wert auf den Tag geschrieben.
- Tag-Eigenschaften (wie Bit-Felder) können nicht zum Konfigurieren der Arbeitsblätter Alarm auf Seite 273 oder Trend auf Seite 301 verwendet werden.
- Obwohl Tag-Eigenschaften auf System-Tags auf Seite 121 angewendet werden können, bleiben diese Eigenschaften nicht erhalten, wenn das Projekt auf ein CE-Gerät herunter geladen wird.

7.6 Verwenden von Tags in Ihrem Projekt

Nach dem Hinzufügen eines Tags zur Projektdatenbank kann dieser Tag in Ihrem Projekt verwendet werden, indem er mit Objekten in einem Bild verbunden wird.

Die grundlegende Vorgehensweise zum Verbinden eines Tags mit Bildobjekten besteht aus den folgenden Schritten:

1. Wählen Sie im WinStudio-Bild das Objekt aus, auf das der Tag angewendet werden soll.
2. Klicken Sie auf eine der Schaltflächen in der Gruppe Animationen auf Seite 243 zum Anwenden dieser Animation auf das Objekt.
3. Führen Sie einen Doppelklick auf das Objekt aus, um dessen Dialog Objekteigenschaften auf Seite 178 zu öffnen.
4. Suchen Sie das Textfeld **Tag** für diese Eigenschaft und geben Sie den Tag-Namen in das Feld ein. Name und Position der Tag-Textfelder sind je nach Art der verwendeten Eigenschaft unterschiedlich.

   Zum Beispiel:



Abb.7-16: Anwenden von Tags auf ein Objekt

Ausführliche Anleitungen zum Anwenden von Tags auf Bildobjekte finden sich in der gesamten Dokumentation an den entsprechenden Stellen.

7.7 Löschen eines Tags aus der Projektdatenbank

Ein nicht mehr verwendeter Tag kann durch Löschen der entsprechenden Zeile im Datenblatt "Anwendervariablen" gelöscht werden.



Es wird dringend empfohlen, vor dem Löschen eines Tags das Werkzeug Querverweisliste auf Seite 96 zu verwenden, um sicherzustellen, dass der Tag nicht an anderer Stelle im Projekt verwendet wird.

(Wenn ein noch verwendeter Tag gelöscht wird, kann das Projekt nicht überprüft und ausgeführt werden.)

Korrigieren Sie vor dem Fortfahren alle Bilder oder Arbeitsblätter, in denen der Tag verwendet wird.

1. Projekt anhalten auf Seite 66, falls es läuft
2. Datenblatt Projekt-Tags auf Seite 116 öffnen
3. Führen Sie an beliebiger Stelle im Datenblatt einen Rechtsklick aus und wählen Sie **Sortieren deaktivieren** aus dem Kontextmenü. Tags können nicht gelöscht werden, wenn das Datenblatt sortiert ist. Die Tags werden wieder in ihre Standard-Reihenfolge gebracht, d.h. die Reihenfolge, in der sie erstellt wurden
4. Suchen Sie im Datenblatt die Zeile für den Tag, der gelöscht werden soll
5. Führen Sie einen Rechtsklick auf die Zeile aus und wählen Sie **Zeile löschen** aus dem Kontextmenü aus. (Wenn die Option nicht verfügbar ist, muss noch wie oben beschrieben die Sortierung deaktiviert werden.) Es wird ein Warndialog angezeigt zur Bestätigung dieses Vorgangs
6. Klicken Sie auf **Ja**. Die Zeile wird aus dem Datenblatt gelöscht
7. Datei abspeichern und schließen

7.8 Tags umbenennen und entfernen

7.8.1 Tags global ersetzen

Wenn die Funktion **Variable global ersetzen** auf der WinStudio-Symbolleiste Tag-Eigenschaften angeklickt wird, wird das folgende Fenster angezeigt:

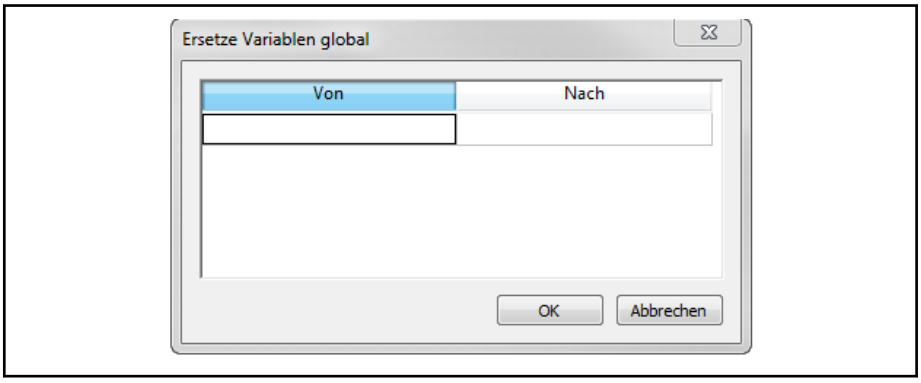


Abb.7-17: Dialog Globales Ersetzen

Aus dem Dialog "Globales Ersetzen" können alle Tags aus allen Dokumenten (Bildern und Arbeitsblättern) im gesamten Projekt ersetzt werden. Es kann sowohl die Spalte **Von** als auch die Spalte **Nach** ersetzt werden.

Beim Ersetzen von zusammengesetzten Tags (Arraygröße > 0 und/oder Typ = Klasse) kann eine bestimmte Array-Position (z.B. **TagA[1]**) oder ein Klassenelement (z.B. **TagB.MemberX**) oder beides (z.B. **TagC[3].MemberY**) konfiguriert werden.

Wenn nur der Hauptname des Tags (z.B. **TagC**) in der Spalte **Von** konfiguriert wird, werden für den in der Spalte **Nach** konfigurierten Tag alle Tags von diesem Haupt-Tag verändert.

Wird ein ungültiger Ersatzwert konfiguriert (z.B. Ersetzen des Tags **Haupt-Tag** aus einem Tag vom Typ Klasse mit einem einfachen Tag (kein Klasse-Tag)), so wird die Schaltfläche **OK** deaktiviert.

Beim Anklicken der Schaltfläche **OK** werden die im Dialog "Globales Ersetzen" konfigurierten Tags in der Reihenfolge ersetzt, in der sie in der Dialogschnittstelle konfiguriert wurden.



Vor dem Ausführen dieses Befehls müssen alle Dokumente (Bildern und Arbeitsblätter) geschlossen werden.

Beim Ändern des Tag-Namens auf dem Datenbank-Arbeitsblatt Projekt-Tags auf Seite 116 fragt "WinStudio", ob dieser Tag im gesamten Projekt ersetzt werden soll.

Die Option **Ersetzen** wird im Menü **Bearbeiten** erstellt. Durch Verwenden dieser Option wird der Dialog "Globales Ersetzen" aufgerufen. Änderungen werden allerdings nur auf das aktuell aktive Bild oder das aktuell aktive Arbeitsblatt angewendet.

7.8.2 Tags in einem Dokument oder Bildobjekt ersetzen

Zum Ersetzen aller in einem Bildobjekt vorkommenden Tags führen Sie einen Doppelklick auf das Objekt aus, um den Dialog "Objekt-Eigenschaften" zu öffnen und klicken dann auf **Ersetzen**.

Zum Ersetzen aller im aktuellen Dokument vorkommender Tags führen Sie eine der beiden Aktionen aus:

- Öffnen Sie das Menü "WinStudio-Bild" im Hauptmenü von IndraWorks und wählen Sie den Eintrag **Variablen und Strings im aktuellen Dokument ersetzen**
- Blenden Sie die Symbolleiste "WinStudio Tabellenfunktionen" ein und klicken Sie dort auf die Schaltfläche **Variablen und Strings im aktuellen Dokument ersetzen**

Im beiden Fällen öffnet sich der Dialog "Ersetzen", der im folgenden beschrieben wird.

Durch Klicken auf die Registerkarte **Ganzer Tag-Name** können ein oder mehrere Tags ersetzt werden. In der linken Spalte **Von** finden Sie die aktuell verwendete Tags. In der rechten Spalte **Nach** können Sie die neuen Namen eintragen.

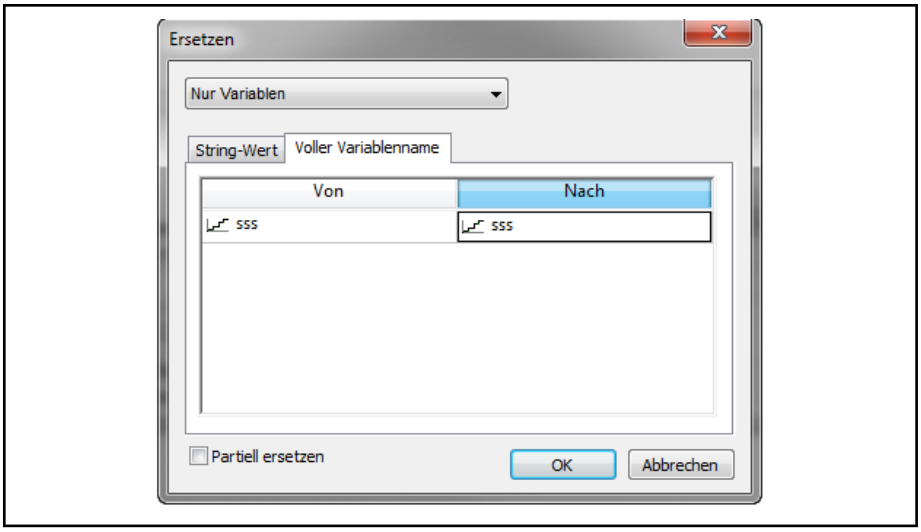


Abb.7-18: Dialog "Ersetzen", Registerkarte"Ganzer Tag-Name"

Bitte beachten Sie, dass hierdurch keine Tags "umbenannt" oder "gelöscht" werden – es werden nur die im Objekt verwendeten Tags durch andere Tags aus der Datenbank ersetzt.

Die Aktivierung der Option "Partiell ersetzen" erleichtert das Ersetzen von Parametern (z.B. Array Index) innerhalb von Klassen und Arrays.

Es können auch eine oder mehrere Zeichenketten (z.B. Schaltflächenbeschriftungen, Beschreibungstexte) durch Klicken auf die Registerkarte **String-Wert** ersetzt werden.

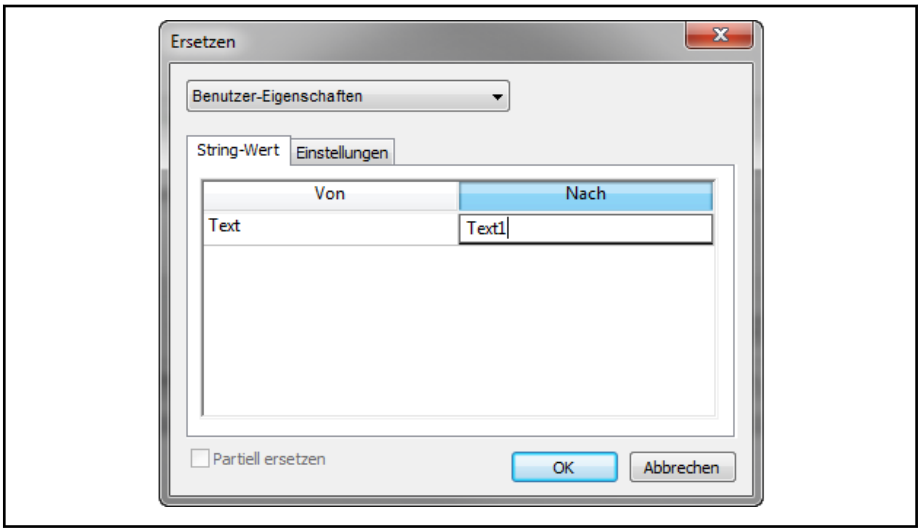


Abb.7-19: Dialog "Ersetzen", Registerkarte"Zeichenkette"

Klicken Sie auf **OK**, wenn Sie alle Einträge bearbeiten haben.

7.8.3 Nicht verwendete Tags aus der Projektdatenbank entfernen

Das Werkzeug **Nicht verwendete Tags entfernen** wird verwendet, um die Projektdatenbank nach nicht verwendeten Tags zu durchsuchen, die dann ausgewählt und entfernt werden können.

„Nicht verwendete Tags“ sind Tags, die in der Projektdatenbank definiert wurden, aber in keinem Bild und keinem Task-Arbeitsblatt verwendet werden.

Da das Projekt über eine begrenzte Anzahl von verfügbaren Tags verfügt (bestimmt durch Ihre Produkt-/Lizenzart), kann es sinnvoll sein, einige oder

alle nicht verwendeten Tags zu entfernen, um die Anzahl der Tags im Projekt zu verringern.

Um nicht verwendete Tags zu suchen und zu entfernen gehen Sie folgendermaßen vor:

1. Speichern und schließen Sie alle offenen Projektbilder und -arbeitsblätter.
2. Führen Sie die Funktion "Anwendung verifizieren" aus dem Kontextmenü des Ordners "WinStudio" aus und drücken Sie im Dialog "Anwendung verifizieren" die Schaltfläche "Prüfen".

   Die Entwicklungsanwendung überprüft automatisch Ihr Projekt. Werden nicht verwendete Tags gefunden, wird ein Hinweis im Meldungsfenster ausgegeben

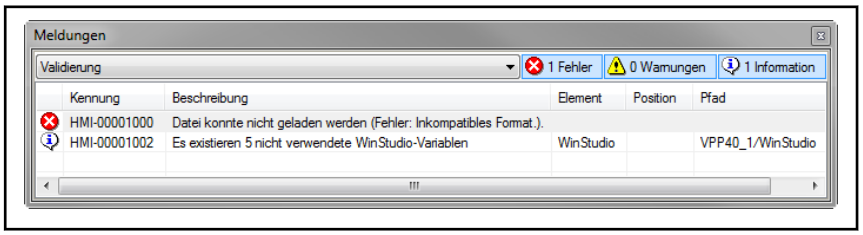


Abb.7-20: Hinweis auf nicht verwendete Tag bei "Anwendung verifizieren"

Doppelklicken Sie auf die entsprechende Zeile im Meldungsfenster. Es öffnet sich folgender Dialog:

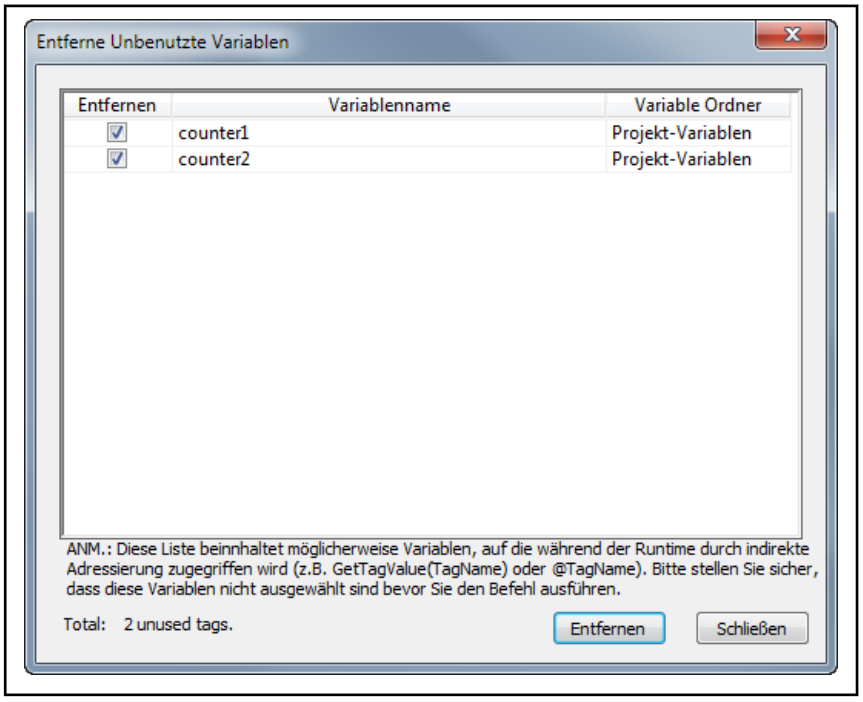


Abb.7-21: Liste nicht verwendeter Tags im Dialog "Nicht verwendete Tags entfernen"

3. *Entscheiden Sie, welche Tags Sie ggf. entfernen möchten*
   - Wenn Sie alle aufgeführten Tags entfernen möchten, klicken Sie auf **Entfernen**
   - Wenn Sie einige der aufgeführten Tags behalten möchten, entfernen Sie die Option „Entfernen“ für diese Tags und klicken dann auf **Entfernen**



Unter den aufgeführten Tags befinden sich ggf. einige Tags, auf die während der Laufzeit mit indirekter Syntax zugegriffen wird (z.B. **HoleTagWert(TagName)** oder **@TagName**, wobei der Wert von **TagName** der Name eines nicht verwendeten Tags ist).

- Wenn Sie keine der aufgeführten Tags entfernen möchten, klicken Sie auf **Schließen**

WinStudio entfernt die ausgewählten Tags und fragt dann, ob Sie das Projekt erneut überprüfen wollen

4. Klicken Sie zum erneuten Überprüfen des Projekts auf **Ja**

7.8.4 Tag-Datenbank zurücksetzen

Zum erneuten Laden der Tag-Datenbank auf die lokale Station führen Sie die Funktion **Tag-Datenbank Zurücksetzen** aus.

Dieser Befehl beeinflusst alle im Ordner "Anwendungsvariablen" gespeicherten Tags.

Diese Option ist hilfreich zum Zurücksetzen der Anwendungsvariablen und zum Wiederherstellen der Werte auf den Stand beim erstmaligen Laden des Projekts.

Wenn das Projekt angehalten wird, die Entwicklungsumgebung aber geöffnet bleibt, so werden die Tags beim erneuten Ausführen des Projekts nicht zurückgesetzt. Daher kann dieser Befehl zum Zurücksetzen dieser Tags vor dem erneuten Ausführen des Projekts verwendet werden.

Wenn dieser Befehl ausgeführt wird, wird der für jeden Tag (Dialog Tag-Eigenschaften auf Seite 153) konfigurierte **Anfangswert** in den entsprechenden Tag geschrieben.

Wenn kein **Anfangswert** für einen numerischen Tag (**Boolean**, **Integer** oder **Real**) konfiguriert wurde, wird der Wert "0" (Null) in den Tag geschrieben.

Wenn kein **Anfangswert** für einen String-Tag konfiguriert wurde, wird der leere Wert (" ") in den Tag geschrieben.

Dieser Befehl ist deaktiviert (grau), wenn mindestens ein Laufzeit-Task auf der lokalen Station läuft.

Vor dem Ausführen dieses Befehls müssen alle Laufzeit-Tasks geschlossen werden (**Anwendung beenden** im Kontextmenü des Ordners "WinStudio").



Die im Ordner "Interne Variablen" und im Ordner "Verteilte Variablen" gespeicherten Tags sind von diesem Befehl nicht betroffen.



Wenn die Anwendungsvariablen bei jedem Ausführen des Projekts (**Anwendung starten** im Kontextmenü des Ordners "WinStudio") zurückgesetzt werden sollen, kann beim Starten des Projekts die Option **Variablen-Datenbank beim Start zurücksetzen** auf der Registerkarte Persönliche Einstellungen des Dialogs Projekteinstellungen auf Seite 83 ausgewählt werden.

7.8.5 Verwenden vorhandener Tags

Klicken Sie auf **Objektsuche**  zum Öffnen des Dialogs "Objektsuche", in dem alle aktuell für das Projekt konfigurierten **Tags** und **Funktionen** aufgeführt sind.

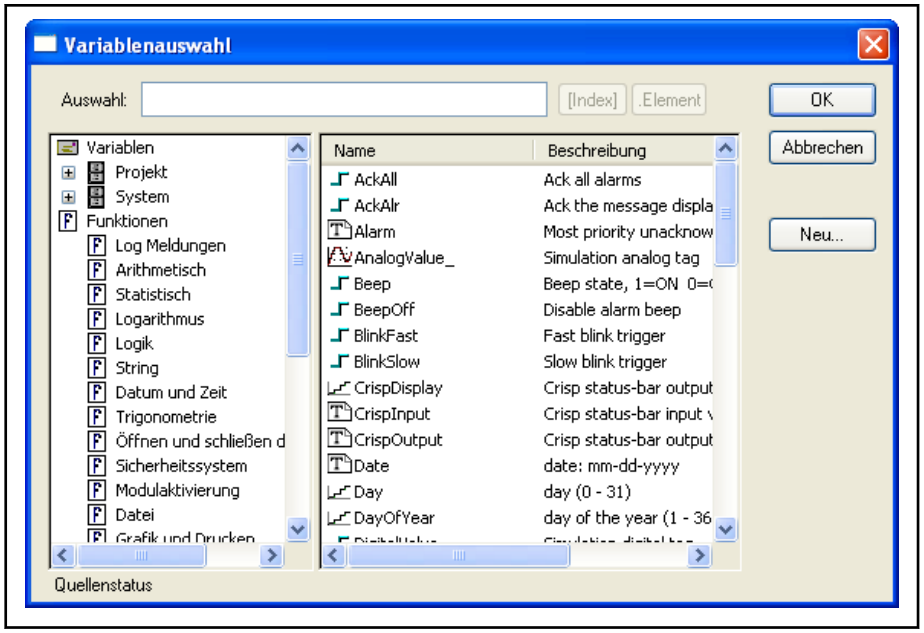


Abb.7-22: Dialog Objektsuche

Zum Auswählen von bestehenden Tags/Funktionen führen Sie einen Doppelklick auf den Tag-/Funktions-Namen aus und klicken dann auf **OK**, um die Box zu schließen. Der ausgewählte Name wird im Textfeld **Tag-Name** angezeigt.

- Zum Auswählen eines bestimmten Arrayindex klicken Sie nach Festlegen des Namens für den Array-Tag auf Seite 125 auf die Schaltfläche **Index**.
- Zum Auswählen eines bestimmten Elements einer Klassenvariable klicken Sie nach Festlegen des Namens für den Klasse-Tag auf Seite 118 auf die Schaltfläche **Element**.
- Zum Erstellen eines neuen Tags klicken Sie auf die Schaltfläche **Neu**.
- Wenn der Dialog "Neuer Tag" angezeigt wird, geben Sie die folgenden Daten ein und klicken zum Schließen des Felds auf **OK**.
  - **Name**
  - **Arraygröße**
  - **Typ** (Boolean, Integer, Real, String, Class:Control, Class:msgonline oder Class:Alr)
  - **Beschreibung**
  - **Gültigkeitsbereich** (lokal oder Server)

7.8.6 Ändern von Tag-Eigenschaften

Der Dialog "Tag-Eigenschaften" ist erreichbar unter "WinStudio Querverweise".

Klicken Sie zum Konfigurieren von Parametern für jeden Tag auf **Tag-Eigenschaften**.

(Weitere Informationen zum Festlegen von Tag-Eigenschaften unter Erläuterung von Tag-Eigenschaften und -Parametern auf Seite 136.)

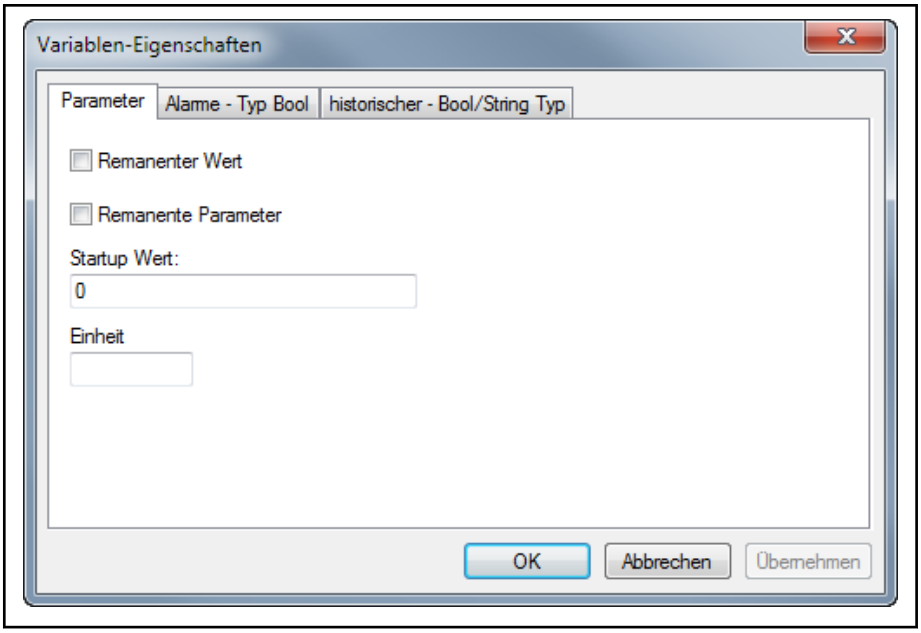


Abb.7-23: Beispiel für Dialog Tag-Eigenschaften

Der für Tags vom Typ Boolean, Integer, Real und String angezeigte Dialog "Tag-Eigenschaften" unterscheidet sich inhaltlich.

7.9 Importieren einer externen Datenbank

7.9.1 Import-Assistent

Der Import-Assistent ist ein leistungsfähiges Hilfsmittel zur Zeitverkürzung bei der technischen Projektentwicklung.

Mit dem Import-Assistenten können Tags aus verschiedenen Datenquellen direkt in die Tag-Datenbank des Projekts importiert werden. Je nach Datenquelle können nicht nur Tag-Namen sondern auch die Kommunikationsschnittstelle (die Verbindung zwischen den Tags und den SPS-Adressen) importiert werden.

Durch Auswahl des Eintrags **Variablenimport-Assistent** im Kontextmenü "Werkzeuge" des Ordners "WinStudio" wird ein Dialog "Assistent Datenbankimport" angezeigt, der Schritt für Schritt durch das Importieren von Tags führt.

Es sind drei Schritte zum Importieren von Tags aus nachstehenden Datenquell-Typen notwendig:

- WinStudio-Projektdatenbank
- OPC-Server-Datenbank
- CSV-Datenbank
- ODBC-Datenbank

1. Schritt: Wählen Sie den Quelltyp

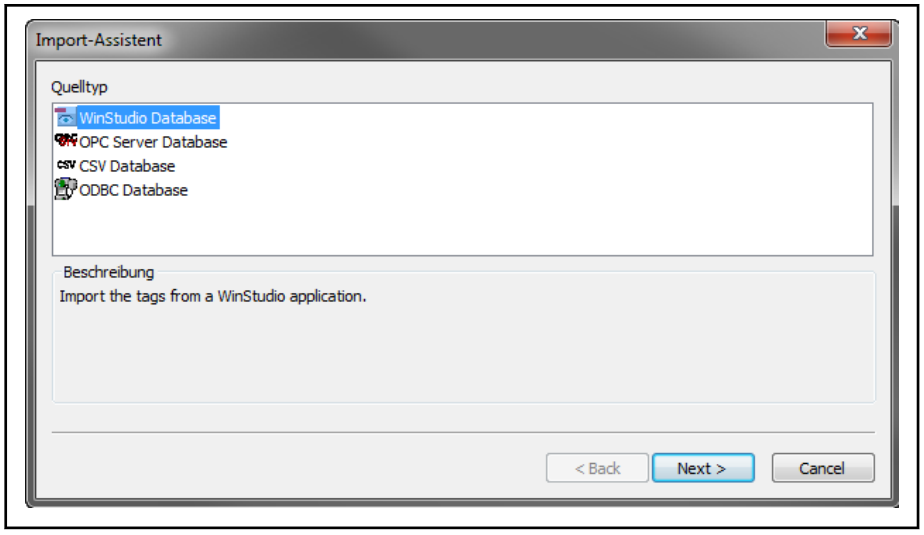


Abb.7-24: Import-Assistent

Klicken Sie auf den Datenquellentyp, von dem die Tags importiert werden. Klicken Sie auf **Weiter**.

Fahren Sie mit dem entsprechenden Abschnitt der Anweisungen fort, die Sie zum Fertigstellen des Datenbankimports benötigen: "Datenquellentypen" auf Seite 157

2. Schritt: Konfigurieren der Quelltyp-Einstellungen

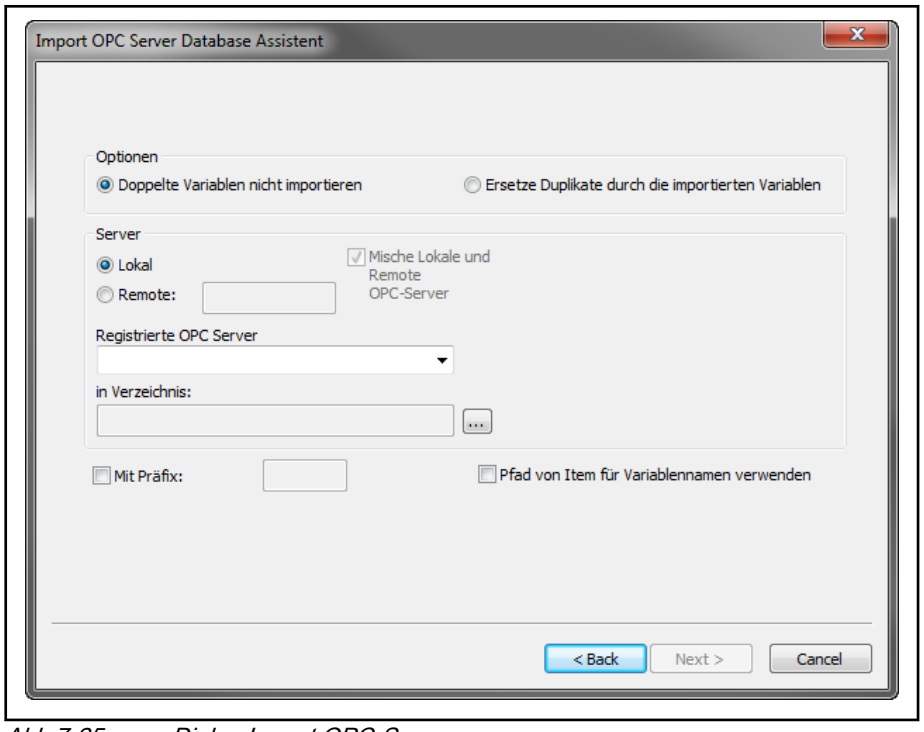


Abb.7-25: Dialog Import OPC-Server

Die meisten Einstellungen im zweiten Fenster sind abhängig von dem im ersten Schritt ausgewählten Quelltyp. Der obige Screenshot ist ein Beispiel für einen Datenquellentyp (OPC Server-Datenbank). Die für alle Quelltypen gültigen Einstellungen sind im Folgenden beschrieben:

- **Optionen**: Wählen Sie **Doppelte Tags nicht importieren**, wenn Sie nicht wollen, dass bestehende Tags in der Tag-Datenbank des aktuellen Pro-

jekts durch importierte Tags mit dem gleichen Namen überschrieben werden.

Wählen Sie **Duplikate durch importierte Tags ersetzen**, um Tags in der Tag-Datenbank mit importierten Tags mit gleichem Namen zu überschreiben.

- **Präfix verwenden**: Aktivieren, um einen Präfix (bis zu 4 Zeichen) festzulegen, das an den Namen der importierten Tags angehängt wird.

  Die Verwendung eines Präfix ist sinnvoll zur Unterscheidung der importierten Tags von manuell erstellten Tags.

☞ Die anderen Einstellungen unterscheiden sich je nach der im ersten Schritt gewählten Datenquelle und werden in den jeweiligen Abschnitten für jeden "Datenquellentypen" auf Seite 157 beschrieben.

Nach dem Konfigurieren der Einstellungen in diesem Dialog klicken Sie auf **Weiter**.

3. Schritt: Filtern der Tags

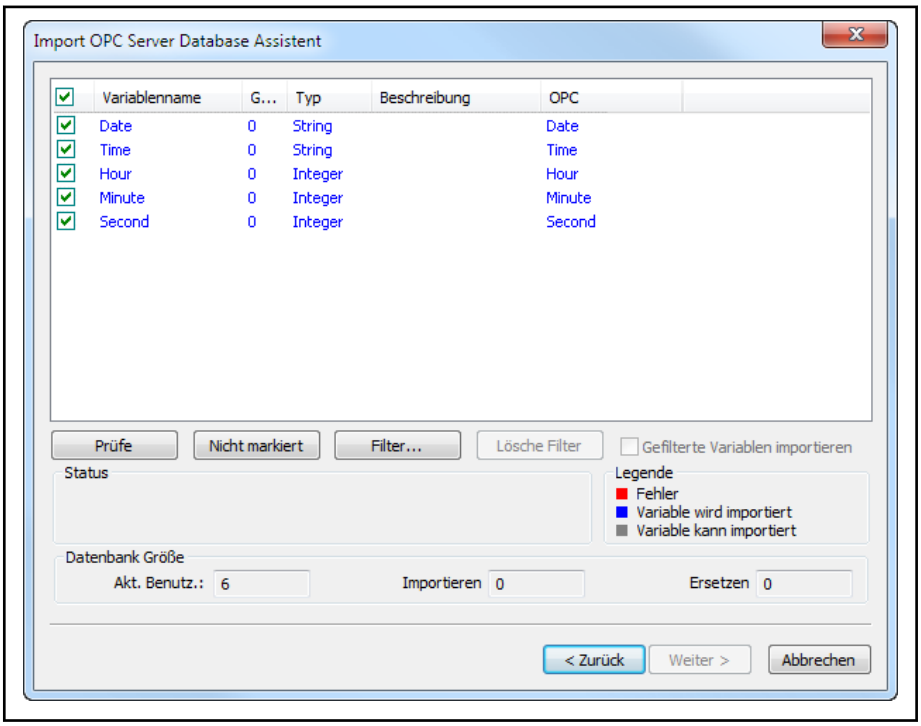


Abb.7-26: Dialog Import OPC-Server

Die obige Abbildung zeigt ein Beispiel für einen Datenquellentyp (OPC Server-Datenbank). Die Felder und Einstellungen, die für alle Datenquellentypen gleich sind, beinhalten die folgenden:

- **Raster**: Zeigt die Liste der in der Datenquelle gefundenen Tags an.
  - **Kontrollkästchen**: Aktivieren zum Importieren der Datenquelle in die Tag-Datenbank des aktuellen Projekts.
  - **Variablenname**: Name des Tags
  - **Größe**: Bereichsgröße des Tags
  - **Typ**: Datentyp des Tags (Boolean, Integer, Real, String oder Class:<ClassName>)
  - **Beschreibung**: Beschreibung des Tags

- **Prüfen**: Klicken zum Auswählen/Importieren aller Tags im Raster
- **Auswahl aufheben**: Klicken um die Auswahl aller Tags im Raster aufzuheben
- **Filter**: Anklicken zum Filtern aller Tags. Der Dialog Filter wird angezeigt, mit dem eine Maske für jede Spalte im Raster festgelegt werden kann. Platzhalter (* und ?) können zum Filtern von Daten verwendet werden.
- **Filter aufheben**: Klicken, um den Filter zurückzusetzen.
- **Nur gefilterte Tags importieren**: Diese Option auswählen, um nur die im Raster sichtbaren (gefilterten) Tags zu importieren.
- **Status**: Zeigt eine Meldung an, die den Status des aktuell im Raster ausgewählten Tags beschreibt. Diese Information ist besonders nützlich zur Anzeige, warum ein Tag nicht importiert werden kann.
- **Legende**: Erklärt die Bedeutung der Farben für den Tag-Status:
  - (Rot) **Fehler:** Tag kann nicht importiert werden, weil er von "WinStudio" nicht unterstützt wird. Siehe Textfeld für genauere Fehlerbeschreibung.
  - (Blau) **Variable wird importiert:** Tag wird nach Klicken auf Fertigstellen importiert.
  - (Grau) **Variable kann importiert werden:** Tag kann importiert werden, wurde aber nicht ausgewählt.
- Feld **Datenbankgröße**: Zeigt die zusammenfassenden Informationen zum aktuellen Import-Assistenten an:
  - **Aktuell**: Gibt die Anzahl der aktuell vorhandenen Tags im Datenblatt "Anwendungsvariablen" an.
  - **Importieren**: Gibt die Anzahl der für den Import ausgewählten Tags an
  - **Ersetzen**: Gibt die Anzahl der in der Projekt-Tag-Datenbank des aktuellen Projekts konfigurierten Tags an, die durch einen importierten Tag mit gleichem Namen ersetzt werden.

Nach dem Auswählen der Tags für den Import Klicken Sie auf Fertigstellen oder auf Abbrechen, um den Vorgang abzubrechen.

Die anderen Einstellungen unterscheiden sich je nach der im ersten Schritt gewählten Datenquelle und werden in den jeweiligen Abschnitten für jeden Datenquellentyp nachstehend beschrieben.

Datenquellentypen

Fahren Sie mit dem entsprechenden Abschnitt der Anweisungen fort, die Sie zum Fertigstellen des Datenbankimports benötigen.

- Importieren aus anderen Projektdatenbanken auf Seite 157
- Importieren aus OPC-Server-Datenbanken auf Seite 159
- Importieren aus CSV-Datenbanken auf Seite 161
- Importieren aus ODBC-Datenbanken auf Seite 162

7.9.2 Importieren aus...

Importieren aus anderen Projektdatenbanken

Mit dem Assistenten können Sie die Objekte (Tags und Arbeitsblätter) von anderen "WinStudio"-Projekten importieren.

Werden von einem Remote-Rechner nur die Tags importiert (und nicht das ganze Projekt), so kann das TCP/IP-Client-Arbeitsblatt automatisch erstellt werden, um die Tags zwischen beiden Stationen (lokal und Remote) zu ver-

binden und den Wert dieser Tags während der Laufzeit von beiden Stationen verwenden zu lassen.

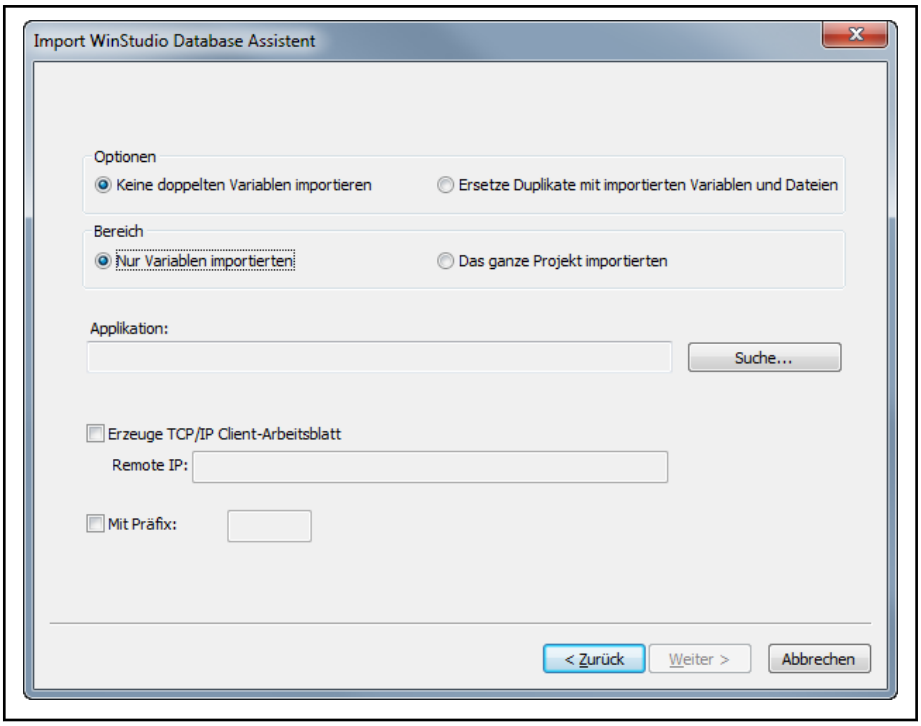


Abb.7-27: Dialog "Projektdatenbank-Import-Assistent "

- **Nur Tags importieren**: wenn diese Option gewählt ist, werden die Tags des anderen Projekts in das aktuelle Projekt importiert. Die anderen Elemente des Projekts (Arbeitsblätter) werden nicht importiert.
- **Gesamtes Projekt importieren**: wenn diese Option gewählt ist, werden die folgenden Elemente des anderen Projekts in das aktuelle Projekt importiert.
  - Tag-Datenbank
  - Globale Prozeduren
  - Bilder
  - Bildergruppen
  - Webseiten
  - Alarme
  - Trends
  - Rezepturen
  - Berichte
  - ODBC
  - Berechnung
  - Skripte
  - Scheduler
  - Treiber
  - OPC
  - TCP/IP
  - DDE

Diese Option ist hilfreich für das Zusammenführen von Projekten und den Import von Projektvorlagen.

☞ Wenn Sie die Option **Gesamtes Projekt importieren** auswählen, werden die folgenden Arbeitsblätter immer importiert, ungeachtet der bestehenden Arbeitsblätter mit der gleichen Nummer im aktuellen Projekt: ODBC, Berechnung, Skript, Scheduler, Treiber, OPC, TCP/IP und DDE.

Wenn im aktuellen Projekt Arbeitsblätter mit der gleichen Nummer vorhanden sind, werden aus dem anderen Projekt importierte Arbeitsblätter als zusätzliche Blätter in das aktuelle Projekt importiert und die Nummer jedes Arbeitsblatts wird automatisch erhöht, um das Ersetzen von Dateien im aktuellen Projekt zu verhindern.

- **Keine Duplikate importieren**: Wenn diese Option ausgewählt ist, werden die folgenden Elemente nicht importiert, falls bereits ein gleichwertiges Element im aktuellen Projekt existiert:
  - Tag-Datenbank (Tags mit dem gleichen Namen werden nicht importiert)
  - Globale Prozeduren (globale Prozeduren werden nicht importiert)
  - Bilder (Bilder mit dem gleichen Namen werden nicht importiert)
  - Bildergruppen (Bildergruppen mit dem gleichen Namen werden nicht importiert)
  - Webseiten (Webseiten mit dem gleichen Namen werden nicht importiert)
  - Alarme (Alarme, die Tags mit dem gleichen Namen zugewiesen sind, werden nicht importiert)
  - Trend (Trendprotokolle, die Tags mit dem gleichen Namen zugewiesen sind, werden nicht importiert)
  - Rezepturen (Rezepturen mit dem gleichen Namen werden nicht importiert)
  - Berichte (Berichte mit dem gleichen Namen werden nicht importiert)
  - Skript (das Startup-Skript wird nicht importiert)
- **Anwendung**: Klicken Sie auf **Durchsuchen** und wählen Sie die Datei **projekt_name.APP** mit den zu importierenden Tags aus.
- **TCP/IP-Client-Arbeitsblatt erzeugen**: Wenn Ihr Projekt Dateien gemeinsam mit einem anderen "WinStudio"-Projekt, das auf einem Remote-Server läuft, verwenden soll, wählen Sie diese Option und geben die IP-Adresse des Servers ein. "WinStudio" konfiguriert das TCP/IP-Client-Arbeitsblatt automatisch für den Datenaustausch mit dem Remote-Projekt.
- **Präfix verwenden**: Wählen Sie diese Option zum Festlegen eines Präfix (bis zu 4 Zeichen), das den Namen der importierten Tags vorangestellt wird. Es ist sinnvoll, die importierten Tags von manuell erstellten Tags zu unterscheiden.

Importieren aus OPC-Server-Datenbanken

Mit diesem Assistenten können Tags von einem lokalen oder von einem Remote-OPC-Server importiert werden.

Beim Importieren von Tags vom OPC-Server wird das OPC-Client-Arbeitsblatt automatisch so erstellt, dass es die Tags verbindet. Damit entfällt das

Konfigurieren der Kommunikationsschnittstelle zwischen dem OPC-Client aus "WinStudio" und dem externen OPC-Server.

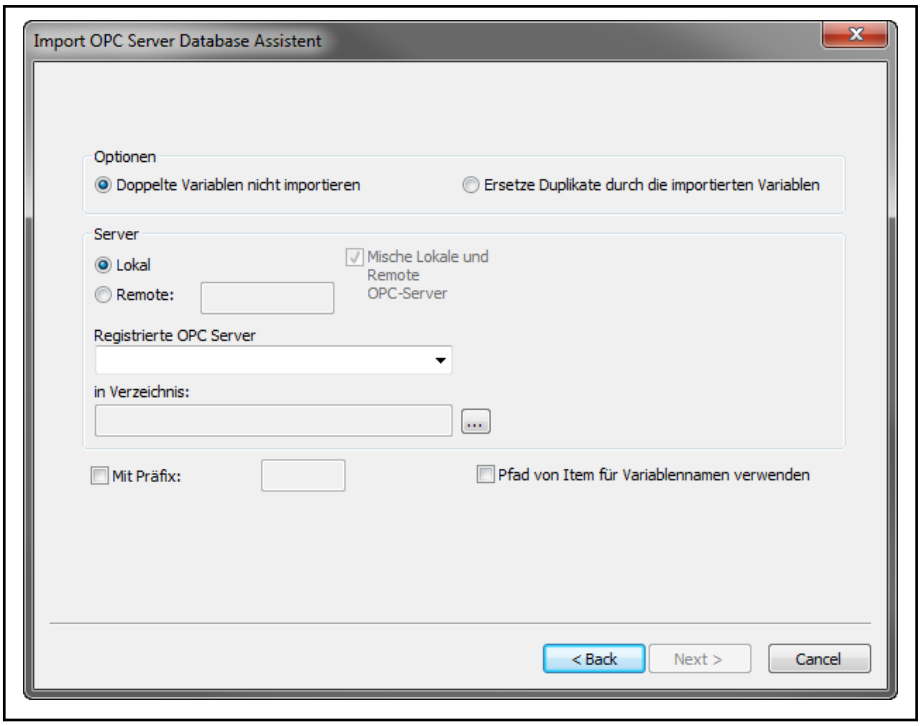


Abb.7-28: Import-Assistent OPC-Server-Datenbank

- **Lokal/Remote**: Bietet die folgenden Optionen:
  - **Lokal:** Wählen Sie diese Option zum Importieren von Tags von einem auf einem lokalen Rechner installierten OPC-Server
  - **Remote:** Wählen Sie diese Option zum Importieren von Tags von einem auf einem Remote-Rechner installierten OPC-Server. Geben Sie die IP-Adresse (oder den Host-Namen) des Remote-Rechners, auf dem "WinStudio" läuft, in das Feld **Remote** ein
- **Zusammenführen von Lokal- und Remote-OPC-Servern**: Wenn Sie einen Remote-Server gewählt haben, wählen Sie diese Option aus, um die Liste der auf dem lokalen und auf dem Remote-Rechner installierten OPC-Server anzuzeigen. Wählen Sie dieses Kontrollkästchen ab, um nur die Liste der auf dem Remote-Rechner installierten OPC-Server anzuzeigen
- **Bezeichner**: Zeigt die Liste der verfügbaren OPC-Server an
- **Verzweigung**: Klicken Sie auf die Schaltfläche Durchsuchen (**...**) , um die Verzweigung des OPC-Servers auszuwählen, aus der die Tags (Elemente) importiert werden sollen. Lassen Sie das Feld leer, wenn Sie Tags aus allen im OPC-Server konfigurierten Verzweigungen importieren wollen
- **Elementpfad für Tag-Name verwenden**: Wählen Sie diese Option, um den Pfadnamen beim Importieren von Tags vom OPC-Server mit dem Elementnamen zu verknüpfen. Wählen Sie diese Option ab, um nur die im OPC-Server konfigurierten Elementnamen zu verwenden

In der in Schritt 3 angezeigten Tabelle des **Import-Assistent** (im Kontextmenü des Ordners "WinStudio") gibt es für diesem Datenquellentyp ein Zusatzfeld mit der Bezeichnung **OPC**, das den Namen des Elements im OPC-Server anzeigt.

Siehe Schritt 1, 2 und 3 des **Import-Assistenten** für Einstellungen und Felder, die für alle Quelltypen gleich sind.

Importieren aus CSV-Datenbanken

Mit diesem Assistenten können Tags aus einer Textdatei in das CSV-Format (Comma Separated Values) oder ein ähnliches Format importiert werden.

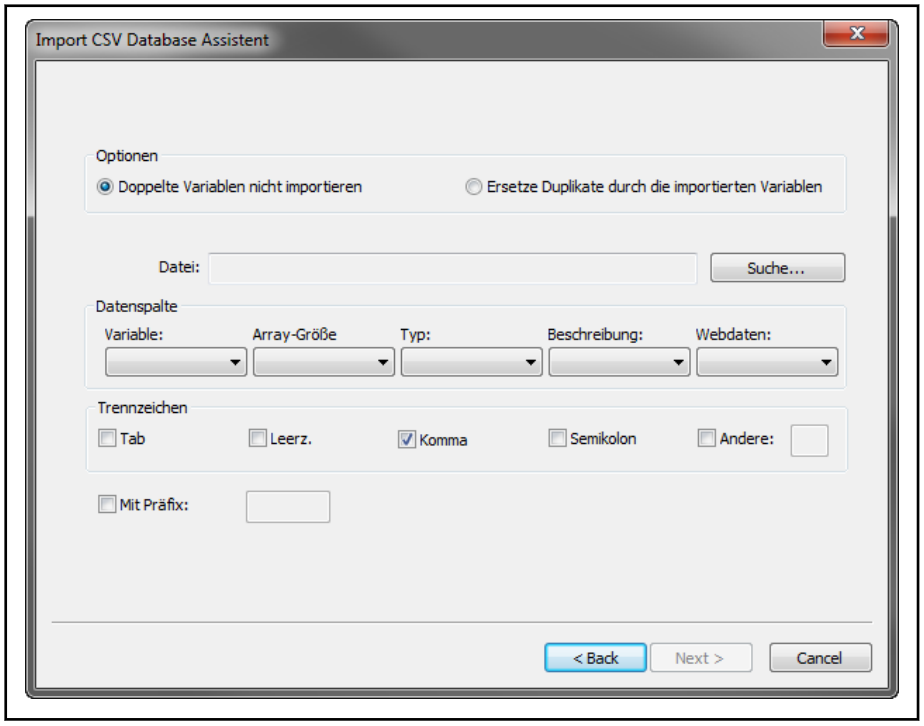


Abb.7-29: Import-Assistent CSV-Datenbank

- **Dateiname**: Klicken zum Öffnen der Textdatei, aus der die Tags importiert werden sollen, auf die Schaltfläche "Durchsuchen".
- **Datenspalte**: Wählen Sie für jede Tag-Eigenschaft eine Nummer, die der Spaltennummer in der Importdatei entspricht.

  Wenn zum Beispiel Tag, Feldgröße und Typ in den zweiten, dritten bzw. ersten Spalten in der Importdatei aufgeführt sind, wählen Sie 2 für Tag, 3 für Feldgröße und 1 für Typ. Die Tag-Eigenschaft (Tag-Name) ist ein Pflichtfeld, die anderen Eigenschaften sind optional.

  Für nicht in der Textdatei enthaltene Eigenschaften wählen Sie die Option **Nicht verwendet**. "WinStudio" setzt Standardwerte ein oder lässt das Feld frei, gemäße der folgenden Liste:

  – **Feldgröße**: 0
  – **Typ**: Integer
  – **Beschreibung**: <Leer>
  – **Web-Daten**: Lokal
- **Feldbegrenzer**: Wählen Sie den/die in der Textdatei verwendeten Feldbegrenzer aus, um die Spalten voneinander zu trennen. Für eine CSV-Datei ist das Komma der Feldbegrenzer (der Standard). Es kann mehr als ein Feldbegrenzer gleichzeitig ausgewählt werden, und über die Option "Andere" kann ein benutzerdefinierter Feldbegrenzer eingegeben werden.



Siehe Schritt 1, 2 und 3 des **Import-Assistenten** für Einstellungen und Felder, die für alle Quelltypen gleich sind.

Importieren aus ODBC-Datenbanken

Mit diesem Assistenten können Tags aus einer externen relationalen SQL-Datenbank wie Microsoft Access, SQL Server, Oracle, My SQL, Sybase oder anderen über deren ODBC-Schnittstelle importiert werden.

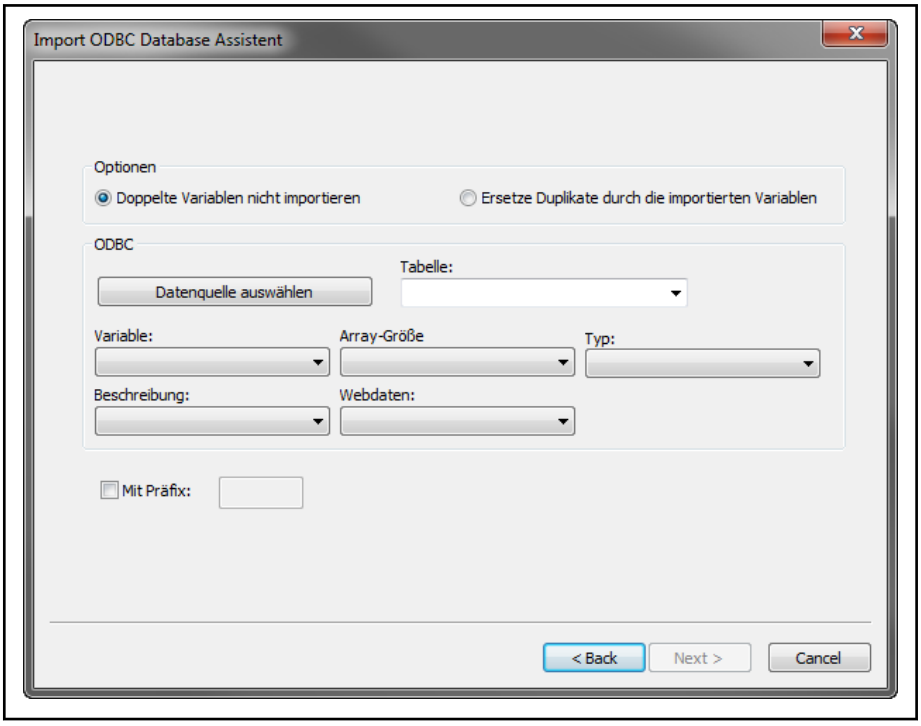


Abb.7-30: Import-Assistent ODBC-Datenbank

- **Datenquelle auswählen**: Klicken zum Auswählen des Namens der ODBC-Datenquelle (DSN), die mit der Datenbank verbunden ist, aus der die Tags importiert werden sollen. Der DSN muss vorher mit dem Fenster Datenquellen (ODBC) erstellt worden sein (**Systemsteuerung ▸ Verwaltung ▸ Datenquellen [ODBC]**).

  Nach dem Auswählen eines DSN werden die anderen Felder in diesem Fenster automatisch mit Informationen aus der gewählten Datenbank gefüllt.
- **Tabelle**: Wählen Sie die Tabelle, die die Tags in der Importdatenbank enthält.
- **Tag**: Wählen Sie den Namen der Spalte, die die Tags in der Importdatenbank enthält.
- **Feldgröße**: Wählen Sie den Namen der Spalte, die die Feldgröße (Arraygröße) für die Tags in der Importdatenbank enthält.
- **Typ**: Wählen Sie den Namen der Spalte, die den Tag-Typ in der Importdatenbank enthält.
- **Beschreibung**: Wählen Sie den Namen der Spalte, die die Tag-Beschreibung in der Importdatenbank enthält.
- **Web-Daten**: Wählen Sie den Namen der Spalte, die die Webdaten für die Tags in der Importdatenbank enthält.

☞ Siehe Schritt 1, 2 und 3 des **Import-Assistenten** für Einstellungen und Felder, die für alle Quelltypen gleich sind.

7.10 Integrieren von Tags

7.10.1 Integrieren von Tags, Allgemeines

Mit der automatischen Tag-Integration können Variablen, die in einer Steuerung konfiguriert sind, im WinStudio-Variablenbrowser (Object-Finder) direkt ausgewählt werden, ohne dass Sie zunächst einen entsprechenden Tag erstellen und ein Treiberblatt konfigurieren müssen.

Sie unterscheidet sich von den anderen Optionen zum Einbinden von Fremdtags in Ihr Projekt durch Funktionen wie das Importieren von Tag-Definitionen oder das Konfigurieren eines Arbeitsblatts zum Verknüpfen von Projekt-Tags mit externen Elementen.

Bei der Tag-Integration sind die Fremdtags dynamisch, d.h. sie werden während der Laufzeit kontinuierlich und bilateral aktualisiert, solange die Fremdsteuerungsanwendung oder das Fremdsteuerungsgerät ebenfalls laufen, und sie können überall dort im Projekt verwendet werden, wo normalerweise Projekt-Tags verwendet werden.

Es gibt zwei Arten von Tag-Integration:

1. **Integriert**: Alle Tags aus der Quelle können in der Objektsuche durchsucht werden, einzelne Tags werden aber weder der Anwendungsvariablen-Datenbank hinzugefügt noch für die Höchstzahl der Tags in dem Projekt gezählt, bis sie tatsächlich in Ihrem Projekt verwendet werden.
2. **Gemeinsam**: Alle Tags aus der Quelle werden automatisch der Anwendungsvariablen-Datenbank hinzugefügt und zählen für die Höchstzahl der Tags in dem Projekt.

☞ Diese Option ist veraltet und aus Kompatibilitätsgründen mit älteren Projekten vorhanden. Wenn Sie diesen Typ in einem neuen Projekt verwenden wollen, wenden Sie sich bitte an den Kundendienst.

Tag-Integration ist Bestandteil der Projekteinstellungen.

- Integration von Tags aus IndraLogic auf Seite 163

  WinStudio kann sich mit BoschRexroth-Steuerungen und IndraLogic-Geräten verbinden und Tags gemeinsam nutzen.
- Verwenden der Objektsuche zum Auswählen integrierter Tags

  Nach dem Hinzufügen einer Tag-Integrationsquelle zu Ihrem Projekt können die Tags aus dieser Quelle wie alle anderen Tags in der Objektsuche eingesehen und ausgewählt werden
- Umbenennen von integrierten Tags in Ihrem Projekt

  Wenn WinStudio Tags von Steuerungen integriert, können die Variablen-Namen nicht identisch übernommen werden. Es werden einige Änderungen vorgenommen, um die Verwaltung von Tags zu verbessern und die lokale Namenssyntax einzuhalten.

7.10.2 Tags gemeinsam mit IndraLogic-Geräten verwenden

WinStudio kann sich mit BoschRexroth-Steuerungen und IndraLogic-Geräten verbinden und Tags gemeinsam nutzen.

Arbeiten mit Tags

Tags automatisch über IndraWorks gemeinsam verwenden

Wenn Sie WinStudio zusammen mit Rexroth IndraWorks auf der gleichen Station betreiben, wird die gesamte Gerätekommunikation und die gemeinsame Nutzung von Tags von IndraWorks verwaltet. Es muss nichts in der Entwicklungsanwendung konfiguriert werden; WinStudio verbindet sich automatisch mit allen registrierten Geräten und stellt die Tags zur Verwendung in Ihrem Projekt zur Verfügung. Die Tags sind dynamisch und werden ständig aktualisiert und können mit Hilfe des Ordners Verteilte Variablen auf Seite 120 oder der Objektsuche auf Seite 152 durchsucht werden.

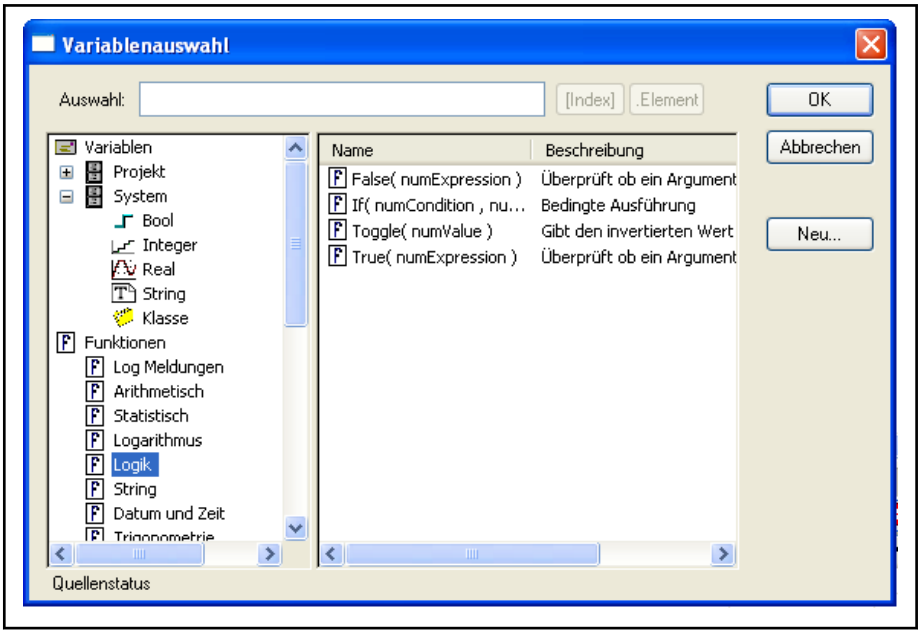


Abb.7-31: Geräte in der Objektsuche durchsuchen

Die Tags sind zum leichteren Durchsuchen in der "Objektsuche" nach Gerät gruppiert und sind außerdem nach Datentyp eingeteilt, um die Kompatibilität mit Objekten und Funktionen aus WinStudio sicherzustellen. Sie können diese gemeinsamen Tags in Ihrem Projekt genau wie andere Projekt-Tags verwenden.

Weitere Informationen über die Verwendung von Rexroth WinStudio und Konfigurieren von IndraLogic-Geräten finden Sie im "IndraWorks Bedien- und Programmierhandbuch".

Konfigurieren von WinStudio zur Kommunikation mit Geräten

Wenn Sie das WinStudio-Projekt und die verwendeten Steuerungen gemeinsam in einem IndraWorks-Projekt verwalten, sind keine weiteren Einstellungen notwendig.

Falls Sie Variablen von Steuerungen verwenden möchten, die nicht in Ihrem-Projekt enthalten sind, können Sie diese folgendermaßen hinzufügen:

1. Klicken Sie im Eigenschaftendialog des Ordners WinStudio auf das Register **Kommunikation**.

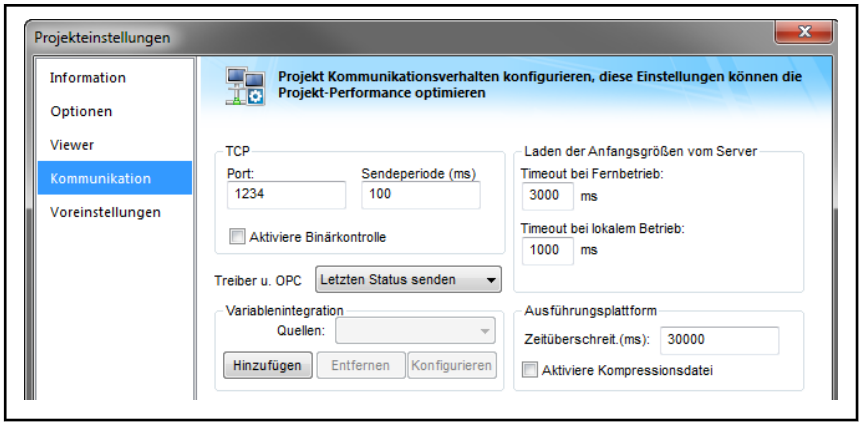


Abb.7-32: Registerkarte Kommunikation des Dialogs Projekteinstellungen

2. Klicken Sie im Bereich "Variablen-Integration" auf **Hinzufügen**. Der Dialog "Variablen-Integrationsquelle" wird angezeigt:

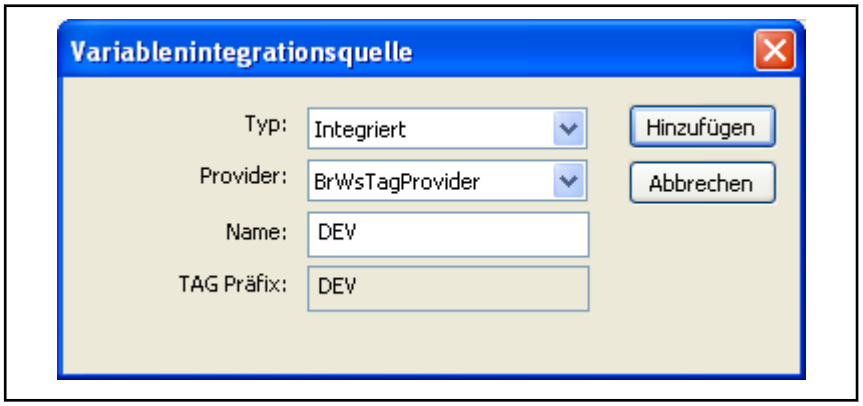


Abb.7-33: Auswahl der Quelle für die Tag-Integration

3. Wählen Sie **Integriert** aus der Liste **Typ**, falls dies noch nicht gewählt ist.
4. Wählen Sie aus der Liste **Provider** die Kommunikationsart aus, die von der Steuerung unterstützt wird.
5. Geben Sie im Feld **Name** einen Namen für das Zielgerät ein. Das ist der Name, der in der "Objektsuche" angezeigt wird und unter der die Tags gruppiert werden. Der Name muss der korrekten Syntax folgen – falls nicht, wird er von WinStudio nicht akzeptiert. Weitere Informationen unter „Erläuterung der Namenssyntax für Tags“ auf Seite 121 Bei der Eingabe des Gerätenamens dupliziert WinStudio diesen automatisch in das Feld **TAG-Präfix**. Dieses Präfix wird an die lokalen Projekt-Tags angefügt, die zur Synchronisation mit den Geräte-Tags erstellt werden. (Siehe „Wie Geräte-Tags in WinStudio umbenannt werden können“ weiter unten.)
6. Klicken Sie auf **Hinzufügen**. Es öffnet sich ein Konfigurationsdialog, in dem Sie spezielle Einstellungen für den gewählten Provider vornehmen können.

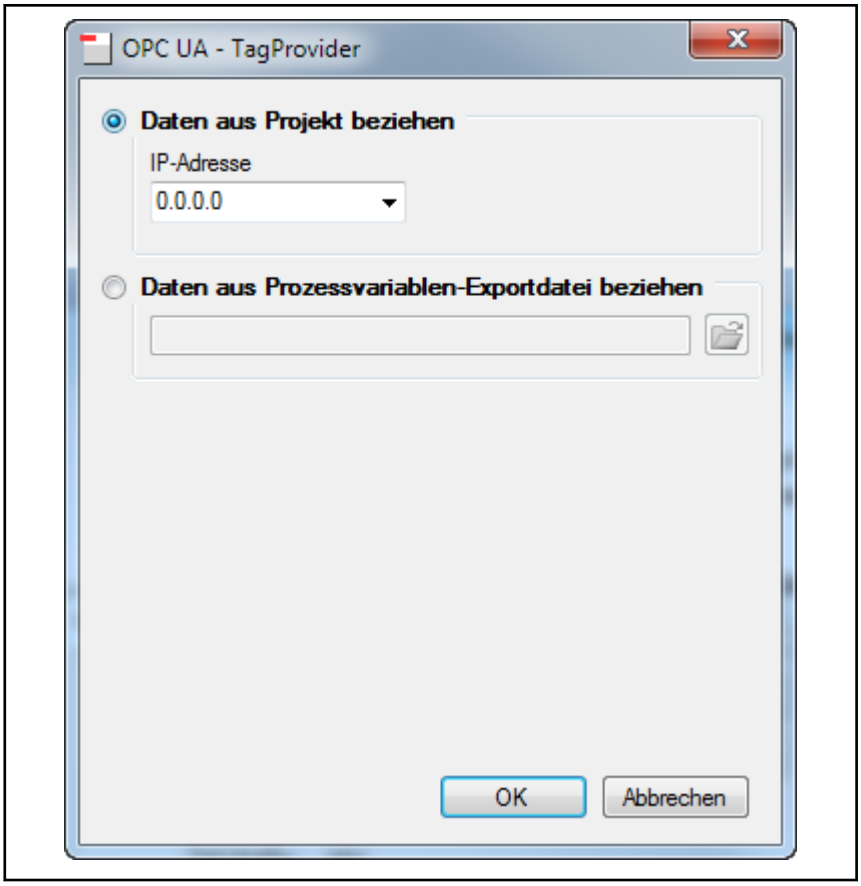


Abb.7-34: Konfiguration eines Variablenproviders für OPC-UA-Kommunikation

7. Klicken Sie auf **OK**.

Wie Geräte-Tags in WinStudio umbenannt werden können

Wenn WinStudio lokale Projekt-Tags entsprechend den Geräte-Tags erstellt, kann es die Geräte-Tag-Namen nicht direkt übernehmen. Es werden einige Änderungen vorgenommen, um die Verwaltung von Tags zu verbessern und die Syntax von WinStudio einzuhalten.

Der vollständige, ursprüngliche Gerätename kann immer wiederhergestellt werden durch Zugriff auf das Tag-Feld "DeviceTag" auf dem entsprechenden Projekt-Tag.

Zum Beispiel **tagName->DeviceTag**.

Einfügen des Tag-Präfix

Da Ihr Projekt sich mit mehreren Geräten mit dem gleichen Steuerungsprogramm und den gleichen Geräte-Tags verbinden könnte, fügt die Entwicklungsanwendung automatisch den beim Hinzufügen des Gerätes festgelegten Gerätenamen/Tag-Präfix ein.

Zum Beispiel für einen Geräte-Tag mit dem Namen...

MeinGerätTAG[1].KlasseMember

...lautet der entsprechende Projekt-Tag in WinStudio..

gerät_name.MeinGerätTAG[1].KlasseMember

So kann zwischen ähnlichen Tags auf verschiedenen Geräten unterschieden werden.

Mehrdimensionale Felder

WinStudio unterstützt keine mehrdimensionale Arrays, daher wird für Geräte-Tags mit mehr als einem Arrayindex jeder Index nach dem ersten durch **_Index_** dargestellt.

Zum Beispiel für einen Geräte-Tag mit dem Namen...

MeinGerätTAG[1][2][3].KlasseMember

...lautet der entsprechende Projekt-Tag in WinStudio..

gerät_name.MeinGerätTAG[1]_2__3_.KlasseMember

Verschachtelte Klassen

WinStudio unterstützt keine verschachtelten Klassen, daher wird für Geräte-Tags mit mehr als einem Klasse-Member jedes Member nach dem ersten durch **_KlasseMember** dargestellt.

Zum Beispiel für einen Geräte-Tag mit dem Namen...

MeinGerätTAG.KlasseMember.KlasseMember2

...lautet der entsprechende Projekt-Tag in WinStudio..

gerät_name.MeinGerätTAG.KlasseMember_KlasseMember2

Aktualisieren von Tag-Informationen von Geräten

In der "Objekt-Suche" werden konfigurierte Geräte mit drei möglichen Symbolen dargestellt:

Symbol	Beschreibung
	Geräte-Tag-Information wurde aus einer Symboldatei extrahiert und dann im Cache-Speicher abgelegt.
	Geräte-Tag-Information wird gerade von einem angeschlossenen Gerät empfangen.
	Verbindung zum Gerät fehlgeschlagen. Tag-Informationen sind evtl. nicht aktuell.

Zum Aktualisieren der Geräte-Tag-Informationen – entweder durch erneutes Einlesen der Symboldatei oder durch einen erneuten Verbindungsversuch zum Gerät – klicken Sie in der "Objektsuche" auf **Geräte aktualisieren**. Alle konfigurierten Geräte werden gleichzeitig aktualisiert.

Änderungen an den Geräte-Tag-Informationen könnten Tag-Verknüpfungen in Ihrem Projekt unterbrechen. Um unterbrochene Bezüge zu finden, überprüfen Sie Ihr Projekt (**Überprüfen** auf der Registerkarte Start der Multifunktionsleiste).

Synchronisieren der Tag-Datenbank mit dem Gerät

Wenn dem Projekt ein Zielgerät hinzugefügt wird, liest die Entwicklungsanwendung die Symboldatei des Gerätes und fügt Ihrer Projekt-Tag-Datenbank entsprechende Tags hinzu. Danach muss die Datenbank synchronisiert werden, um eventuelle Änderungen an den Geräte-Tags aufzunehmen. Zum Synchronisieren der Datenbank:

1. Wenn die Option **Automatisches Synchronisieren** für das Zielgerät ausgewählt wurde, wird die Datenbank automatisch bei jedem Überprüfen des Projekts (**Überprüfen** auf der Registerkarte Start der Multifunktionsleiste) oder beim Durchsuchen der Tags mit der Objektsuche aktualisiert.

   Wenn dies geschieht, erhält WinStudio eine Liste der aktuellen Tags auf dem Zielgerät und vergleicht diese mit Ihrer Tag-Datenbank. Der Dialog "Datenbank-Synchronisation" wird angezeigt mit einer Liste aller Tags, die seit der letzten Synchronisation vom Gerät gelöscht wurden.

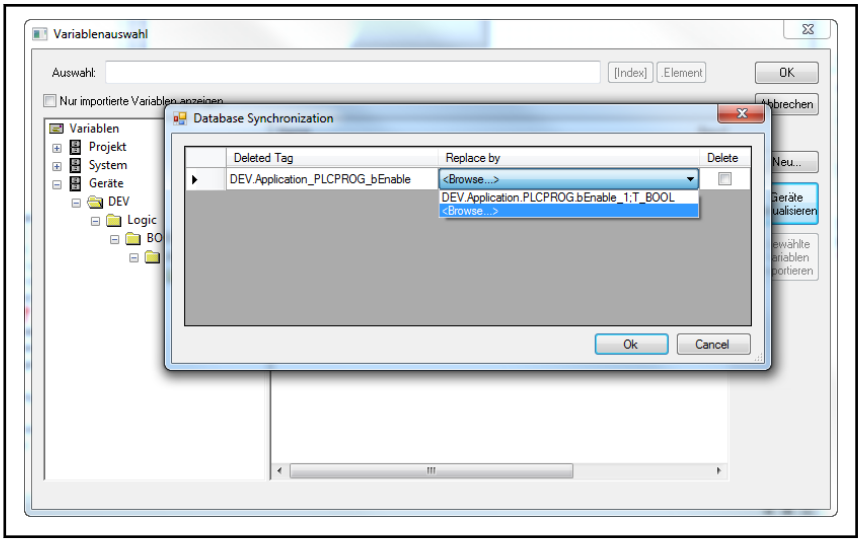


Abb.7-35: Dialog Datenbank-Synchronisation

Für jeden Tag in der Liste kann entweder ein Tag aus dem Projekt gelöscht oder alle Vorkommen des Tags (in Objekten und Arbeitsblättern) durch einen neuen Tag ersetzt werden:

- Zum Ersetzen eines Tags:
  1. Führen Sie einen Klick auf die Spalte **Ersetzen durch** für diesen Tag aus.

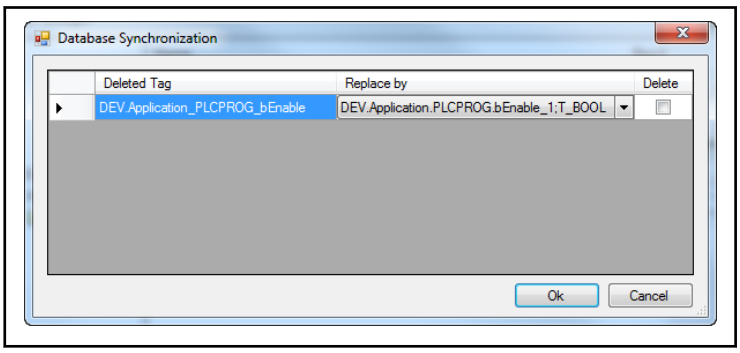


Abb.7-36: WinStudio, hinzugefügte Tags

  2. Wählen Sie aus der Liste eine bestehende Variable aus, oder klicken Sie **Browse** um eine Variable einzufügen.
  3. Schließen Sie den Dialog "Datenbank Synchronisation", indem Sie auf **OK** klicken.
- Zum Löschen eines Tags klicken (aktivieren) Sie das Kontrollkästchen **Löschen** für diesen Tag. Wenn Sie einen Tag aus Ihrem Projekt löschen, während der Tag noch von Objekten und Arbeitsblättern verwendet wird, erhalten Sie beim nächsten Überprüfen oder Ausführen Ihres Projekts Fehlermeldungen. Sie können das Hilfsmittel Querverweisliste auf Seite 96 verwenden, um zu sehen, wo ein Tag in Ihrem Projekt verwendet wird.

2. Klicken Sie auf **OK**, um Ihre Änderungen anzuwenden und den Dialog "Datenbank-Synchronisation" zu schließen.

Bitte bedenken Sie: Wenn Sie einen aufgeführten Tag nicht ersetzen oder löschen, wird dieser Tag beim nächsten Synchronisieren der Datenbank wieder aufgeführt.

8 Bilder und Grafiken

8.1 Bilder und Grafiken, Allgemeines

Die Fähigkeit, durch Interaktion mit Instrumenten (oder Rechnern) den Zustand des Prozesses darzustellen, wird als Mensch-Maschinen-Schnittstelle (Human Machine Interface, HMI) bezeichnet.

Mit "WinStudio" können hochauflösende Prozessbilder zur Überwachung von Prozessen erstellt werden.

Die Grafikwerkzeuge von WinStudio bestehen aus zwei Modulen:

- dem "Bild-Editor" in der Entwicklungsumgebung "IndraWorks Engineering" (zum Erstellen oder Importieren von Grafiken) und
- dem "Viewer" zur Anzeige von WinStudio-Bildern in der Laufzeitumgebung.

Es können "Animationen" zum Erstellen von dynamischen Grafikobjekten oder Symbolen verwendet werden. Durch Animationen kann das Aussehen von Objekten und Symbolen verändert werden, um die Wertänderung eines Tags oder Ausdrucks anzuzeigen. Jedes WinStudio-Bild enthält in der Regel mehrere statische und animierte Objekte.

In WinStudio-Bildern können Sie eine Grafik auswählen, die als Hintergrund angezeigt wird.

Im folgenden WinStudio-Bild können zum Beispiel die statischen Elemente Teil einer Hintergrundgrafik sein, und animierte Objekte können die Änderungen in dargestellten Prozess anzeigen. So entsteht der Eindruck eines dreidimensionalen Bildes.

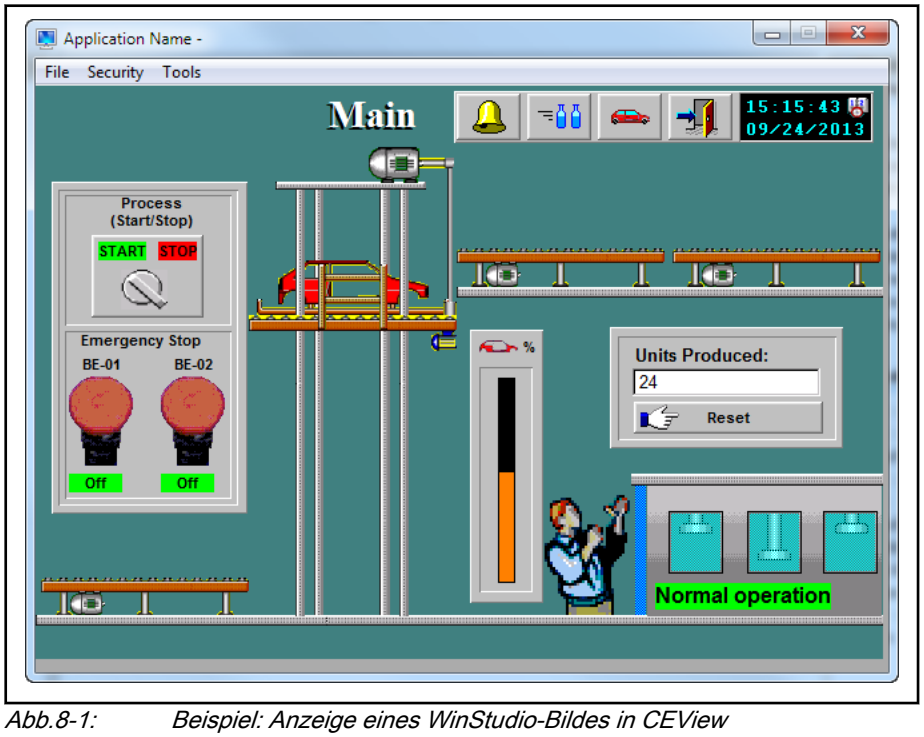


Abb.8-1: Beispiel: Anzeige eines WinStudio-Bildes in CEView

Für die Konfiguration der WinStudio-Anwendung benötigen Sie ein Windows-kompatibles Zeigegerät, zum Beispiel eine Maus oder ein Touch-Pad.

Die Laufzeit des Projekts kann im Viewer auch ohne Zeigegerät ausgeführt werden, wenn für alle Befehle Tastatur-oder Tastenfeldtasten konfiguriert werden.

8.2 Ordner des WinStudios

8.2.1 Ordner WinStudio

Ordner WinStudio, Allgemeines

Der Ordner **WinStudio** befindet sich je nach Applikationstyp, den Sie beim Hinzufügen des IndraControlV-Gerätes gewählt haben, direkt unterhalb des Ordners "VPx-/VEx/VRx" oder im Unterordner "ScreenApplication".

Der Ordner **WinStudio** enthält alle Bilder, Bildgruppen und Symbole in Ihrem Projekt.

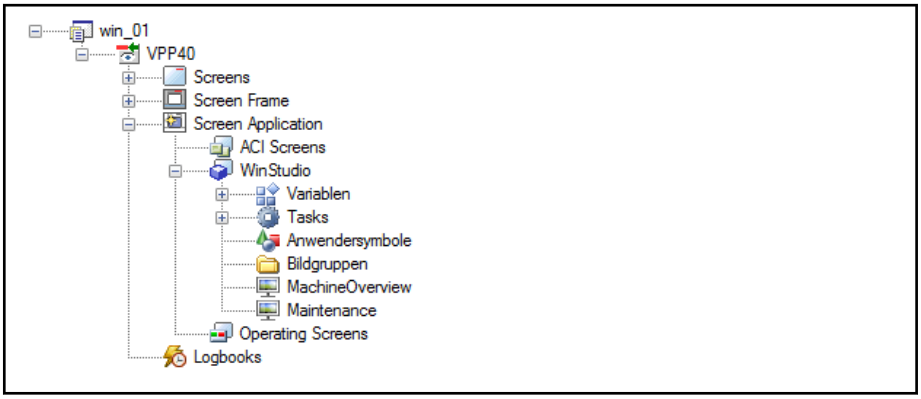


Abb.8-2: Der Ordner "WinStudio"

Die einzelnen Ordner werden auf den folgenden Seiten beschrieben:

- **WinStudio** enthält alle Bilder auf Seite 170, die für das aktuelle Projekt erstellt wurden.

  Diese Bilder können in einer flachen Liste oder in benutzerdefinierte Unterordner strukturiert werden.
- **Variablen** enthält alle Funktionen und Editoren zum Bearbeiten von Variablentypen sowie zur Konfiguration der Prozessanbindung von Variablen mit einem externen Gerät.
- **Tasks** enthält verschiedene Editoren und Funktionen, die Sie für die Erstellung komplexer Anwendungen benötigen ("Mathematikblätter", "Alarme", etc.)
- **Bildgruppen** enthält alle Bildgruppen auf Seite 175 (Einzelne WinStudio-Bilder, die zu handhabbaren Gruppen zusammengefasst sind), die für das aktuelle Projekt angelegt sind.
- **Anwendersymbole** enthält alle benutzerdefinierten Symbole auf Seite 220, wobei es sich um Gruppen von Bildern und/oder Text handeln kann. Sie können für das Projekt kundenspezifische Symbole anlegen und sie in diesem Ordner speichern.
- **Grafikskript** enthält vordefinierte Funktionen, die beim Auftreten bestimmter Bildaktionen ausgeführt werden auf Seite 800, z.B. wenn der Thin Client auf einer Remote-Station gestartet wird.

Dialog Bildeigenschaften

Der Dialog "Bildeigenschaften" wird zum Konfigurieren der Laufzeiteinstellungen verwendet – zum Beispiel Größe, Position, Titelleiste, Sicherheitsstufe und Bildlogik – für ein bestimmtes WinStudio-Bild.

Öffnen des Dialogs

Um den Bildeigenschaften-Dialog anzusehen, führen Sie im Editor des geöffneten WinStudio-Bildes einen Rechtsklick aus und wählen im Kontextmenü den Eintrag "Eigenschaften" aus

Dialog im Detail

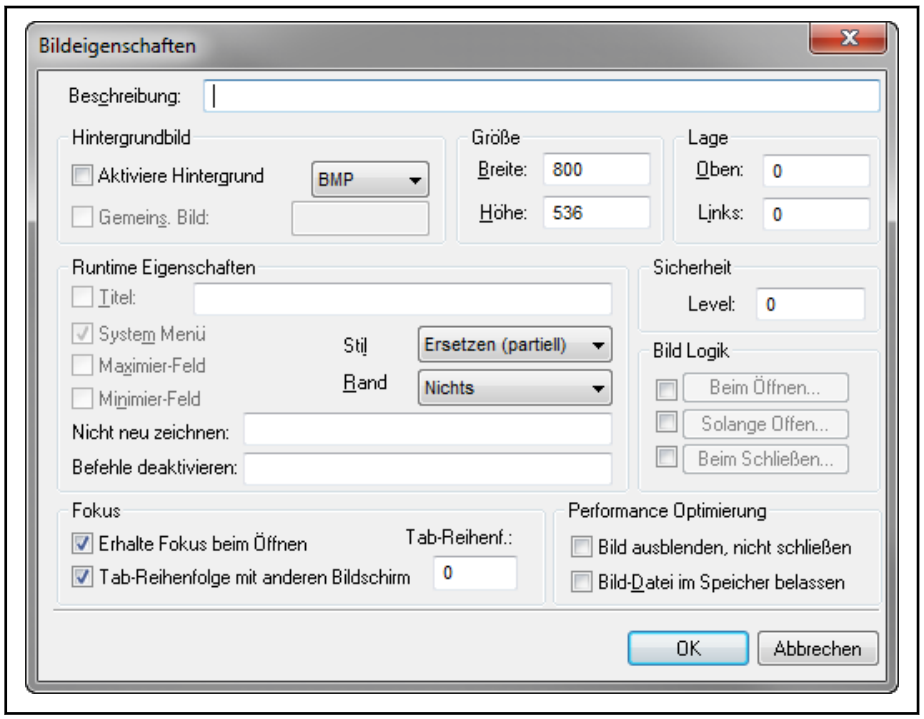


Abb.8-3: Dialog Bildeigenschaften

Elemente des Dialogs "Bildeigenschaften"

Bereich / Elementname		Beschreibung
Beschreibung		Eine kurze Beschreibung des WinStudio-Bildes. Diese wird während der Laufzeit nicht angezeigt.
Hintergrundbild	**Hintergrund aktivieren**	Aktiviert die Hintergrundbildebene und legt den Dateityp der Hintergrundgrafik fest. Wenn diese Option ausgewählt ist, wird automatisch eine neue BMP-Datei mit dem gleichen Namen wie das WinStudio-Bild (z.B. **..\Screens\<Bildname>.BMP**) erstellt. Dieses Bild kann dann ggf. mit einer Bildbearbeitungssoftware bearbeitet werden. Weitere Informationen unter Ändern von Hintergrundfarbe oder -Bild eines Bildes auf Seite 174.
	Gemeinsam benutztes Bild	Verwendet die im Unterordner "Screens" Ihres Projektordners abgelegte Bilddatei. Wenn ein Tag/Ausdruck statt eines festen Namens verwendet werden soll, damit der Wert während der Laufzeit geändert werden kann, so muss der Tag/Ausdruck in geschweifte Klammern gesetzt werden. Zum Beispiel: **{meinTag}** Der Dateiname darf "nicht" die Erweiterung enthalten. Für andere Dateiformate als BMP verwenden Sie die Liste rechts neben der obigen Option **Hintergrund aktivieren** zur Auswahl des korrekten Formats. **Hinweis: In Projekten für das Zielsystem Windows CE werden nur BMP-Dateien unterstützt.**

Bilder und Grafiken

Bereich / Elementname		Beschreibung
Größe	**Breite**	Die Standardbreite der Bildes (in Pixel) bei der ersten Anzeige. Der Benutzer kann die Größe während der Laufzeit ändern, wenn das Bild für Größenänderungen freigegeben ist; siehe **Tabelleneintrag** weiter unten.
	Höhe	Die Standardhöhe des Bildes (in Pixel) bei der ersten Anzeige. Der Benutzer kann die Größe während der Laufzeit ändern, wenn das Bild für Größenänderungen freigegeben ist; siehe **Tabelleneintrag** weiter unten.
Lage	**Nach oben**	Der Standardabstand (in Pixel) zwischen dem oberen Rand des Bildschirms und dem oberen Rand des Bildes beim ersten Anzeigen des Bildes. Der Benutzer kann die Position während der Laufzeit ändern, wenn für das Bild "..." eingestellt sind; siehe **Tabelleneintrag** weiter unten.
	Links	Der Standardabstand (in Pixel) zwischen dem linken Rand des Bildschirms und dem linken Rand des Bildes beim ersten Anzeigen des Bildes. Der Benutzer kann die Position während der Laufzeit ändern, wenn für das Bild "..." eingestellt sind; siehe **Tabelleneintrag** weiter unten.
Sicherheit		Die Mindestsicherheitsstufe, die ein Benutzer für den Zugriff auf das Bild benötigt.
Perfomance Optimierung		Hält das Bild nach dem ersten Aufrufen im Speicher vor, sodass es anschließend schneller aufgerufen werden kann. Alle Benutzeraktionen oder Systemprozesse, die normalerweise das Bild schließen würden, verbergen es nur. **Achtung: Diese Option sollte nur für wichtige Bilder, die schnell geöffnet werden müssen, ausgewählt werden. Wenn zu viele Bilder im Speicher vorgehalten werden, beeinträchtigt dies die gesamte Projektleistung.**

Bereich / Elementname		Beschreibung
Runtime Eigenschaften	**Stil**	Beeinflusst das allgemeine Laufzeitverhalten des Bildes: **Überlagern**: Öffnet das Bild, ohne andere Bilder zu schließen. **Popup**: Öffnet das Bild vor allen anderen Bildern als Dialog, schließt diese aber nicht. **Ersetzen (teilweise)**: Öffnet das Bild und schließt alle anderen Ersetzen- und Popup-Fenster. Dies ist die Standardeinstellung. **Dialog**: Ähnlich wie Popup, aber die anderen Bilder werden außerdem deaktiviert, bis der Dialog geschlossen wird. **Ersetzen (vollständig)**: Ähnlich wie Ersetzen (teilweise), aber überlagerte und Dialoge werden außerdem geschlossen.
	Rahmen	Die Rahmenart des Bildes: **Keine**: Kein Rahmen; das Bild ist ein flaches, unbewegliches Rechteck auf dem Bildschirm. Dies ist die Standardeinstellung. **Schmal**: Ein schmaler Rahmen, der das Bild zu einem beweglichen Fenster macht. Enthält eine Titelleiste. **Größenänderung**: Ein breiter Rahmen, der das Bild zu einem beweglichen Fenster mit veränderbarer Größe macht. Enthält eine Titelleiste.
	Titelleiste	Zeigt die Titelleiste des Fensters mit dem festgelegten Fensternamen an. Wenn ein Tag/Ausdruck als Quelle für diesen Wert festgelegt werden soll, damit der Wert während der Laufzeit durch das Programm geändert werden kann, so muss der Tag/Ausdruck in geschweifte Klammern gesetzt werden. Zum Beispiel: "{meinTag}" **Tipp: Das Festlegen eines Fensternamens ist sinnvoll, auch wenn die Titelleiste nicht angezeigt wird, weil beim Drucken des Bildes der Fenstername in der Kopfzeile der Seite enthalten ist.**
	Systemmenü	Bietet ein Menü mit grundlegenden Windows-Befehlen am linken Ende der Titelleiste.
	Maximieren	Zeigt die Schaltfläche Maximieren am rechten Rand der Titelleiste an.
	Minimieren	Zeigt die Schaltfläche Minimieren am rechten Rand der Titelleiste an.
	Nicht neu zeichnen	Solange dieser Tag/Ausdruck den Wert WAHR besitzt, wird das Bild nicht aktualisiert.
	Befehle deaktivieren	Solange dieser Tag/Ausdruck den Wert WAHR besitzt, ist das Bild für Benutzeraktionen gesperrt, die Grafiken werden aber weiter aktualisiert.
Bild-Logik	**Beim Öffnen**	Kommandos und Ausdrücke, die nach dem Öffnen des Bildes ausgewertet werden sollen, ähnlich einem Mathematik-Arbeitsblatt.
	Wenn geöffnet	Kommandos und Ausdrücke, die bei geöffnetem BILD kontinuierlich ausgewertet werden sollen, ähnlich einem Mathematik-Arbeitsblatt. Wenn außerdem ein Tag/Ausdruck unter **Trigger** konfiguriert wird, werden die aufgeführten Ausdrücke bei jeder Änderung der Triggervariablen bei geöffnetem Bild einmal ausgewertet.
	Beim Schließen	Kommandos und Ausdrücke, die nach dem Schließen des Bildes ausgewertet werden sollen, ähnlich einem Mathematik-Arbeitsblatt.

Bilder und Grafiken

Bereich / Elementname		Beschreibung
Fokus	**Erhalte Fokus beim Öffnen**	Beim Öffnen des Bildes wird automatisch das erste aktivierbare Objekt im Bild aktiviert (gemäß Objekt-ID), so als hätte der Benutzer das Bild mit der Tab-Taste angewählt.
	Tab-Reihenfolge mit anderen Bilder teilen	Wenn der Benutzer das letzte Objekt im BILD mit der Tab-Taste verlässt, wird das nächste geöffnete Bild (gemäß **Tab-Reihenfolge**, s.u.) aktiviert, anstatt zum ersten Objekt im aktuellen Bild zurück zu springen.
	Tab-Reihenfolge	Bestimmt ähnlich der Objekt-ID für Bildobjekte die Tab-Reihenfolge zwischen Bildern, wenn mehrere Bilder geöffnet sind. Wenn der Benutzer das letzte Objekt im Bild mit der Tab-Taste verlässt, wird das geöffnete Bild mit der nächsthöheren Nummer in der Tab-Reihenfolge aktiviert. Jedes Bild sollte über eine eindeutige Tab-Reihenfolgennummer zwischen 0 und 32767 verfügen.

Hintergrundfarbe oder Hintergrundbild eines WinStudio-Bildes

Ein WinStudio-Bild kann entweder einen einfarbigen Hintergrund oder ein editierbares Hintergrundbild haben.

Auswählen der Hintergrundfarbe

Standardmäßig hat ein neu erstelltes Bild einen einfarbig weißen Hintergrund.

Zum Ändern dieser Hintergrundfarbe:

1. Öffnen Sie das Bild zum Bearbeiten im Editor
2. Öffnen Sie das Kontextmenü des Bildes indem Sie an einer freien Stelle des Bildes einen Rechtsklick ausführen. Wählen Sie den Eintrag **Hintergrundfarbe** aus. Es wird ein Standard-Farbauswahlfeld als Kontextmenü angezeigt
3. Verwenden Sie die Farbauswahl zum Wählen einer Farbe. Die Farbe gilt für den gesamten Hintergrund des Bildes



Wenn Sie nur für einen Teil des Bildes eine Hintergrundfarbe bestimmen möchten, zeichnen Sie ein Form-Objekt auf Seite 182 und senden es in den Hintergrund auf Seite 258.

Aktivieren des Hintergrundbilds

Verwenden einer Hintergrundgrafik für ein WinStudio-Bild und anschließendes Bearbeiten:

1. Öffnen Sie das WinStudio-Bild im Editor
2. Öffnen Sie das Kontextmenü des Bildes und wählen den Eintrag **Bildeigenschaften** aus. Der Dialog Bildeigenschaften auf Seite 170 wird angezeigt
3. Wählen Sie die Option **Hintergrund aktivieren**.

   Eine neue BMP-Datei mit dem gleichen Namen wie das WinStudio-Bild wird automatisch im Unterordner "Screens" Ihres Projektordners gespeichert.

   (z.B. **..\Screens\<Bildname>.BMP**)
4. Schließen Sie den Dialog "Bildeigenschaften" durch Klicken auf **OK**
5. Öffnen Sie im Hauptmenü den Eintrag **WinStudio-Bild ▸ Bearbeiten ▸ Hintergrundbild**.

   Microsoft Paint wird gestartet und die BMP-Datei wird zum Bearbeiten geöffnet

Um ein anderes Bildbearbeitungsprogramm als Microsoft Paint zu verwenden, ändern Sie manuell die Datei "Program Settings.ini" durch Hinzufügen der folgenden Einstellung:

```
[Optionen] ImageEditor=dateipfad
```

Bestehenden Grafikdatei als Hintergrundbild

Auswählen einer bestehenden Grafikdatei, vor allem für andere Formate als BMP:

1. Kopieren Sie die Grafikdatei, die Sie verwenden möchten, in den Unterordner "Screens" Ihres Projektordners
2. Öffnen Sie in der Entwicklungsumgebung das entsprechende WinStudio-Bild
3. Wählen Sie im Kontextmenü des Bildes den Eintrag **Bildeigenschaften**. Der Dialog "Bildeigenschaften" wird angezeigt
4. Wählen Sie die Option **Hintergrund aktivieren**
5. Wählen Sie die Option **Gemeinsam benutztes Bild** und geben Sie den Namen der Grafikdatei in das Feld ein (ohne Dateierweiterung)
6. Wählen Sie das Grafikformat der Grafikdatei aus der Liste.

☞ In Projekten für Zielsysteme mit Windows CE werden nur BMP-Dateien unterstützt.

7. Schließen Sie den Dialog "Bildeigenschaften" durch Klicken auf **OK**. Wenn die Entwicklungsanwendung die angegebene Grafikdatei finden und laden kann, wird diese im WinStudio-Bild angezeigt. Wenn diese nicht angezeigt werden kann, erscheint eine Warnmeldung

8.2.2 Ordner Bildgruppen

Der Ordner "Bildgruppen" enthält Elemente, die mehrere Bilder auf Seite 170 zu einer Gruppe zusammenfassen.

Zum Öffnen einer bestimmten Bildgruppe öffnen Sie den Ordner "Bildgruppe" und führen einen Doppelklick auf das Element aus.

Zum Entfernen einer bestimmten Bildgruppe führen Sie einen Rechtsklick auf das entsprechende Element aus und wählen im Kontextmenü den Eintrag "Löschen".

Zum Erstellen einer neuen Bildgruppe:

1. Öffnen Sie das Kontextmenü des Ordners "Bildgruppen" und wählen Sie "Neue Bildgruppe". Es öffnet sich der Konfigurationsdialog für Bildgruppen:

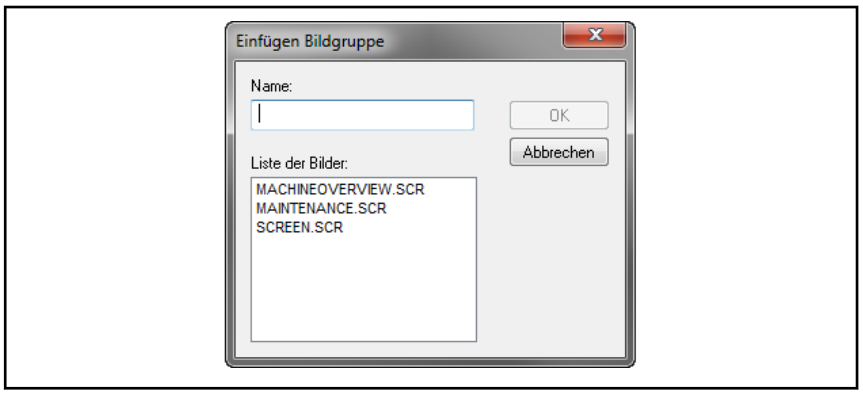


Abb.8-4: Dialog Einfügen einer Bildgruppe

2. Geben Sie im Feld **Name** einen Namen für die neue Bildgruppe ein

3. Wählen Sie aus der Bilderliste diejenigen Bilder aus, die in die Bildgruppe aufgenommen werden sollen.

   Die Liste enthält nur Bilder, die sich aktuell im Ordner "WinStudio" oder in einem unterlagerten Bildordner befinden.

   Zum Auswählen mehrerer Bilder halten Sie beim Klicken auf den Bildnamen die <Strg>-Taste gedrückt.

   Nach Beenden der Auswahl lassen Sie die <Strg>-Taste wieder los.
4. Schließen Sie den Dialog "Bildgruppe einfügen" durch Klicken auf **OK**

8.2.3 Ordner Webseiten

Mit "WinStudio" können Sie WinStudio-Bilder auch in HTML-Format speichern, um diese in einem Web-Browser anzuzeigen.

HTML-Seiten können nicht direkt projektiert werden; sie werden aus bereits bestehenden WinStudio-Bildern erzeugt und im "Web-Ordner" abgelegt.

Zum Erstellen einer HTML-Seite muss zuerst ein WinStudio-Bild erstellt werden. Konfigurieren Sie wie gewohnt ein Bild (Objekte erstellen, Eigenschaften hinzufügen usw.) unter Beachtung, dass dieses Bild als Webseite verwendet werden soll. Speichern Sie das Bild nach dem Fertigstellen wie gewohnt. Dann wählen Sie im Kontextmenü des WinStudio-Bildes im Projektexplorer den Eintrag "Als HTML speichern".

HINWEIS **Die mit "Als HTML speichern" erzeugten Webseiten sind unabhängig von den Bilddateien, aus denen sie erzeugt wurden.**

Wird das WinStudio-Bild verändert, erscheinen die Änderungen erst nach erneutem Ausführen von "Als HTML speichern" auf der Webseite.

Zum Betrachten Ihrer Webseiten während der Laufzeit müssen zuerst die Projekteinstellungen konfiguriert werden.

Wählen Sie dazu den Eintrag **Werkzeuge ▸ Web-Einstellungen** im Kontextmenü des Ordners "WinStudio".

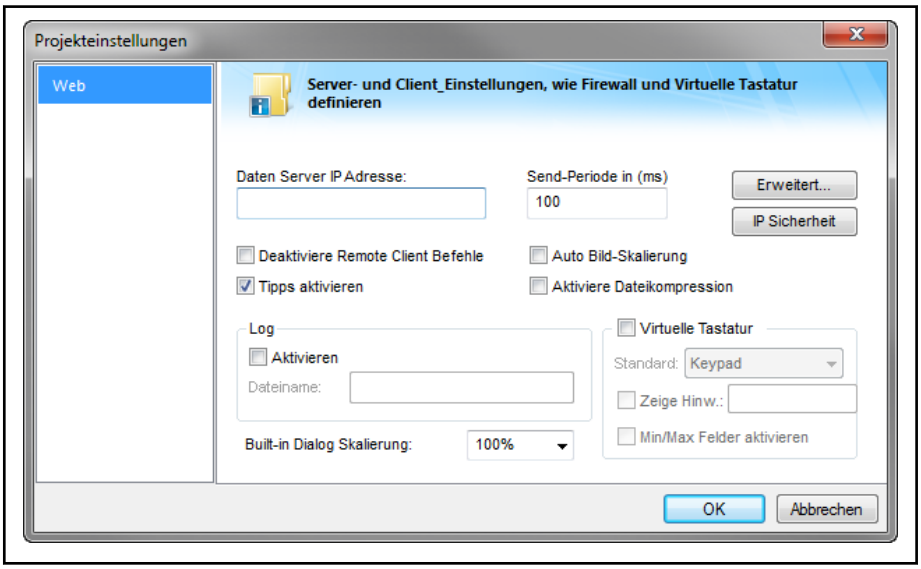


Abb.8-5: Dialog Projekteinstellungen – Registerkarte Web

1. Geben Sie die IP-Adresse des Rechners, auf dem das Projekt ausgeführt wird, in das Feld **IP-Adresse Datenserver** ein

2. Geben Sie einen Wert in das Feld **Sendezeitspanne (ms)** ein, um die Sendezeitspanne (in Millisekunden) festzulegen, mit der Daten zwischen dem Server und den Thin-Client-Stationen ausgetauscht werden
3. *Die folgenden Kontrollkästchen klicken (aktivieren), falls zutreffend:*
   - **Befehle des Remote Client deaktivieren**: Aktivieren Sie diese Option, um zu verhindern, dass ein Remote-Client Befehle von Ihrem Thin Client an Ihren Server schickt.
   - **ToolTips aktivieren**: Aktivieren Sie Option, wenn beim Betrachten von WinStudio-Bildern auf dem Thin Client (Browser) Windows ToolTips angezeigt werden sollen.
   - **Autom. Bildskalierung**: Aktivieren Sie diese Option, um die in einem Browser-Fenster angezeigten WinStudio-Bilder automatisch in der Größe anzupassen. Aktivieren Sie diese Option bei Remote-Ausführung auf einem Thin-Client, wenn "WinStudio" bei Größenänderung des Browserfensters die Bildgröße automatisch ändern soll.
   - **Dateikomprimierung aktivieren**: Aktivieren Sie diese Option, um die Dateien im Unterordner "Web" Ihres Projektordners zu komprimieren. Diese Option ist zur Verringerung der Ladezeit nützlich, besonders bei einer langsamen Verbindung zwischen Ihrem Server und dem Thin Client.
4. Klicken Sie zum Öffnen des Dialogs IP-Sicherheit auf die Schaltfläche "IP-Sicherheit" (siehe Abbildung).

   Mit Hilfe der Parameter dieses Dialogs wird der Bereich von IP-Adressen für die Rechner angegeben, die als Thin Clients auf Ihr Projekt zugreifen dürfen.

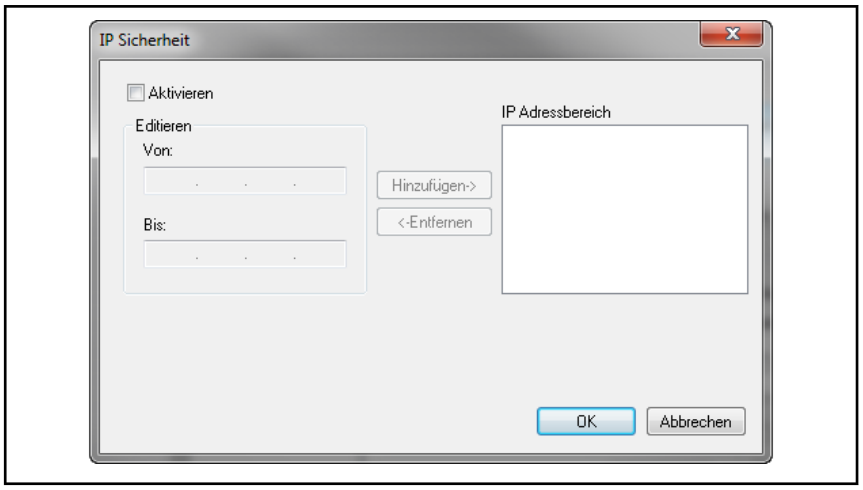


Abb.8-6: Dialog IP-Sicherheit

   Aktivieren Sie die Option "Aktivieren", wenn Sie einen oder mehrere Bereiche von IP-Adressen festlegen möchten, von denen Sie sich über einen Web-Browser verbinden möchten.

   Geben Sie anschließend im Feld "Von" die untere Grenze und im Feld "Bis" die obere Grenze eines IP-Adressbereichs ein.

   Mit den Schaltflächen "Hinzufügen" und "Entfernen" verschieben Sie die IP-Adressen in die Liste "IP-Adressbereich". "WinStudio" erlaubt den in diesem Bereich aufgeführten Rechner IP-Adressen Zugriff auf das Projekt als Thin Clients.
5. Zum Aktivieren der Protokollierung für den Thin Client klicken (aktivieren) Sie im Bereich "Log" das Kontrollkästchen "Aktivieren" und geben

im Feld "Dateiname" einen Dateinamen ein, um auf der Thin-Client-Station eine Protokolldatei zu erzeugen. Diese Protokolldatei kann zu Fehleranalysezwecken eingesetzt werden.

6. Schließen Sie den Dialog "Projekteinstellungen" durch Klicken auf **OK**.

☞ Wenn im Dialog "Projekteinstellungen" Web-Daten verändert werden, muss das Projekt erneut überprüft werden, damit die neuen Einstellungen wirksam werden. Zum Überprüfen des Projekts wählen Sie den Eintrag "Anwendung verifizieren" im Kontextmenü des Ordners "WinStudio". (Wenn im Entwicklungssystem Fenster geöffnet sind, fordert "WinStudio" Sie auf, diese vor dem Überprüfen des Projekts zu schließen).

Da die Webseiten Daten aus dem Projekt mit Hilfe des Web-Servers anzeigen, müssen zur Darstellung Ihrer Webseiten sowohl das Laufzeitsystem als auch der Web-Server und der TCP/IP-Server ausgeführt werden.

Weitere Informationen zu WinStudio Web-Anwendungen finden Sie unter Bereitstellung als Web-Anwendung auf Seite 449.

8.3 Verwenden von Bildobjekten und Animationen

8.3.1 Bearbeiten von Bildobjekten

Bearbeiten von Bildobjekten, Allgemeines

Für das Bearbeiten von WinStudio-Bildern stehen neben den Funktionen im Kontextmenü des Bild-Editors zahlreiche Funktionen in den Symbolleisten zur Verfügung.

Um Symbolleisten ein- oder auszublenden führen Sie einen Rechtsklick im Symbolleistenbereich aus und wählen die gewünschten Symbolleisten an bzw. ab.

Alle Funktionen, die in den WinStudio-Symbolleisten verfügbar sind, finden Sie zusätzlich im Hauptmenü von IndraWorks im Menüpunkt " WinStudio-Bild".

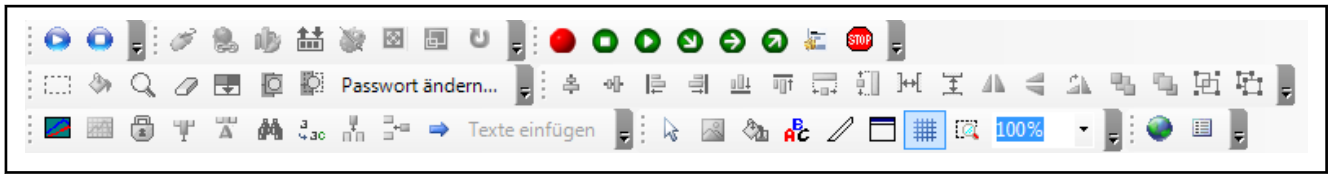


Abb.8-7: Symbolleisten in WinStudio

Bearbeiten von Objekteigenschaften

Der Objekteigenschaftsdialog zeigt die konfigurierbaren Eigenschaften eines Bildobjekts oder einer Animation an. Jeder Typ von Bildobjekten oder Animationen hat seine spezifischen Eigenschaften, aber alle werden im gleichen Grunddialog angezeigt.

Öffnen des Dialogs

Um den Dialog "Objekteigenschaften" zu öffnen wählen Sie eine der folgenden Möglichkeiten:

- Wählen Sie das Bildobjekt aus und drücken Sie dann <Alt>+<Eingabe>
- Führen sie einen Rechtsklick auf das Bildobjekt aus und wählen dann **Eigenschaften** aus dem Kontextmenü
- Führen Sie einen Doppelklick auf das Bildobjekt aus.

Dialog im Detail

Folgende Eigenschaften sind allen Objekten gemeinsam:

	Beschreibung
Pin-Nadel 	Wenn die Pin-Nadel gedrückt ist, bleibt das Fenster "Objekteigenschaften" geöffnet, solange der Bildeditor geöffnet ist. Sie können in diesem Modus die Eigenschafen aller Bildobjekte nacheinander ansehen oder bearbeiten, ohne den Dialog jeweils zu schließen und erneut zu öffnen.
Ersetzen	Öffnet den Dialog "Ersetzen", zum Ersetzen von Zeichenketten, Tags oder Eigenschaften auf Seite 179 für das gewählte Objekt oder die gewählte Objektgruppe.
Hinweis	Anzeige eines ToolTips während der Laufzeit, wenn der Benutzer den Cursor über das Objekt bewegt. Dies kann als schnelle Hilfestellung für den Benutzer verwendet werden. Der Text im Feld **Hinweis** wird außerdem temporär in den System-Tag auf Seite 121 "Hint" geschrieben, so dass beim Bewegen des Cursors über ein bestimmtes Objekt eine Aktion gemäß des Werts dieses Tags ausgelöst werden kann. Zum Anzeigen von Hinweisen/ToolTips während der Laufzeit muss im Dialog "Projekteinstellungen" die Option **ToolTips aktivieren** ausgewählt werden. Sie können diese Funktion getrennt für den Standard-Viewer (auf der Registerkarte **Viewer** in den Projekteigenschaften von WinStudio) und für Web-Thin-Clients (im Dialog **Web-Einstellungen**) aktivieren/deaktivieren.
	Die Combobox auf der rechten Seite des Dialogs erlaubt das Auswählen von bestimmten Objekten innerhalb einer, Objektgruppe oder von Animationen zum Bearbeiten.

Ersetzen von Projekt-Tags in einem Dokument oder Bildobjekt

Zum Ersetzen aller in einem Bildobjekt vorkommenden Tags führen Sie einen Doppelklick auf das Objekt aus, um den Dialog "Objekt-Eigenschaften" zu öffnen und klicken dann auf **Ersetzen**.

Zum Ersetzen aller im aktuellen Dokument vorkommender Tags führen Sie eine der beiden Aktionen aus:

- Öffnen Sie das Menü "WinStudio-Bild" im Hauptmenü von IndraWorks und wählen Sie den Eintrag **Variablen und Strings im aktuellen Dokument ersetzen**
- Blenden Sie die Symbolleiste "WinStudio Tabellenfunktionen" ein und klicken Sie dort auf die Schaltfläche **Variablen und Strings im aktuellen Dokument ersetzen**

Im beiden Fällen öffnet sich der Dialog "Ersetzen", der im folgenden beschrieben wird.

Durch Klicken auf die Registerkarte **Ganzer Tag-Name** können ein oder mehrere Tags ersetzt werden. In der linken Spalte **Von** finden Sie die aktuell verwendete Tags. In der rechten Spalte **Nach** können Sie die neuen Namen eintragen.

Bilder und Grafiken

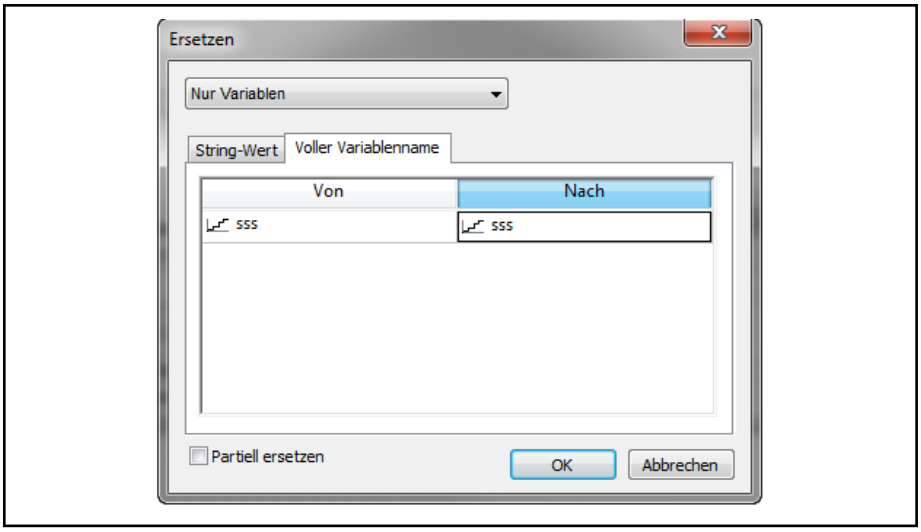


Abb.8-8: Dialog "Ersetzen", Registerkarte"Ganzer Tag-Name"

Bitte beachten Sie, dass hierdurch keine Tags "umbenannt" oder "gelöscht" werden – es werden nur die im Objekt verwendeten Tags durch andere Tags aus der Datenbank ersetzt.

Die Aktivierung der Option "Partiell ersetzen" erleichtert das Ersetzen von Parametern (z.B. Array Index) innerhalb von Klassen und Arrays.

Es können auch eine oder mehrere Zeichenketten (z.B. Schaltflächenbeschriftungen, Beschreibungstexte) durch Klicken auf die Registerkarte **String-Wert** ersetzt werden.

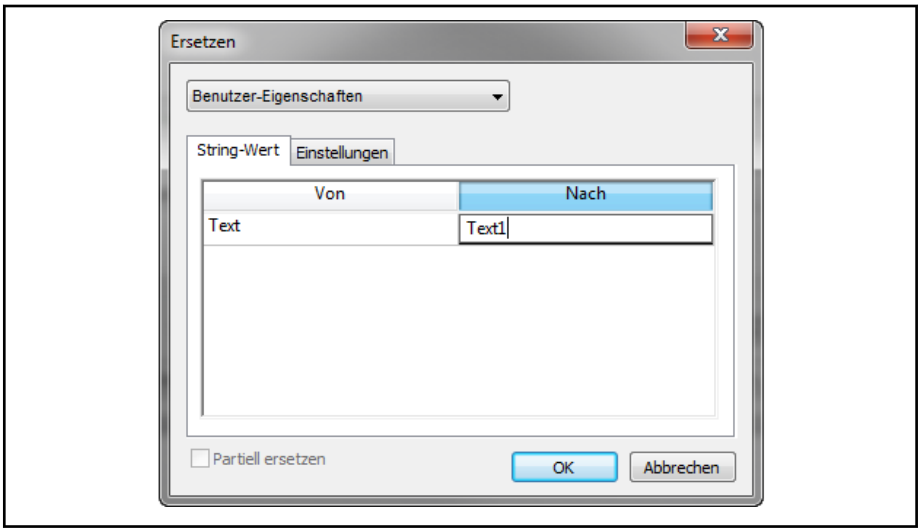


Abb.8-9: Dialog "Ersetzen", Registerkarte"Zeichenkette"

Klicken Sie auf **OK**, wenn Sie alle Einträge bearbeiten haben.

Anordnen von Objekten

Die Symbolleisten "WinStudio Ausrichtungsfunktionen" und "WinStudio Objekteigenschaften" bieten zahlreiche Funktionen zum Ausrichten, Gruppieren und zur Gestaltung von Bildobjekten an.



Abb.8-10: Symbolleiste "Ausrichtungsfunktionen"

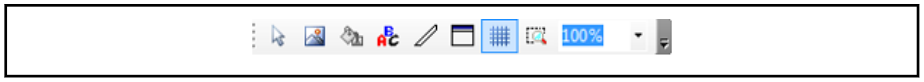


Abb.8-11: Symbolleiste "Objekteigenschaften"

Einige Funktionen in den Symbolleisten sind nur verfügbar, wenn ein oder mehrere Objekte im WinStudio-Bild ausgewählt wurden.

Wenn man die Maus über ein aktives "Werkzeug" bewegt, wird die Funktion im Klartext angezeigt.

Die Werkzeuge sind in die folgenden Gruppen gegliedert:

- **Anordnen**: Anordnen von Objekten in einem WinStudio-Bild, einschließlich
  - In den Vordergrund und In den Hintergrund auf Seite 258
  - Nach Vorne und Nach Hinten, eine Ebene auf Seite 259
  - Objekte gruppieren und Gruppierung aufheben auf Seite 259
  - Ausrichten, Zentrieren und Verteilen auf Seite 260
  - Drehen und Spiegeln auf Seite 262
- Position anpassen auf Seite 263
- **Größe**: Gleiche Breite / Gleiche Höhe für ausgewählte Objekte auf Seite 263 eines Bildobjekts in einem WinStudio-Bild
- **Stil**: Ändern der Farbgebung von Bildobjekten
  - Füllfarbe auf Seite 264 eines Bildobjekts
  - Linienfarbe auf Seite 265 eines Bildobjekts
- **Farbe einer Komponente bearbeiten**: Farbschnittstelle auf Seite 265
- **Schriftarten**: Ändern der Schriftarten auf Seite 269 eines Bildobjekts.

Verschieben von Objekten im Bild-Editor sperren

Das Verschieben von Objekten im Bild-Editor kann gesperrt werden, um nach dem endgültigen Festlegen des Bild-Layouts unbeabsichtigtes Ändern der Positionen von Bild-Objekten zu verhindern.

- Öffnen Sie im WinStudio-Bild-Editor das Kontextmenü und wählen den Eintrag **Verschieben sperren**.
- Um das Verschieben von Bild-Objekten wieder zuzulassen, führen Sie die Funktion **Verschieben sperren** erneut aus.

Alternativ lässt sich die Funktion "Verschieben Sperren" mit der Tastenkombination <Strg>+<D> aktivieren bzw. deaktivieren.

Rückgängig

Im WinStudio-Bildeditor können Sie die letzte Aktion rückgängig machen, indem Sie in der Symbolleiste "Standard" die Schaltfläche **Rückgängig** wählen.

Alternativ können Sie die Tastenkombination <Strg>+<Z> verwenden. Auf diese Weise können Sie bis zu 20 Aktionen vor der letzten Aktion rückgängig machen.

(Aktionen der Symbolleiste "WinStudio-Objekteigenschaften" erhöhen nicht die Anzahl der **Rückgängig**-Schritte, d.h. die Einstellungen können nicht mit dieser Funktion rückgängig gemacht werden.)

Rastereinstellungen

Zum Anzeigen/Verbergen des Rasters im Bild-Editor klicken Sie in der Symbolleiste "WinStudio-Objekteigenschaften" auf **Raster ein-/ausschalten**.

Bilder und Grafiken

Zum Bearbeiten der Rastereinstellungen öffnen Sie den Dialog "Rastereinstellungen". Führen Sie dazu auf einer freien Stelle im Bild-Editor einen Rechtsklick aus, gefolgt von einem Klick auf **Rastereinstellungen** im Kontextmenü.

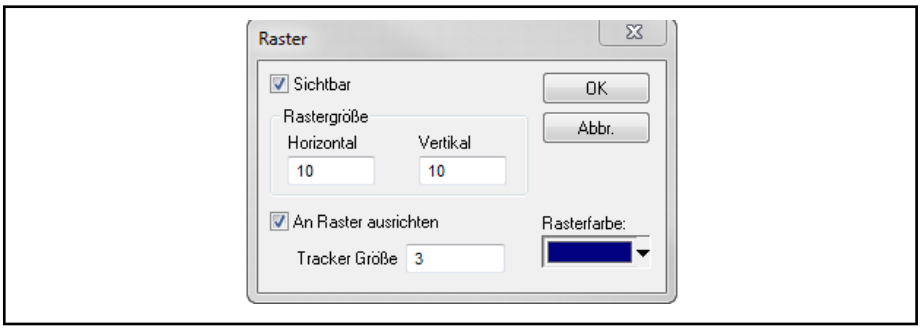


Abb.8-12: Dialog Rastereinstellungen

8.3.2 Statische Objekte

Statische Objekte, Allgemeines

Wenn Sie den WinStudio-Editor öffnen, ist das Werkzeugfenster "WinStudio Screen Tools" verfügbar.

Das Werkzeugfenster enthält verschiedene Objekte, die Sie in ein WinStudio-Bild einfügen können: Rechtecke, Polygone, Linien, Textfelder etc.

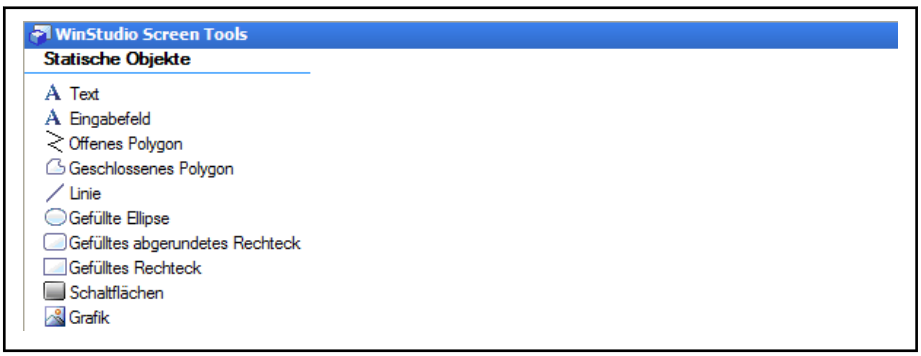


Abb.8-13: Werkzeugfenster "WinStudio Screen Tools"

Objekt Linie

Klicken Sie zum Zeichnen einer orthogonalen Linie auf den Eintrag **Linie** im Werkzeugfenster.

Danach klicken Sie in den Zeichenbereich des WinStudio-Bildeditors:

1. Klicken Sie mit der linken Maustaste, um den Startpunkt der Linie festzulegen
2. Ziehen Sie die Linie mit dem Cursor auf die gewünschte Länge
3. Zum Ausrichten der Linie ziehen Sie an einem der markierten Endpunkte.
4. Zur Ansicht der Objekteigenschaften führen sie einen Doppelklick auf das Objekt aus.

Der Dialog "Objekteigenschaften" wird wie folgt angezeigt:

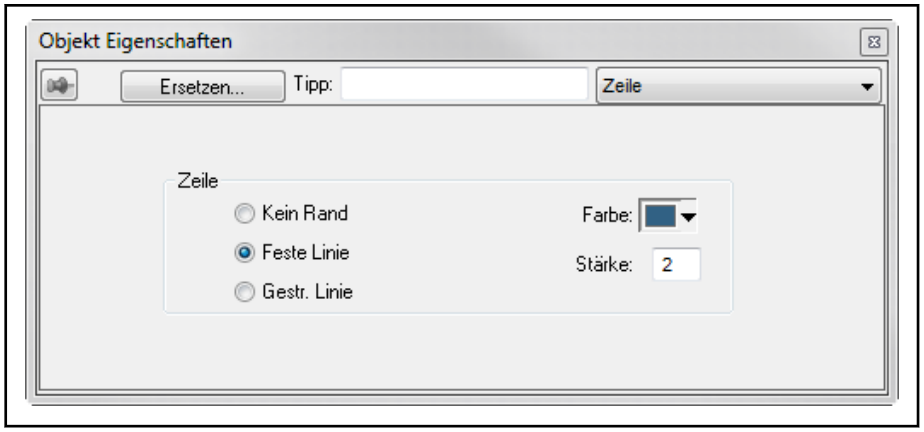


Abb.8-14: Objekteigenschaften: Line

Verwenden Sie den Dialog "Objekteigenschaften" zum Festlegen der folgenden Parameter für die Linie:

- **Linie**: Legen Sie durch Klicken auf die Schaltfläche **Ohne Linie**, **Durchgehende Linie** oder **Gestrichelte Linie** eine Linienart fest
- **Farbe**: Legen Sie durch Klicken auf die Schaltfläche **Farbe** eine Linienfarbe fest. Wenn sich der Dialog "Farbe" öffnet, wählen Sie eine Farbe durch Anklicken aus und schließen den Dialog wieder
- **Stärke**: Legen Sie durch Eingabe einer Zahl in das Textfeld die Linienbreite (in Pixel) fest

Objekt Offenes Polygon

Klicken Sie im Werkzeugfenster **WinStudio Screen Tools** auf **Offenes Polygon**, um ein offenes Vieleck mit Rand in der festgelegten Vordergrundfarbe zu zeichnen.

Zeichnen eines offenen Polygons im Zeichnungsbereich:

1. Klicken Sie mit der linken Maustaste, um den Startpunkt des Polygons festzulegen.
2. Bewegen Sie den Cursor an eine neue Stelle und klicken Sie wieder, um den zweiten Eckpunkt festzulegen.
3. Wiederholen Sie diesen Vorgang, bis das gewünschte Polygon gezeichnet ist.
4. Mit einem Doppelklick wird das Zeichnen des Polygons beendet.

   Zum Ändern der Form eines Polygons nach dem Zeichnen wählen Sie es aus und ziehen an einem der Eckpunkte.



Wenn die einzelnen Punkte eines Polygons nicht gezogen werden können, sind sie eventuell gruppiert auf Seite 259. Um die Gruppierung der Eckpunkte aufzuheben, führen sie einen Rechtsklick auf das Polygon aus und wählen **Gruppierung aufheben** aus dem Kontextmenü.

Zur Anzeige der Objekteigenschaften führen sie einen Doppelklick auf das Polygon-Objekt aus. Der Dialog "Objekteigenschaften" wird wie folgt angezeigt.

Bilder und Grafiken



Abb.8-15: Objekteigenschaften: Offenes Polygon

Verwenden Sie den Dialog "Objekteigenschaften" zum Festlegen der folgenden Parameter für das Polygon:

- **Linie**: Legen Sie durch Klicken auf die Schaltfläche **Ohne Linie**, **Durchgehende Linie** oder **Gestrichelte Linie** eine Linienart für den Rahmen fest.
- **Farbe**: Legen Sie durch Klicken auf die Schaltfläche **Farbe** eine Rahmenlinienfarbe fest. Wenn sich der Dialog Farbe öffnet, wählen Sie eine Farbe durch Anklicken aus und schließen den Dialog wieder.
- **Stärke**: Legen Sie durch Eingabe einer Zahl in das Textfeld die Rahmenlinienbreite (in Pixel) fest.

Objekt Geschlossenes Polygon

Klicken Sie im Werkzeugfenster **WinStudio Screen Tools** auf **Geschlossenes Polygon**, um ein geschlossenes Vieleck mit Rand in der festgelegten Vordergrundfarbe zu zeichnen.

Zeichnen eines geschlossenen Polygons im Zeichnungsbereich:

1. Klicken Sie mit der linken Maustaste, um den Startpunkt des Polygons festzulegen.
2. Bewegen Sie den Cursor an eine neue Stelle und klicken Sie wieder, um den zweiten Eckpunkt festzulegen.
3. Wiederholen Sie diesen Vorgang, bis das gewünschte Polygon gezeichnet ist.
4. Mit einem Doppelklick oder Rechtsklick wird das Zeichnen des Polygons beendet.
5. Zur Ansicht der Objekteigenschaften führen sie einen Doppelklick auf das Polygonobjekt aus.

   Zum Ändern der Form eines Polygons nach dem Zeichnen wählen Sie es aus und ziehen an einem der Eckpunkte.



Wenn die einzelnen Punkte eines Polygons nicht gezogen werden können, sind sie eventuell gruppiert auf Seite 259. Um die Gruppierung der Eckpunkte aufzuheben, führen sie einen Rechtsklick auf das Polygon aus und wählen **Gruppierung aufheben** aus dem Kontextmenü.

Der Dialog "Objekteigenschaften" wird wie folgt angezeigt.



Abb.8-16: Dialog Objekteigenschaften: Geschlossenes Polygon

Verwenden Sie den Dialog "Objekteigenschaften" zum Festlegen der folgenden Parameter für das Polygon:

- **Typ**: Legen Sie durch Klicken auf die Schaltfläche **Ohne Linie**, **Durchgehende Linie** oder **Gestrichelte Linie** eine Linienart für den Rahmen fest.
- **Farbe**: Legen Sie durch Klicken auf die Schaltfläche **Farbe** eine Rahmenlinienfarbe fest. Wenn sich der Dialog "Farbe" öffnet, wählen Sie eine Farbe durch Anklicken aus und schließen den Dialog wieder.
- **Stärke**: Legen Sie durch Eingabe einer Zahl in das Textfeld die Rahmenlinienbreite (in Pixel) fest.
- **Fülleffekt**: Zum Festlegen des Fülleffekts für das Polygon klicken Sie auf **Ungefüllt** oder **Füllen**.

  Wenn die Option **Füllen** ausgewählt ist, kann durch Klicken auf die Schaltfläche **Farbe** eine Füllfarbe ausgewählt werden. Wenn der Dialog "Farbe" angezeigt wird, wählen Sie eine Farbe durch Anklicken aus und schließen den Dialog wieder.

Objekt Rechteck

Zum Erstellen von Rechtecken klicken Sie im Werkzeugfenster **WinStudio Screen Tools** auf **Gefülltes Rechteck** und gehen wie folgt vor:

1. Klicken Sie in den Zeichnungsbereich und ziehen mit Maus/Cursor, um das Rechteck zu zeichnen.
2. Wenn das Rechteck die gewünschte Größe hat, lassen Sie die Maustaste los.
3. Mit einem Doppelklick auf das Objekt öffnet sich der Dialog "Objekteigenschaften".



Abb.8-17: Objekteigenschaften: Rechteck

Verwenden Sie den Dialog "Objekteigenschaften" zum Festlegen der folgenden Parameter für das Rechteck:

- **Typ**: Legen Sie durch Klicken auf **Ohne**, **Durchgehend**, **Gestrichelt**, **Geprägt**, **Relief** oder **Gravur** eine Linienart für den Rahmen fest
- **Farbe**: Legen Sie durch Klicken auf die Schaltfläche **Farbe** zum Öffnen des Dialogs "Farbe" eine Rahmenlinienfarbe fest. Klicken Sie auf die gewünschte Farbe und schließen Sie den Dialog wieder
- **Stärke**: Legen Sie durch Eingabe einer Zahl in das entsprechende Textfeld die Rahmenlinienbreite (in Pixel) fest
- **Hintergrund**: Zum Festlegen der Hintergrundfarbe für das Rechteck klicken Sie auf **Ungefüllt** oder **Füllen**

  Wenn die Option **Füllen** ausgewählt ist, kann durch Klicken auf das Farbrechteck eine Füllfarbe ausgewählt werden. Wenn der Dialog Farbe angezeigt wird, wählen Sie eine Farbe durch Anklicken aus und schließen den Dialog wieder.
- **Beschriftung**: Klicken Sie auf diese Schaltfläche zum Öffnen des Dialogs "Beschriftung", in dem ein Anzeigetext eingegeben werden kann

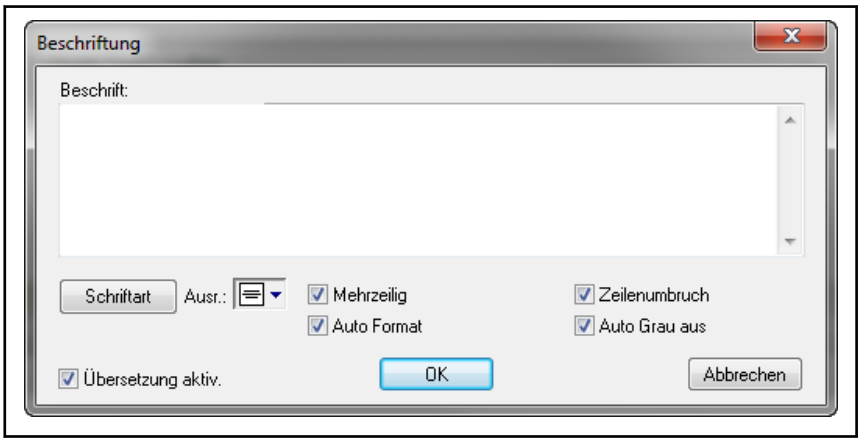


Abb.8-18: Dialog Beschriftung

  - **Beschriftung**: Geben Sie den Text ein, den Sie in dem Rechteckobjekt anzeigen möchten. Die Beschriftung kann Tags enthalten, wenn diese in geschweifte Klammern gesetzt werden (z.B. **{tagname}**).
  - **Schriftart**: Legen Sie durch Klicken auf die Schaltfläche **Schriftart** eine Schriftart für den Text fest.
  - **Ausrichten**: Legen Sie die Ausrichtung für die Beschriftung des Rechtecks fest.
  - **Mehrzeilig**: Durch Aktivieren dieser Option wird die Beschriftung im Rechteck in mehr als einer Zeile angezeigt.
  - **Auto-Format**: Wenn diese Option aktiviert ist und die Beschriftung einen Dezimalwert in geschweiften Klammern (z.B. **{1,2345}**) oder einen Tag vom Typ Real auf Seite 124 enthält (siehe **Beschriftung** oben), dann wird der Wert gemäß der von der Funktion SetDecimalPoints auf Seite 681 erstellten virtuelle Tabelle formatiert.
  - **Zeilenumbruch**: Wenn diese Option aktiviert ist, fügt das Objekt bei Bedarf automatisch einen Zeilenumbruch ein.
  - **Automatisch ausblenden**: Die Beschriftung wird zur Laufzeit ausgegraut dargestellt, wenn die auf das Rechteck angewendete Befehlsanimation auf Seite 243 durch das Feld **Deaktivieren** oder durch das Sicherheitssystem deaktiviert wurde.

– **Übersetzung aktivieren**: Durch Anklicken (Freigeben) wird während der Laufzeit die Übersetzung mit Hilfe des Übersetzungswerkzeugs auf Seite 437 freigegeben.

Objekt Abgerundetes Rechteck

Zum Zeichnen von abgerundeten Rechtecken klicken Sie im Werkzeugfenster **WinStudio Screen Tools** auf **Abgerundetes Rechteck** und gehen wie folgt vor:

1. Klicken Sie in den Zeichnungsbereich und ziehen mit Maus/Cursor, um das Rechteck zu erstellen.
2. Durch Loslassen der Maustaste wird das Zeichnen des Objekts beendet.
3. Mit einem Doppelklick auf das Objekt öffnet sich der Dialog "Objekteigenschaften".

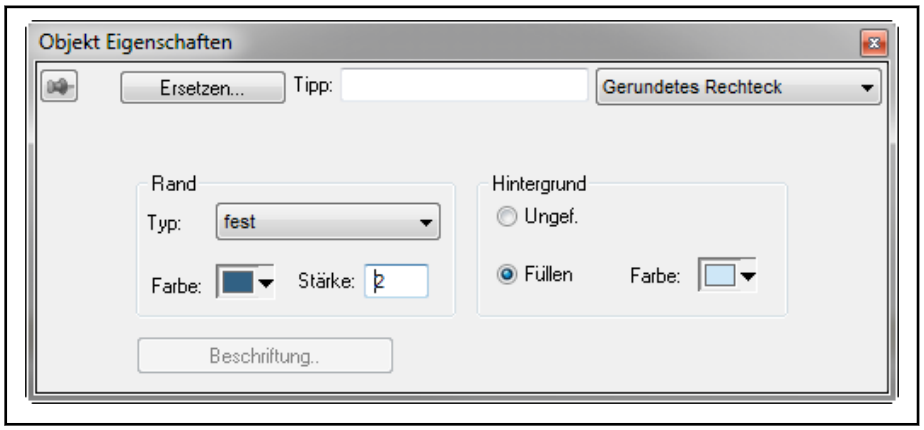


Abb.8-19: Objekteigenschaften: Abgerundetes Rechteck

Das Element "Abgerundetes Rechteck" kann nicht zum Erstellen von Balkendiagrammen für Projekte, die auf Zielsystemen mit Windows CE laufen, verwendet werden.

Ein abgerundetes Rechteck hat an der unteren rechten Ecke einen zusätzlichen Ziehpunkt, mit dem der Bogenwinkel verändert werden kann.

Verwenden Sie den Dialog "Objekteigenschaften" zum Festlegen der folgenden Parameter für das Rechteck:

- **Linie**: Legen Sie durch Klicken auf die Schaltfläche **Ohne Linie**, **Durchgehende Linie** oder **Gestrichelte Linie** eine Linienart fest.
- **Farbe**: Legen Sie durch Klicken auf die Schaltfläche **Farbe** zum Öffnen des Dialogs "Farbe" eine Rahmenlinienfarbe fest. Klicken Sie auf die gewünschte Farbe und schließen Sie den Dialog wieder.
- **Stärke**: Legen Sie durch Eingabe einer Zahl in das entsprechende Textfeld die Rahmenlinienbreite (in Pixel) fest.
- **Hintergrundes**: Zum Festlegen der Hintergrundfarbe für das Rechteck klicken Sie auf **Nicht füllen** oder **Füllen**. Wenn die Option **Füllen** ausgewählt ist, kann durch Klicken auf die Schaltfläche **Farbe** eine Füllfarbe ausgewählt werden.

  Wenn der Dialog "Farbe" angezeigt wird, wählen Sie eine Farbe durch Anklicken aus und schließen den Dialog wieder.

- **Beschriftung**: Diese Option ist für dieses Objekt nicht verfügbar.

Objekt Ellipse

Klicken Sie zum Zeichnen von Ellipsen, Sehnen, Bogen und Ringen im Werkzeugfenster **WinStudio Screen Tools** auf **Ellipse**:

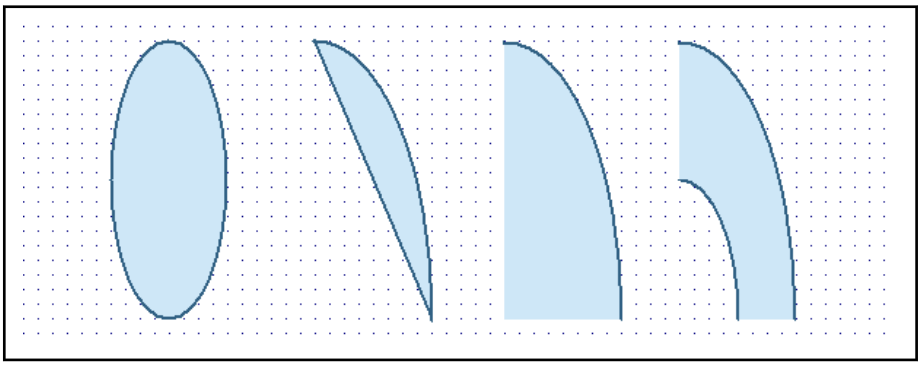


Abb.8-20: Ellipse, Sehne, Bogen und Ring

Die Form "Ring" ist besonders beim Erstellen von Installationszeichnungen hilfreich.

Zum Erstellen einer Ellipse befolgen Sie die folgenden Schritte:

1. Klicken Sie in den Zeichnungsbereich und ziehen Sie mit Maus/Cursor, um die Ellipse zu erstellen.
2. Durch Loslassen der Maustaste wird das Zeichnen der Ellipse beendet.
3. Im Dialog "Objekteigenschaften" kann die Form in eine Sehne, einen Bogen oder einen Ring geändert werden.
4. Mit einem Doppelklick auf das Objekt öffnet sich der Dialog "Objekteigenschaften".

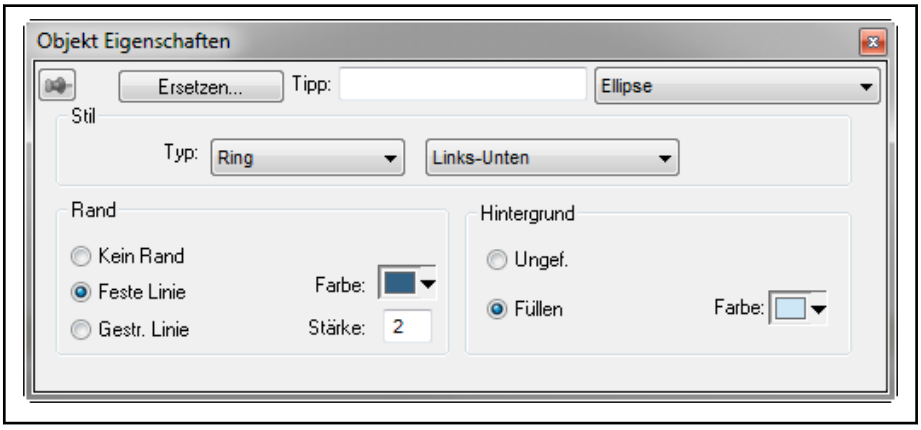


Abb.8-21: Objekteigenschaften: Ellipse

Verwenden Sie den Dialog "Objekteigenschaften" zum Festlegen der folgenden Parameter für die Ellipse:

- **Stil**: Legen Sie durch Auswählen von **Ellipse**, **Bogen**, **Sehne** oder **Ring** aus der Auswahlliste den Stil des Objekts fest.

  Für die Objekte "Bogen, Sehne oder Ring" muss zusätzlich der Quadrant (Links unten, Links oben, Rechts unten, Rechts oben) für die Darstellung festgelegt werden.

  Zur Darstellung eines halbrunden Rohrs werden zum Beispiel zwei **Ring**-Objekte erstellt. Legen Sie für einen **Links unten** und für den ande-

ren **Rechts unten** fest und verbinden dann die beiden Objekte, um eine Halbröhre zu erstellen.

- **Hintergrund**: Zum Festlegen der Hintergrundfarbe für die Ellipse klicken Sie auf **Ungefüllt** oder **Füllen**.

  Wenn die Option **Füllen** ausgewählt ist, kann durch Klicken auf die Schaltfläche **Farbe** eine Füllfarbe ausgewählt werden.

  Wenn der Dialog "Farbe" angezeigt wird, wählen Sie eine Farbe durch Anklicken aus und schließen den Dialog wieder.
- **Rand**: Legen Sie durch Klicken auf die Schaltfläche **Kein Rand**, **Feste Linie** oder **Gestrichelte Linie** eine Linienart für den Rahmen der Ellipse fest.
- **Stärke**: Legen Sie durch Eingabe einer Zahl in das entsprechende Textfeld die Rahmenlinienbreite (in Pixel) fest.

Hinzufügen von Grafiken zu einem WinStudio-Bild

Zum Hinzufügen von Grafiken zu einem WinStudio-BILD kopieren Sie das Bild in die Zwischenablage und fügen es direkt in den Bild-Editor ein.

Dieser Schritt setzt voraus, dass ein WinStudio-Bild im Editor zum Bearbeiten geöffnet ist.

Bitte beachten Sie, dass beim Hinzufügen einer Grafik zu einem Bild bei dieser Methode die Grafikdaten in die Datei des WinStudio-Bildes eingebettet werden.

Mit einem Link zu einer externen Grafikdatei kann das Bild dagegen während der Laufzeit geändert werden. Hierzu wird das Bildobjekt Bildverknüpfung auf Seite 242 verwendet.

---

1. Öffnen Sie die gewünschte Grafik in einem Bildbearbeitungsprogramm, zum Beispiel Microsoft Paint.
2. Markieren Sie die Grafik ganz oder teilweise, entweder mit dem Werkzeug **Auswählen** oder durch Drücken von <Strg>+<A>.
3. Kopieren Sie die Grafikauswahl in die Zwischenablage, entweder mit dem Werkzeug **Kopieren** oder durch Drücken von <Strg>+<C>.
4. Wechseln Sie zum "WinStudio"-Bildeditor.
5. Öffnen Sie das Kontextmenü des geöffneten WinStudio-Bildeditors und wählen den Eintrag **Einfügen** oder drücken Sie <Strg>+<V>. Die Grafikauswahl wird dem WinStudio-Bild als Bitmap-Objekt hinzugefügt.
6. Führen Sie einen Doppelklick auf das Bildobjekt aus. Der Dialog "Objekteigenschaften: Bitmap" wird angezeigt.

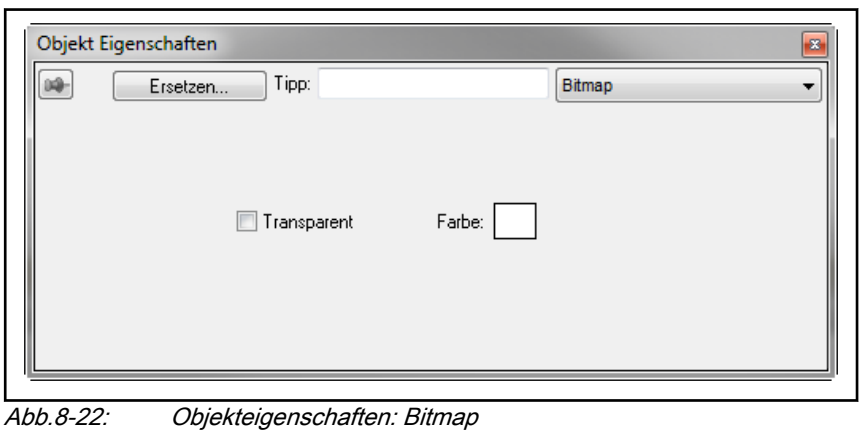


Abb.8-22: Objekteigenschaften: Bitmap

7. Wenn der Bildhintergrund des eingefügten Bildes transparent sein soll, wählen Sie **Transparent** und legen dann mit der Farbauswahl die transparente Farbe fest.
8. Schließen Sie den Dialog "Objekteigenschaften".

8.3.3 Aktive Objekte

Aktive Objekte, Allgemeines

Das Werkzeugfenster "WinStudio Screen Tools" stellt die folgenden Steuerelemente zum Erstellen von interaktiven Objekten zur Verfügung.

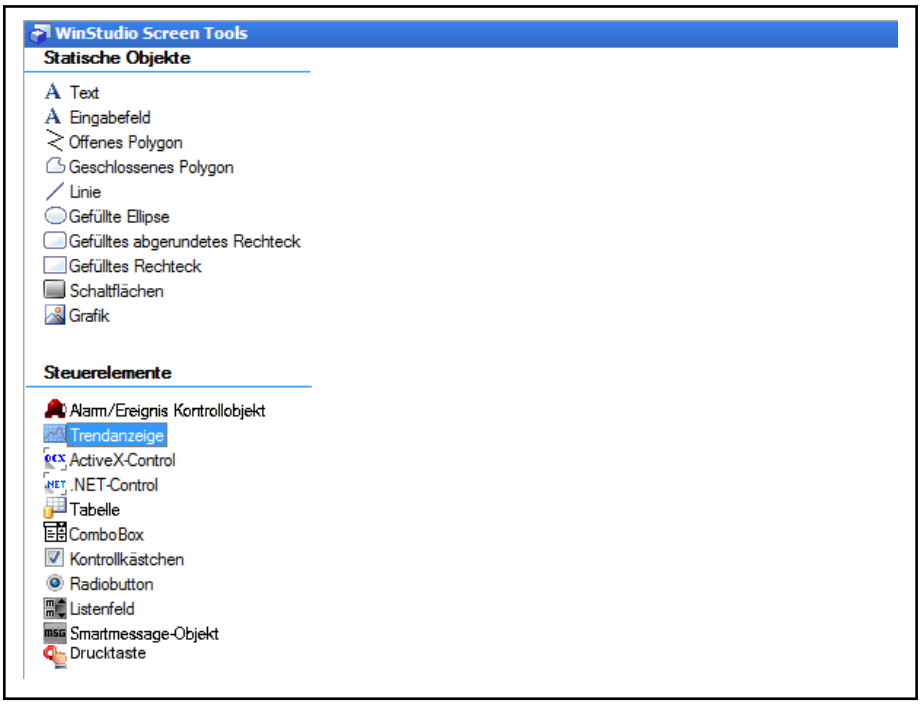


Abb.8-23: WinStudio Screen Tools

Objekt Text

Zum Erstellen von Textobjekten klicken Sie im Werkzeugfenster **WinStudio Screen Tools** auf **Text** und gehen wie folgt vor:

1. Klicken Sie in den Zeichnungsbereich.

   Wenn der Cursor angezeigt wird, können Sie eine Textzeile eingeben.
2. Nach der Eingabe der Zeichenkette führen sie einen Doppelklick auf das neue Textobjekt aus, um den Dialog "Objekteigenschaften" anzuzeigen.

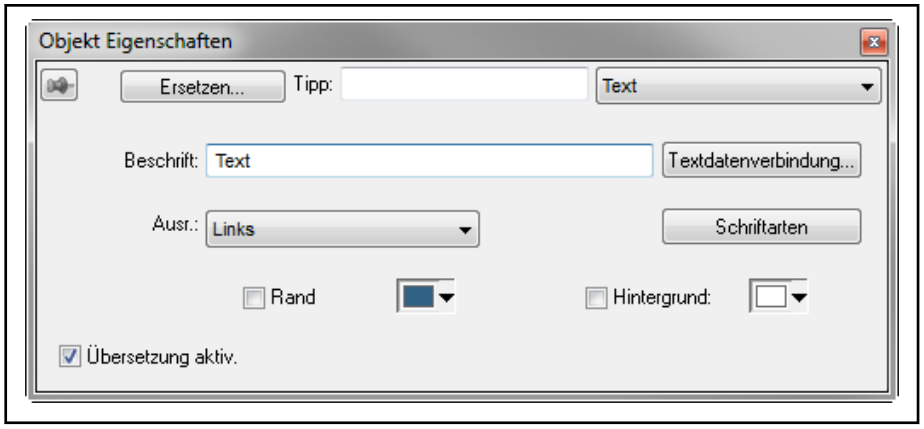


Abb.8-24: Objekteigenschaften: Text

Im Dialog "Objekteigenschaften" werden die folgenden Eigenschaften festgelegt:

- **Beschriftung**: Eingabefeld zum Editieren der im Objekt "Text" eingegebenen Zeichenkette.
- Schaltfläche **Text-Daten-Verknüpfung**: Klicken Sie, um die Animation Text-Daten-Verknüpfung auf Seite 249 auf das Objekt "Text" anzuwenden. Wenn die Beschriftung keine Platzhalterzeichen ("###") für die Text-Daten-Verknüpfung enthält, werden durch Klicken auf diese Schaltfläche diese Zeichen automatisch angehängt.
- **Ausrichten**: Richten Sie den Text durch Auswahl von **Links**, **Zentriert** oder **Rechts** aus der Auswahlliste aus.
- **Schriftarten**: Legen Sie durch Klicken auf die Schaltfläche **Schriftart** folgende Textparameter fest:
  - **Schriftart**
  - **Schriftstil**
  - **Größe**
  - **Effekte**
  - **Farbe**
  - **Skript**
- **Rand**: Aktivieren Sie das Kontrollkästchen "Rand", um das Textfeld mit einem Rahmen zu versehen, und klicken Sie auf die Farbauswahl rechts daneben, um eine passende Rahmenfarbe zu wählen.
- **Hintergrund**: Legen Sie durch Klicken auf die Schaltfläche **Hintergrund** eine Hintergrundfarbe fest.

  Wenn der Dialog "Farbe" angezeigt wird, wählen Sie eine Farbe durch Anklicken aus und schließen den Dialog wieder.
- **Übersetzung aktivieren**: Wenn Sie diese Option aktivieren, wird der Text in die Übersetzungsdatei aufgenommen.

Objekt Schaltfläche

Zum Einfügen von Schaltflächen klicken Sie im Werkzeugfenster **WinStudio Screen Tools** auf **Schaltfläche** und gehen wie folgt vor:

1. Klicken Sie in den Zeichnungsbereich und ziehen mit Maus/Cursor, um die Schaltfläche zu erstellen.
2. Wenn die Schaltfläche die gewünschte Größe hat, lassen Sie die Maustaste los.

3. Mit einem Doppelklick auf das Objekt öffnet sich der Dialog "Objekteigenschaften".

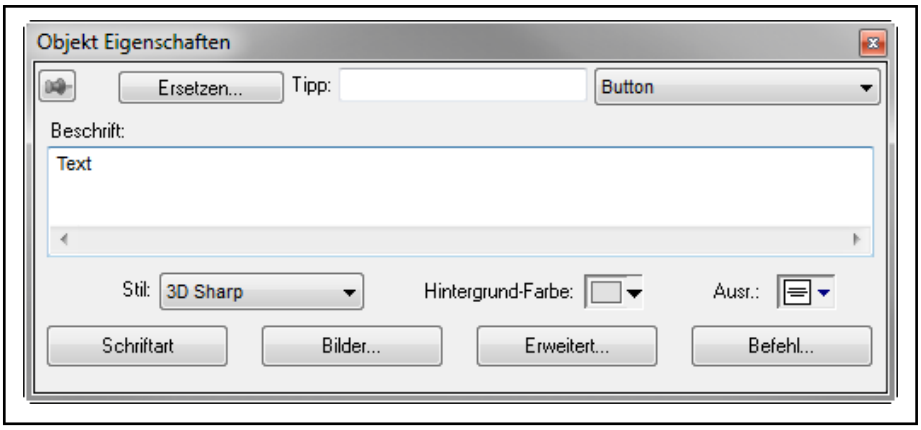


Abb.8-25: Objekteigenschaften: Schaltfläche

Folgende Parameter des Objektes "Schaltfläche" sind im Dialog "Objekteigenschaften" konfigurierbar:

- **Beschriftung**:Texteingabefeld für die Beschriftung der Schaltfläche. Der Anzeigetext kann Tags enthalten. Diese sind in geschweiften Klammern (z.B. `{tagname}`) zu setzen.
- **Stil**: Wählen Sie eine der folgenden Darstellungsarten einer Schaltfläche:
  - **3D Sharp**: Eine erhabene Schaltfläche mit sichtbarer Kontur im Randbereich geeignet für Touchscreen-Bildschirme
  - **3D Soft**: Eine erhabene, abgerundete Schaltfläche, geeignet für Touchscreen-Bildschirme
  - **OS Like**: Eine Schaltfläche passend zum Betriebssystem, auf dem der Projekt-Client ausgeführt wird, geeignet für Windows-Rechner mit Web Thin Client oder Secure Viewer
  - **Standard** : Flache Standard-Schaltfläche

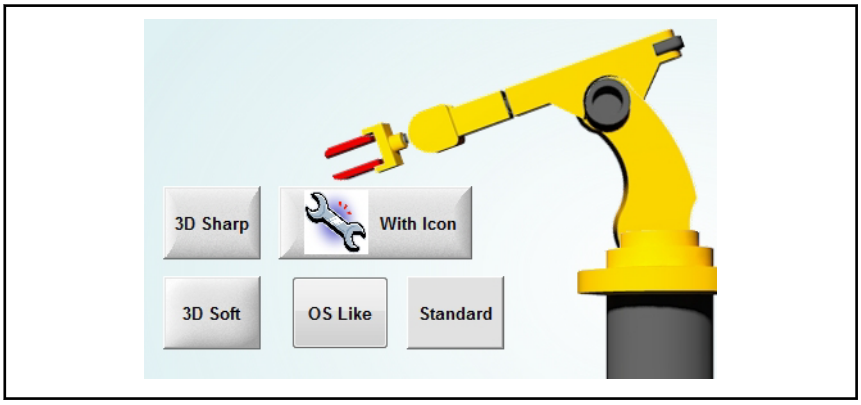


Abb.8-26: Beispiele für Schaltflächen

- **Hintergrundfarbe**: Wählen Sie eine Hintergrundfarbe für die Schaltfläche
- **Ausrichten**: Legen Sie die Ausrichtung für die Beschriftung der Schaltfläche fest
- **Schriftarten**: Legen Sie durch Klicken auf die Schaltfläche **Schriftart** folgende Parameter für die Beschriftung fest:
  - Schriftart
  - Schriftstil

  - Größe
  - Effekte
  - Farbe
- **Bilder**: Durch Klicken auf die Schaltfläche **Bilder** kann eine Grafikdatei in die Schaltfläche eingefügt werden.

  Wenn der Dialog "Bilder" angezeigt wird, können die folgenden Parameter festgelegt werden:

  - **Datei:** Geben Sie den Dateipfad für die Grafikdatei ein. Sie können zum Öffnen eines Standard-Windows-Dateibrowsers auch auf die Schaltfläche Suchen rechts neben dem Kästchen klicken
  - Standardmäßig wird das Bild in seiner Originalgröße angezeigt. Um dies zu ändern, wählen Sie Benutzerdefiniert aus der Liste Größe und konfigurieren dann die gewünschte Breite und Höhe (in Pixel) für das Bild
  - **Position:** Wählen Sie die Position der Grafik in Relation zur Beschriftung
  - **Offset:** Geben Sie den Versatz ein (in Pixel)
  - **Transparente Farbe:** Wählen Sie die transparente Farbe für das Bild aus. In diesen Bereichen ist die Hintergrundfarbe der Schaltfläche sichtbar (siehe oben)
- **Weitere Eigenschaften**: Legen Sie durch Klicken auf die Schaltfläche **Weitere Eigenschaften** die erweiterten Eigenschaften für die Schaltfläche fest.

  Hier können die folgenden Parameter festgelegt werden:

  - **Übersetzung aktivieren (optional):** Durch Anklicken dieser Option, wird der Text in die Übersetzungsdatei aufgenommen.
  - **Mehrzeilig:** Durch Aktivieren dieser Option wird die Beschriftung der Schaltfläche in mehr als einer Zeile angezeigt
  - **Zeilenumbruch:** Wenn diese Option aktiviert ist, fügt das Objekt bei Bedarf automatisch einen Zeilenumbruch ein
  - **Automatisch ausblenden:** Die Beschriftung wird zur Laufzeit ausgegraut dargestellt, wenn die auf die Schaltfläche angewendete Befehlsanimation auf Seite 243 durch das Feld "Deaktivieren" oder durch das Sicherheitssystem auf Seite 407 deaktiviert wurde
  - **Auto-Format:** Wenn diese Option aktiviert ist und die Beschriftung einen Dezimalwert in geschweiften Klammern (z.B. **{1,2345}**) oder einen Tag vom Typ Real auf Seite 124 enthält, dann wird der Wert gemäß der von der Funktion SetDecimalPoints auf Seite SetDecimalPoints erstellten virtuelle Tabelle formatiert
- **Befehl**: Durch Klicken wird der Schaltfläche automatisch eine Befehlsanimation auf Seite 243 hinzugefügt und dann zu den Eigenschaften der Animation gewechselt

Objekt Drucktaste

Klicken Sie im Werkzeugfenster **WinStudio Screen Tools** auf **Drucktaste**, um mit Hilfe der Befehlsanimation auf Seite 243 ein Drucktastenobjekt zu erstellen.

"WinStudio" bietet die folgenden vorkonfigurierten Typen von Drucktasten. Die Tastentypen zeigen das Verhalten der gleichnamigen Hardwaretasten:

- **Momentan** (Standard): Ändert beim Drücken der Drucktaste den Zustand (**Offen** oder **Geschlossen**) und kehrt zum ursprünglichen Zustand

zurück, wenn die Drucktaste losgelassen wird. Diese Drucktasten-Art ist beim Öffnen des WinStudio-Bildes immer in ihrer Normalstellung.

- **Schalter**: Ändert beim Drücken der Drucktaste den Zustand (**Offen** oder **Geschlossen**) und kehrt nicht zum ursprünglichen Zustand zurück, wenn die Drucktaste losgelassen wird. Zum Ändern des aktuellen Zustands muss die Drucktaste erneut gedrückt werden. Diese Drucktasten-Art behält ihren Zustand bei Änderung des WinStudio-Bildes bei.
- **Verriegelt**: Ändert beim Drücken der Drucktaste den Zustand (**Offen** oder **Geschlossen**) und bleibt in diesem Zustand bis sie durch Ändern des Tags "Reset" gelöst wird.

"WinStudio" bietet die folgenden Drucktastenausprägungen an:

- Taste mit Einrahmung und Kontrolleuchte
- Taste ohne Einrahmung (vollflächiges Rechteck) als Standard

Jeweils in 3D-Darstellung oder mit Schatteneffekt.

Hinzufügen von einer oder mehreren vorkonfigurierten Drucktasten zu einem Bild:

1. Klicken Sie auf das Werkzeug **Drucktaste** und positionieren Sie die Maus (Zeiger) im WinStudio-Bildeditor
2. Erstellen Sie die rechteckige Drucktaste durch Anklicken und Ziehen

   Wenn die Schaltfläche die gewünschte Größe hat, lassen Sie die Maustaste los.
3. Mit einem Doppelklick auf das Objekt öffnet sich der Dialog "Objekteigenschaften".



Alternativ können Sie zum Öffnen des Dialogs "Objekteigenschaften" einen Rechtsklick auf das Drucktastenobjekt durchführen oder das Objekt markieren, <Alt>+<Eingabe> drücken und **Eigenschaften** aus dem Kontextmenü auswählen.

Objekteigenschaften: Drucktaste

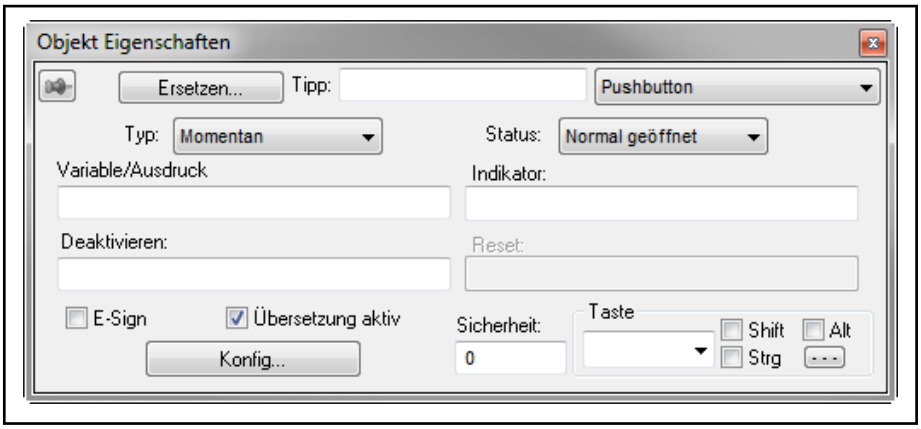


Abb.8-27: Objekteigenschaften: Drucktaste

Dieser Dialog wird zur Festlegung der folgenden Parameter verwendet:

- **Typ**: Klicken Sie zum Auswählen der Drucktasten-Art (**Momentan** (Standard), **Schalter** oder **Verriegelt**).
- **Status**: Wählen Sie die Logik der Tastenfunktion durch Klicken auf das Auswahlfeld "Status". Die Taste kann als "Schließer" (normal offen) oder als "Öffner" (normal geschlossen) projektiert werden.

Zum Beispiel: Die Taste vom Typ "Momentan" wird als "Schließer" (normal offen) projektiert. So lange die Taste nicht betätigt wird, gibt das Laufzeitsystem das Ergebnis "Schalter offen" aus. Die gleiche Taste als "Öffner" (normal geschlossen) projektiert gibt ohne Betätigung zur Laufzeit das Ergebnis "Schalter geschlossen" aus.

- **Tag/ Ausdr**: Eingabefeld zur Auswahl eines WinStudio-Tags oder zur Eingabe eines Programmausdrucks:
  - Geben Sie einen Tag ein dem zur Laufzeit ein bestimmter Wert übergeben wird. Dieser Wert ist abhängig vom Zustand der Taste (z.B. gedrückt) , von der voreingestellten Logik (z.B. normal offen) und von dem im Menü "Konfiguration" parametrierten "Schreibwert".
  - Geben Sie einen Programm-Ausdruck ein, der beim Betätigen der Taste ausgeführt werden soll.

☞ "WinStudio" schreibt das Ergebnis eines Ausdrucks "nicht" in das Feld **Tag/Ausdr** in einen Tag.

- **Indikator**: Geben Sie einen Tag ein, um eine Anzeige zu definieren, mit der die Drucktaste eine bestimmte Farbe annimmt, wenn der Tag-Wert einen von zwei festgelegten Werten annimmt. Es müssen sowohl die Farben als auch die Tag-Werte im Dialog "Konfiguration" definiert werden. Wenn dieses Feld leer bleibt, wechselt die Anzeige beim Drücken der Drucktaste automatisch die Farbe.
- **E-Sign**: Wenn diese Option angewählt ist, wird der Benutzer vor dem Ausführen der Animation zur Eingabe der elektronischen Signatur aufgefordert.
- **Reset** (nur aktiv für Drucktasten vom Typ **Verriegelt**): Geben Sie wie folgt einen Tag zur Steuerung des Verriegelungszustands der Drucktaste ein:
  - Bei Eingabe einer Null bleibt die Drucktaste nach dem Drücken im arretierten Zustand.
  - Bei Eingabe eines Werts ungleich Null wird die Arretierung einer Drucktaste nach dem Drücken aufgehoben. Bevor die Drucktaste erneut gedrückt werden kann, muss der Tag-Wert auf Null zurückgesetzt werden.
- Bereich **Taste**: Legen Sie einen Tastaturcode fest oder erstellen Sie eine Tastenkombination zum Zustandswechsel einer Drucktaste, wenn kein Zeigegerät (Maus oder Touchscreen) vorhanden ist oder wenn Sie zusätzlich zu den Drucktasten Tastaturkürzel erstellen möchten.
- Auswahlliste **Taste**: Geben Sie eine Taste in das Textfeld ein oder wählen Sie eine alphanumerische Taste aus der Auswahlliste. Geben Sie nur einen einzelnen Buchstaben oder eine einzelne Taste ein. In diesem Feld können keine Zahlen eingegeben werden. Klicken (Aktivieren) Sie das Feld **Umschalt**, **Strg** oder **Alt** zum Erstellen einer Tastaturkombination, d.h. die Taste Umschalt, Strg oder Alt muss zusammen mit der in der Auswahlliste festgelegten Taste gedrückt werden. Öffnen Sie den Dialog "Tasten ändern" durch Klicken auf die Such-Schaltfläche **...**, um Ihre Tastaturkombinationen zu verändern. Für Tastaturkombinationen kann die Position der Taste Umschalt, Strg oder Alt auf der Tastatur auf Seite 80 mit **Links**, **Rechts** oder **Links oder rechts** festgelegt werden. Bei Auswahl von **Links oder rechts** wird der Befehl jedes Mal ausgeführt,

wenn eine dieser Tasten in Verbindung mit der in der Auswahlliste festgelegten Taste gedrückt wird.

- **Deaktivieren**: Geben Sie zum Deaktivieren der Drucktaste einen Wert ungleich Null ein; dadurch hat das Drücken dieser Taste keine Auswirkung. Dieses Feld ist standardmäßig leer.
- **Übersetzung aktiv**: Durch Anklicken dieser Option, wird der Text in die Übersetzungsdatei aufgenommen. (Siehe Übersetzungswerkzeug auf Seite 437 für weitere Informationen.)
- **Sicherheit**: Geben Sie einen Wert zur Festlegung einer Sicherheitsstufe (0 bis 255) für diese Schaltfläche ein. Wenn der Benutzer nicht über die festgelegte Sicherheitsstufe verfügt, wird die Drucktaste deaktiviert. Wenn der Benutzer über die entsprechende Sicherheitsstufe verfügt oder wenn dieses Feld leer bleibt, ist die Drucktaste aktiv.
- **Konfig**: Klicken Sie zum Öffnen des Dialogs "Konfiguration", mit dem "Stil" und Zustandsparameter für die Drucktaste festgelegt werden können.

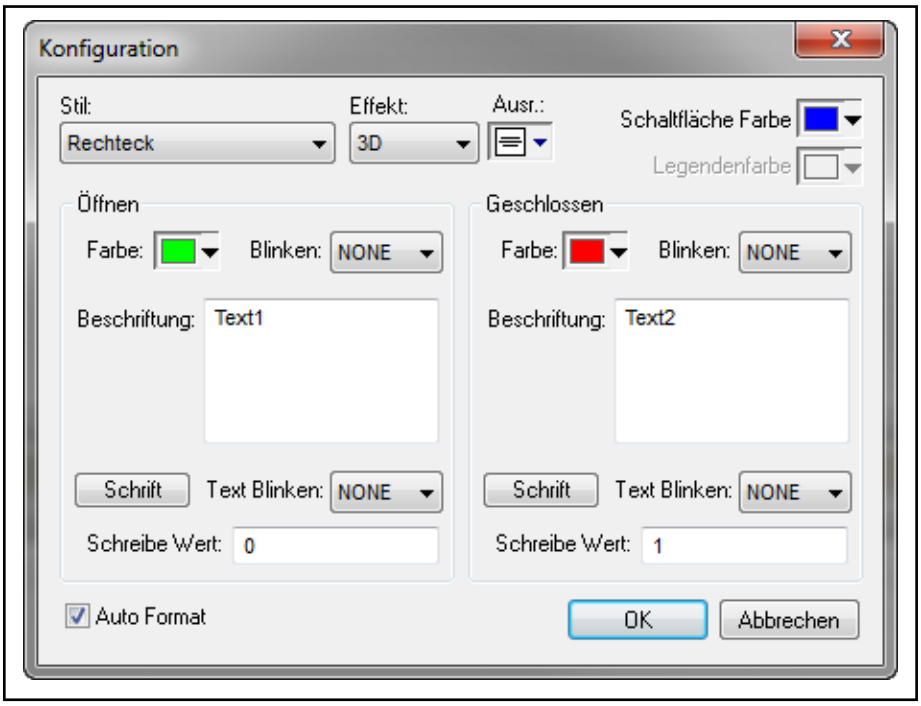


Abb.8-28: Dialog Konfiguration

Dieser Dialog stellt die folgenden Parameter zur Verfügung:

- **Stil**: Klicken Sie auf die Combobox-Schaltfläche zum Auswählen eines Stils für die Drucktaste (**Rechteck** (Standard) oder **Rechteck mit Blende**).
- Combobox **Effekt**: Anklicken, um einen 3D-Effekt für die Drucktaste auszuwählen.
  - **Schwebend** (Standard): Die Drucktaste erscheint als flaches Objekt mit Schatten
  - **3D**: Die Drucktaste hat abgerundete Ecken und scheint sich beim Drücken in dem Bildschirm „hineinzudrücken“.

  Die Parameter **Stil** und **Effekt** können kombiniert werden, um vier verschiedene Drucktasten zu erstellen, wie in der folgenden Abbildung dargestellt:

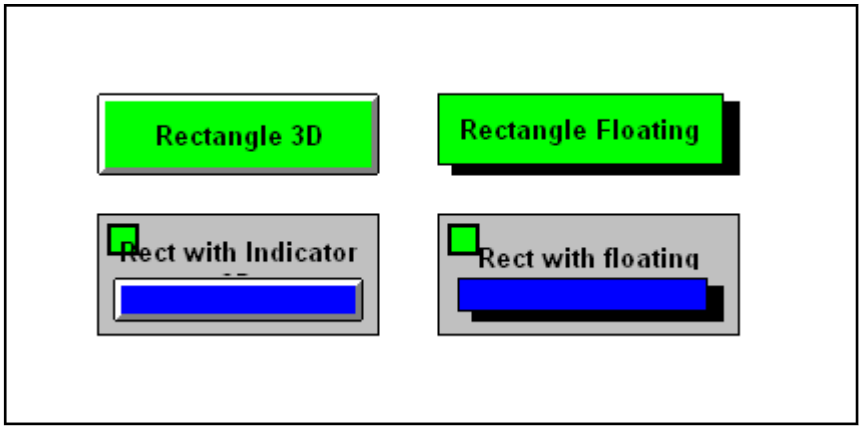


Abb.8-29: Stile für Drucktasten

- **Ausrichten**: Legen Sie die Ausrichtung für die Beschriftung der Drucktaste fest
- **Tastenfarbe**: Anklicken, um die konstante Farbe der Taste festzulegen, wenn es sich um eine Tastenausprägung "mit Einrahmung" handelt

  Wenn es sich um ein Tastenobjekt "ohne Einrahmung" handelt, wird die hier eingestellte Tastenfarbe nicht oder nur abwechselnd eingeblendet sofern die Option "Blinken" angewählt ist.
- **Legendenfarbe**: Anklicken um die Farbe der Einrahmung festzulegen, wenn es sich um eine Tastenausprägung "mit Einrahmung" handelt (Tastenhintergrund = Legende)

  Dieses Feld wird inaktiv, wenn die "Tastenausprägung ohne Einrahmung" gewählt wurde.
- **Offen** und **Geschlossen**: Mit den folgenden Parametern wird das Aussehen des Drucktastenobjekts im offenen und geschlossenen Zustand konfiguriert.
  - **Farbe**: Anklicken zum Festlegen einer Standardfarbe für eine Anzeige in jedem **Zustand**. Die Farbumschaltung ist nur wirksam, wenn es sich um eine Tastenausprägung "ohne Einrahmung" handelt.
  - **Blinken**: Anklicken zum Festlegen ob die im Feld **Farbe** festgelegte Farbe blinkt und wie schnell sie für jeden Zustand blinkt (**Ohne** (kein Blinken, "Standard"), **Langsam** und **Schnell**).

    Wenn für die Farbe Blinken eingestellt ist, wechselt die Taste zwischen der im Feld **Farbe** festgelegten Farbe und der konfigurierten Grundfarbe der Taste (Schaltfläche Farbe) .
  - Textfeld **Beschriftung**: Verwenden Sie dieses Textfeld zur Eingabe einer Beschriftung für die Taste.

    Wenn es sich um eine Tastenausprägung "mit Einrahmung" handelt, wird der Text nicht auf dem Button, sondern auf der Rahmenfläche angezeigt. Die Beschriftung kann Tags enthalten, wenn diese in geschweifte Klammern gesetzt werden (z.B. `{tagname}`).
  - Schaltfläche **Schriftarten**: Anklicken zum Öffnen des Dialogs "Schriftart", in dem die Eigenschaften für die Schrift der Meldung für jeden Zustand festgelegt oder geändert werden können.
  - Auswahlliste **Text Blinken**: Anklicken zum Festlegen, ob der festgelegte Text blinkt und wie schnell er für jeden Zustand blinkt (**Ohne** (kein Blinken, "Standard"), **Langsam** und **Schnell**).

Anders als bei blinkenden Farben wird der blinkende Text ein- und ausgeblendet.

– Auswahlliste **Wert schreiben**: Eingabe eines Wertes in einem der Felder

  Wenn die Drucktaste sich in entsprechenden Zustand befindet (**Offen** oder **Geschlossen**), schreibt "WinStudio" diesen Wert in den im Feld **Tag/Ausdr** festgelegten Tag (Dialog "Objekteigenschaften").

- **Auto-Format**: Wenn diese Option aktiviert ist und die Beschriftung einen Dezimalwert in geschweiften Klammern (z.B. **{1,2345}**) oder einen Tag vom Typ Real auf Seite 124 enthält (siehe **Beschriftung** oben), dann wird der Wert gemäß der von der Funktion SetDecimalPoints auf Seite SetDecimalPoints erstellten virtuelle Tabelle formatiert.

Objekt Kontrollkästchen

Das Objekt Kontrollkästchen ist ein Eingabeelement, mit dem der Bediener eine Funktion auf dem Bildschirm aktivieren oder deaktivieren kann.

Zum Einfügen eines Kontrollkästchen in Ihr WinStudio-Bild klicken Sie im Werkzeugfenster **WinStudio Screen Tools** auf **Kontrollkästchen**:

1. Klicken Sie in den Zeichnungsbereich und ziehen Sie mit Maus/Cursor, um das Kontrollkästchen und dessen Beschriftung zu zeichnen.
2. Wenn das Objekt die gewünschte Größe hat, lassen Sie die Maustaste los.
3. Mit einem Doppelklick auf das Objekt öffnet sich der Dialog "Objekteigenschaften".

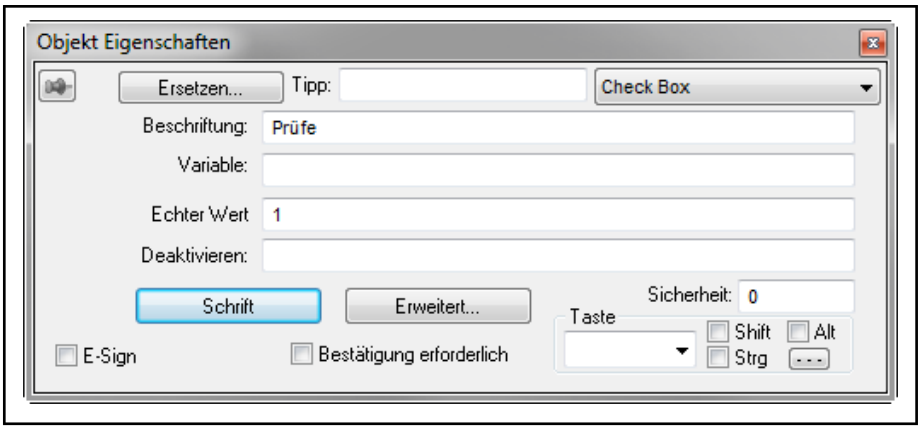


Abb.8-30: Objekteigenschaften: Kontrollkästchen

Zum Ändern der Standardgröße des Kontrollkästchens bearbeiten Sie Ihre Projektdatei (**projekt_name.app**) und fügen die folgende Einstellung hinzu:

[Objekte]

CheckBoxSize=höhe_in_pixel

Diese Einstellung sollte mit Vorsicht verwendet werden, denn sie ändert die Größe aller Kontrollkästchen in Ihrem Projekt. Außerdem funktioniert dies nur für Projekte, die im Projekt-Viewer laufen (Viewer.exe).

Es funktioniert nicht für Projekte, die in CEView oder dem Web Thin Client laufen.

Verwenden Sie den Dialog "Objekteigenschaften" zum Festlegen der folgenden Parameter für das Objekt Kontrollkästchen:

- **Beschriftung**: Legen Sie durch Eingabe von Text in das Textfeld eine Beschriftung fest. Die Beschriftung kann Tags enthalten, wenn diese in geschweifte Klammern gesetzt werden (z.B. `{tagname}`).
- **Schriftarten**: Legen Sie durch Klicken auf die Schaltfläche **Schriftart** eine Schriftart für den Text fest.
- **E-Sign**: Wenn diese Option angewählt ist, wird der Benutzer vor dem Ausführen des Befehls zur Eingabe der elektronischen Signatur aufgefordert.
- Kontrollkästchen **Bestätigen**: Dieses Kästchen anklicken (aktivieren), um sicherzustellen, dass "WinStudio" zur Bestätigung der Aktion während der Laufzeit auffordert.
- Auswahlliste **Taste**: Wählen Sie eine Taste aus der Liste, um diese Taste mit dem Objekt oder der Objektgruppe zu verbinden. Zur Laufzeit kann diese Taste zum Aktivieren/Deaktivieren des Kontrollkästchens gedrückt werden.
  - Klicken (Aktivieren) Sie das Feld **Umschalt**, **Strg** oder **Alt** zum Erstellen einer Tastaturkombination, d.h. die Taste <Umschalt>, <Strg> oder <Alt> muss zusammen mit der in der Auswahlliste festgelegten Taste gedrückt werden.
  - Öffnen Sie den Dialog "Tasten ändern" durch Klicken auf die Such-Schaltfläche **...**, um Ihre Tastaturkombinationen zu verändern. Für Tastaturkombinationen kann die Position der Taste Umschalt, Strg oder Alt auf der Tastatur auf Seite 80 mit **Links**, **Rechts** oder **Links oder rechts** festgelegt werden. Bei Auswahl von **Links oder rechts** wird der Befehl jedes Mal ausgeführt, wenn eine dieser Tasten in Verbindung mit der in der Auswahlliste festgelegten Taste gedrückt wird.
- Feld **Deaktivieren**: Geben Sie in diesem Feld einen Tag oder einen Ausdruck zum Freigeben und Deaktivieren des Objekts ein. Bei Eingabe eines Wertes ungleich Null wird das Objekt Kontrollkästchen deaktiviert.
- Feld **Sicherheit** Geben Sie in diesem Feld einen Wert zur Festlegung einer Sicherheitsstufe für das Objekt ein, wie unter Sicherheit definiert. Wenn ein Benutzer sich anmeldet und nicht über die angegebene Sicherheitsstufe verfügt, deaktiviert "WinStudio" das Objekt.
- Feld **Variable**: Wenn der Benutzer während der Laufzeit auf das Kontrollkästchen klickt, wird der Wert dieses Tags aktualisiert. Wenn keine **Rückmeldung** festgelegt wurde, wird der Wert dieses Tags außerdem zur Anzeige des aktuellen Zustands des Objekts verwendet.
- Feld **Wert Wahr**: Legen Sie einen Wert fest zur Änderung des Steuerungszustands auf WAHR und zur Anzeige, dass das Steuerelement sich im Zustand WAHR befindet. Weitere Informationen über Zustände entnehmen Sie bitte der Zustandstabelle.
- Schaltfläche **Weitere Einstellungen**: Klicken Sie auf diese Schaltfläche zum Öffnen des Dialogs "Weitere Einstellungen":

Bilder und Grafiken

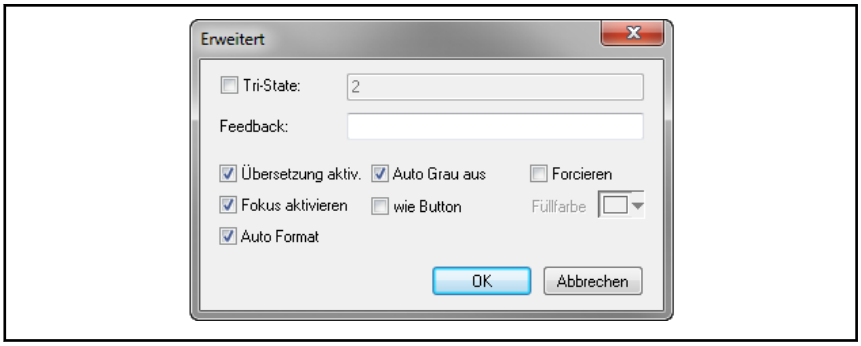


Abb.8-31: Dialog Weitere Einstellungen

– **Dritt-Zustand**: Bei Aktivierung hat das Steuerelement einen dritten Zustand (ausgegraut). Der dritte Zustand wird angezeigt, wenn der im Feld **Rückmeldung** konfigurierte Tag den im Feld **Dritt-Zustand** festgelegten Wert annimmt.

  Wenn das Feld **Rückmeldung** leer bleibt, wird der dritte Zustand angezeigt, wenn der im Feld **Tag** konfigurierte Tag den im Feld **Dritt-Zustand** festgelegten Wert annimmt.

HINWEIS	**Das Feld Dritt-Zustand darf nicht mit dem gleichen Wert wie das Feld Wert Wahr konfiguriert sein. Es darf auch nicht mit einer leeren Zeichenkette belegt sein.**

– **Rückmeldung**: Wert, der den Zustand des Objekts anzeigt (FALSCH, WAHR oder DRITT-ZUSTAND).

  Wenn der in **Rückmeldung** konfigurierte Tag gleich dem in **Wert Wahr** konfigurierten Tag ist, dann wird der Zustand auf WAHR gesetzt.

  Wenn der in **Rückmeldung** konfigurierte Tag gleich dem in **Dritt-Zustand** konfigurierten Tag ist, dann wird der Zustand auf DRITT-ZUSTAND gesetzt.

  Wenn keine dieser Bedingungen erfüllt ist, wird der Zustand auf FALSCH gesetzt.

  Wenn das Feld **Rückmeldung** leer bleibt, wird der im Feld **Tag** konfigurierte Tag als **Rückmeldungs**-Tag verwendet.

– **Übersetzung aktiv**: Durch Anklicken dieser Option, wird der Text in die Übersetzungsdatei aufgenommen.

– **Automatisch ausblenden**: Ändert die Beschriftung des Objekts in grau, wenn es durch das Feld **Deaktivieren** oder durch das Sicherheitssystem deaktiviert ist.

– **Erzwingen**: Dieses Kontrollkästchen anklicken (aktivieren), um das Erkennen einer Tag-Änderung durch die Tag-Datenbank zu erzwingen, wenn der Benutzer auf das Objekt klickt, auch wenn der Wert des entsprechenden Tags sich nicht ändert.

– **Fokus aktivieren**: Wenn diese Option angewählt ist, kann das Objekt während der Laufzeit durch die Navigationstasten aktiviert werden.

– **Wie Button**: Wenn diese Option angewählt ist, wird das Steuerelement nicht als Standard-Kontrollkästchen, sondern als Schaltfläche dargestellt.

- **Füllfarbe**: Festlegen der Füllfarbe für die Schaltfläche. Diese Option ist nur aktiv, wenn die Option **Wie Button** angewählt ist.
- **Auto-Format**: Wenn diese Option aktiviert ist und die Beschriftung einen Dezimalwert in geschweiften Klammern (z.B. **{1,2345}**) oder einen Tag vom Typ Real auf Seite 124 enthält (siehe **Beschriftung** oben), dann wird der Wert gemäß der von der Funktion SetDecimalPoints auf Seite SetDecimalPoints erstellten virtuelle Tabelle formatiert.

Betriebsmodi

Das Objekt Kontrollkästchen hat zwei verschiedene Betriebsmodi: Normal und Dritt-Zustand.

Weitere Informationen unter Betriebsmodi für Objekte Kontrollkästchen und Optionsfeld auf Seite 204.

Objekt Radiobutton

Das Objekt Radiobutton bzw. Optionsfeld ist hilfreich zum Erstellen einer Eingabemaske in welcher der Benutzer eine Option aus mehreren Optionen auswählen kann.

Zum Einfügen eines Optionsfelds in ein WinStudio-Bild klicken Sie im Werkzeugfenster **WinStudio Screen Tools** auf **Radiobutton**:

1. Klicken Sie in den Zeichnungsbereich und ziehen mit Maus/Cursor, um das Optionsfeld und dessen Beschriftung zu zeichnen.
2. Wenn das Objekt die gewünschte Größe hat, lassen Sie die Maustaste los.
3. Mit einem Doppelklick auf das Objekt öffnet sich der Dialog "Objekteigenschaften".

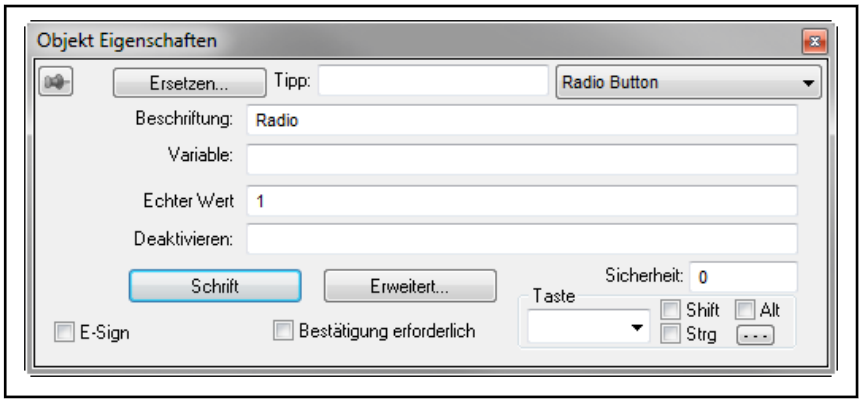


Abb.8-32: Objekteigenschaften: Optionsfeld

Zum Ändern der Standardgröße des Optionsfelds bearbeiten Sie Ihre Projektdatei (**projekt_name.app**) und fügen die folgende Einstellung hinzu:

[Objekte]

RadioButtonSize=höhe_in_pixel

Diese Einstellung sollte mit Vorsicht verwendet werden, denn sie ändert die Größe aller Optionsfelder in Ihrem Projekt.

Außerdem funktioniert dies nur für Projekte, die im Projekt-Viewer laufen (Viewer.exe).

Es funktioniert nicht für Projekte, die in CEView oder dem Web Thin Client laufen.

Bilder und Grafiken

Verwenden Sie den Dialog "Objekteigenschaften" zum Festlegen der folgenden Parameter für das Objekt Optionsfeld:

- **Beschriftung**: Legen Sie durch Eingabe von Text in das Textfeld eine Beschriftung fest. Die Beschriftung kann Tags enthalten, wenn diese in geschweifte Klammern gesetzt werden (z.B. `{tagname}`).
- **Schriftarten**: Legen Sie durch Klicken auf die Schaltfläche Schriftart eine Schriftart für die Beschriftung fest.
- **E-Sign**: Wenn diese Option angewählt ist, wird der Benutzer vor dem Ausführen des Befehls zur Eingabe der elektronischen Signatur aufgefordert.
- Kontrollkästchen **Bestätigung erforderlich**: Dieses Kästchen anklicken (aktivieren), um sicherzustellen, dass "WinStudio" zur Bestätigung der Aktion während der Laufzeit auffordert.
- Auswahlliste **Taste**: Wählen Sie eine Taste aus der Liste, um ein Tastaturkürzel mit dem Objekt oder der Objektgruppe zu verknüpfen. Dann kann dieses Tastaturkürzel zum Aktivieren/Deaktivieren des Optionsfelds gedrückt werden.
  - Klicken (Aktivieren) Sie das Feld **Umschalt**, **Strg** und/oder **Alt** zum Erstellen einer Tastaturkombination, d.h. die Taste <Umschalt>, <Strg> oder <Alt> muss zusammen mit der in der Auswahlliste festgelegten Taste gedrückt werden.
  - Öffnen Sie den Dialog "Tasten ändern" durch Klicken auf die Such-Schaltfläche **...**, um Ihre Tastaturkombinationen zu verändern. Für Tastaturkombinationen kann die Position der Taste Umschalt, Strg oder Alt auf der Tastatur auf Seite 80 mit **Links**, **Rechts** oder **Links oder rechts** festgelegt werden. Bei Auswahl von **Links oder rechts** wird der Befehl jedes Mal ausgeführt, wenn eine dieser Tasten in Verbindung mit der in der Auswahlliste festgelegten Taste gedrückt wird.
- Feld **Deaktivieren**: Geben Sie in diesem Feld einen Tag oder einen Ausdruck zum Freigeben und Deaktivieren des Objekts ein. Bei Eingabe eines Wertes ungleich Null wird das Objekt Optionsfeld deaktiviert.
- **Sicherheit** Geben Sie in diesem Feld einen Wert zur Festlegung einer Sicherheitsstufe für das Objekt ein, wie in den Sicherheitseinstellungen definiert. Wenn ein Benutzer sich anmeldet und nicht über die angegebene Sicherheitsstufe verfügt, deaktiviert "WinStudio" das Objekt.
- Feld **Variable**: Wenn der Benutzer während der Laufzeit auf das Optionsfeld klickt, wird der Wert des hier eingetragenen Tags aktualisiert. Wenn keine **Rückmeldung** festgelegt wurde, wird der Wert dieses Tags außerdem zur Anzeige des aktuellen Zustands des Objekts verwendet.
- **Wert Wahr**: Legen Sie einen Wert fest zur Änderung des Steuerungszustands auf WAHR und zur Anzeige, dass das Steuerelement sich im Zustand WAHR befindet. Weitere Informationen über Zustände entnehmen Sie bitte der Zustandstabelle.
- **Erweitert**: Klicken Sie auf diese Schaltfläche zum Öffnen des Dialogs für weitere Einstellungen:

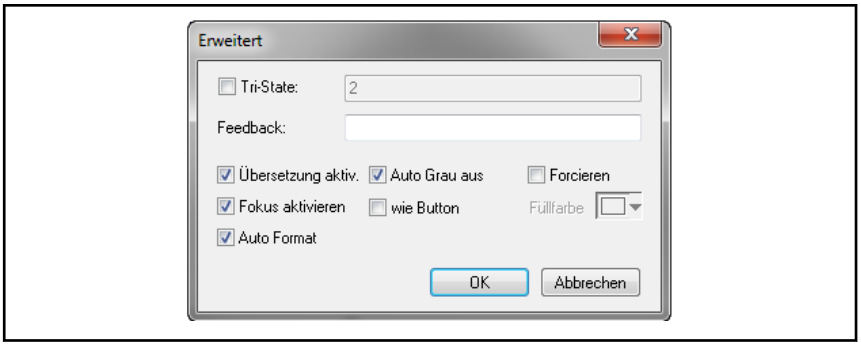


Abb.8-33: Dialog Weitere Einstellungen

– **Dritt-Zustand**: Bei Aktivierung hat das Steuerelement einen dritten Zustand. Der dritte Zustand wird angezeigt, wenn der im Feld **Feedback** konfigurierte Tag den im Feld **Dritt-Zustand** (Auswahlelement gegraut) festgelegten Wert annimmt. Wenn das Feld **Rückmeldung** leer bleibt, wird der dritte Zustand angezeigt, wenn der im Feld **Tag** konfigurierte Tag den im Feld **Dritt-Zustand** festgelegten Wert annimmt.

HINWEIS	**Das Feld Dritt-Zustand darf nicht mit dem gleichen Wert wie das Feld Wert Wahr konfiguriert sein. Es darf auch nicht mit einer leeren Zeichenkette belegt sein.**

– **Rückmeldung**: Wert, der den Zustand des Objekts anzeigt (FALSCH, WAHR oder DRITT-ZUSTAND). Wenn der in **Rückmeldung** konfigurierte Tag gleich dem in **Wert Wahr** konfigurierten Tag ist, dann wird der Zustand auf WAHR gesetzt.

  Wenn der in **Rückmeldung** konfigurierte Tag gleich dem in **Dritt-Zustand** konfigurierten Tag ist, dann wird der Zustand auf DRITT-ZUSTAND gesetzt.

  Wenn keine dieser Bedingungen erfüllt ist, wird der Zustand auf FALSCH gesetzt.

  Wenn das Feld **Rückmeldung** leer bleibt, wird der im Feld **Tag** konfigurierte Tag als **Rückmeldungs**-Tag verwendet.

– **Übersetzung aktiv**: Durch Anklicken dieser Option, wird der Text in die Übersetzungsdatei aufgenommen.

– **Automatisch ausblenden**: Ändert die Beschriftung des Objekts in grau, wenn es durch das Feld **Deaktivieren** oder durch das Sicherheitssystem deaktiviert ist.

– **Forcieren**: Dieses Kontrollkästchen anklicken (aktivieren), um das Erkennen einer Tag-Änderung durch die Tag-Datenbank zu erzwingen, wenn der Benutzer auf das Objekt klickt, auch wenn der Wert des entsprechenden Tags sich nicht ändert.

– **Fokus aktivieren**: Wenn diese Option angewählt ist, kann das Objekt während der Laufzeit durch die Navigationstasten aktiviert werden.

– **wie Button**: Wenn diese Option angewählt ist, wird das Steuerelement nicht als Standard-Optionsfeld, sondern als Schaltfläche dargestellt.

- **Füllfarbe**: Festlegen der Füllfarbe für das Optionsfeld. Diese Option ist nur aktiv, wenn die Option **wie Button** angewählt ist.
- **Auto-Format**: Wenn diese Option aktiviert ist und die Beschriftung einen Dezimalwert in geschweiften Klammern (z.B. **{1,2345}**) oder einen Tag vom Typ Real auf Seite 124 enthält (siehe **Beschriftung** oben), dann wird der Wert gemäß der von der Funktion SetDecimalPoints auf Seite SetDecimalPoints erstellten virtuelle Tabelle formatiert.

Betriebsmodi

Das Objekt Optionsfeld hat zwei verschiedene Betriebsmodi: Normal und Dritt-Zustand.

Weitere Informationen unter Betriebsmodi für Objekte Kontrollkästchen und Optionsfeld auf Seite 204.

Betriebsmodi für die Objekte Kontrollkästchen und Optionsfeld

Die Objekte Kontrollkästchen und Optionsfeld haben jeweils zwei Betriebsmodi: "Normal" und "Dritt-Zustand".

Diese Betriebsmodi funktionieren im wesentlichen für beide Objekte gleich.

Normalmodus

Im Normalmodus hat das Objekt zwei mögliche Zustände: FALSCH (nicht ausgewählt) und WAHR (ausgewählt).

Aussehen von Objekten im Normalmodus

Objektzustand	Kontrollkästchen		Optionsfeld	
	Standard	Schaltflächen-Stil	Standard	Schaltflächen-Stil
FALSCH	☐ Check	Check	○ Radio	Radio
WAHR	☑ Check	Check	◉ Radio	Radio

Wenn der Benutzer das Objekt während der Laufzeit anklickt/auswählt – d.h. wenn der Objektzustand auf WAHR gesetzt ist – wird dem in den Objekteigenschaften konfigurierte Tag der im Feld **Wert wahr** konfigurierte Wert zugewiesen.

Wenn der Benutzer das Objekt abwählt oder die Markierung aufhebt – d.h. wenn der Objektstatus auf FALSCH zurückgesetzt wird – wird der im Feld **Tag** konfigurierten Tags dem im Feld **Wert wahr** konfigurierten Wert NICHT gleichgesetzt. In den meisten Fällen bedeutet dies, dass er auf "0" oder eine leere Zeichenkette gesetzt wird. Es gibt jedoch folgende Ausnahmen:

- Wenn das Feld **Wert wahr** gleich "0" konfiguriert ist, entspricht "NICHT WAHR" dem Wert "1".
- Wenn das Feld **Wert wahr** komplett leer gelassen wird, entspricht "NICHT WAHR" der Zeichenkette "UNSELECTED".

Modus Dritt-Zustand

Der Modus Dritt-Zustand wird durch Auswählen der Option **Dritt-Zustand** im Dialog "Objekteigenschaften"des Objekts aktiviert. Dies fügt dem Objekt einen dritten möglichen Zustand hinzu.

Aussehen von Objekten im Modus Dritt-Zustand

Objektzustand	Kontrollkästchen		Optionsfeld	
	Standard	Schaltflächen-Stil	Standard	Schaltflächen-Stil
FALSCH	☐ Check	Check	○ Radio	Radio
WAHR	☑ Check	Check	◉ Radio	Radio
DRITT-ZUSTAND	☑ Check	Check	◉ Radio	Radio

Erstmaliges Anklicken/Auswählen des Objekts funktioniert genau wie im Normalmodus.

Wenn der Benutzer das Objekt während der Laufzeit zum zweiten Mal anklickt/auswählt, wird der Objektzustand von WAHR auf DRITT-ZUSTAND gesetzt und der Wert des im Feld **Variable** in den Objekteigenschaften konfigurierten Tags wird dem im Feld **Dritt-Zustand** konfigurierten Wert gleichgesetzt.

Wenn der Benutzer das Objekt zum dritten Mal angeklickt/auswählt, hängt die Änderung des Objekts vom Objekttyp ab:

- Das Objekt Kontrollkästchen wird von DRITT-ZUSTAND auf FALSCH zurückgesetzt.

  Der Tag im Feld "Variable" erhält dann automatisch den kleinsten positiven Integer-Wert, der weder im Zustand WAHR noch DRITT-ZUSTAND konfiguriert ist.

  – **Beispiel1:**

    Konfigurierter Wert für Zustand WAHR: "Wert WAHR" =1

    Konfigurierter Wert für dritten Zustand: "Dritt Zustand" = 2

    Tagwert im Feld "Variable" für Zustand FALSCH = 0 (automatisch zur Laufzeit gesetzt)

  – **Beispiel2:**

    Konfigurierter Wert für Zustand WAHR: "Wert WAHR" = 0

    Konfigurierter Wert für Dritt-Zustand: "Dritt Zustand"" = 1

    Tagwert in Feld "Variable" für Zustand FALSCH = 2 (automatisch zur Laufzeit gesetzt)

- Das Objekt Optionsfeld wird von DRITT-ZUSTAND auf FALSCH zurückgesetzt und der Wert des im Feld **Variable** konfigurierten Tags wird dem im Feld **Wert wahr** konfigurierten Wert gleichgesetzt. Wenn der Benutzer das Objekt weiter anklickt, wechselt das Objekt einfach zwischen WAHR und DRITT-ZUSTAND.

  Zum tatsächlichen Löschen des Objekts – d.h. zum Zurücksetzen des Objekts auf FALSCH – muss der Benutzer ein anderes Optionsfeld in der gleichen Gruppe anklicken/auswählen.

Diese Funktionalität wird in der folgenden Tabelle zusammengefasst:

Bilder und Grafiken

Aktueller Zustand des Objekts	Veränderung des Objekts beim Anklicken	
	Kontrollkästchen	Optionsfeld
FALSCH	Objektzustand wird auf WAHR gesetzt und Wert wird gleich **Wert wahr** gesetzt	
WAHR	Objektzustand wird auf DRITT-ZUSTAND gesetzt und Wert wird gleich **Dritt-Zustand** gesetzt	
DRITT-ZUSTAND	Objektzustand wird auf FALSCH gesetzt, und der Tag im Feld "Variable" erhält dann automatisch den kleinsten positiven Integer-Wert, der weder im Zustand WAHR (Wert wahr) noch DRITT-ZUSTAND konfiguriert ist.	Objektzustand wird auf WAHR gesetzt und Wert wird gleich **Wert wahr** gesetzt

Objekt Combobox

Klicken Sie im Werkzeugfenster **WinStudio Screen Tools** auf **Combobox** zum Einfügen einer Auswahlliste mit vordefinierten Listeneinträgen.

Wenn in der Laufzeitumgebung die Anzahl der Listeneinträge größer ist als der zur Verfügung stehende Platz (Listenlänge), wird für die Liste ein Rollbalken aktiviert. Wenn während der Laufzeit eine Bezeichnung aus der Liste gewählt wird, minimiert sich die Combobox und die ausgewählte Bezeichnung wird in der Combobox angezeigt.

Mit einem Doppelklick auf das Objekt Combobox öffnet sich der Dialog "Objekteigenschaften".



Abb.8-34: Objekteigenschaften: Combobox

Dieser Dialog kann zur Festlegung der folgenden Parameter verwendet werden:

- **Marke**: Bei Eingabe eines String-Tags wird der aktuell in der Combobox angezeigte Listeneintrag in den String-Tag übernommen.
- **Position**: Geben Sie einen Tag vom Typ Integer ein, in dem der Index der aktuell in der Combobox ausgewählten Listenposition gespeichert wird. Eine Änderung dieses Tag-Werts ändert den Listeneintrag in der Anzeige.
- **Deaktivieren**: Geben Sie zum Deaktivieren dieser Combobox einen Tag mit Wert ungleich Null ein. Zum Freigeben der Befehlsanimation geben Sie Null ein oder lassen das Feld frei (Standard). Wenn die Combobox deaktiviert wird, erscheint sie während der Laufzeit grau hinterlegt.
- **Datenquellen**: Klicken Sie zum Öffnen des Dialogs "Datenquellen" (siehe unten).
- **Erweitert**: Klicken Sie zum Öffnen des Dialogs "Combobox – Weitere Einstellungen":

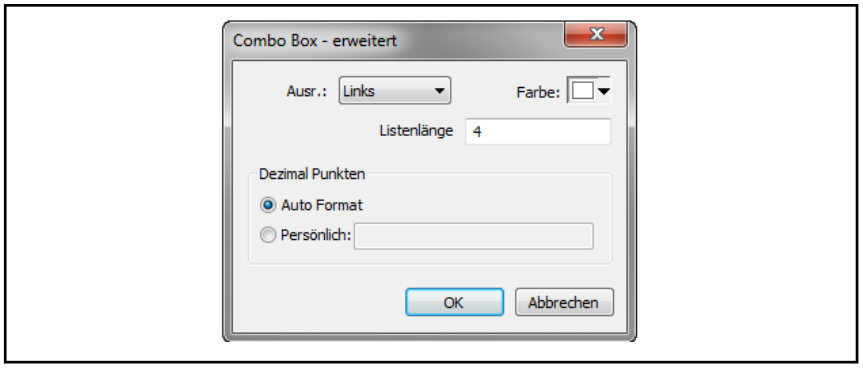


Abb.8-35: Combobox – Weitere Einstellungen

- **Ausrichten**: Wählen Sie die Ausrichtung der Listeneinträge (**Links**, **Mitte** oder **Rechts**). Dies beeinflusst die Ausrichtung sowohl in der Combobox als auch in deren Liste.
- **Farbe**: Klicken zum Festlegen einer Hintergrundfarbe für die Combobox.
- **Listenlänge**: Geben Sie eine ganze Zahl (oder einen Tag vom Typ Integer auf Seite 124) ein zum Festlegen der Anzahl der Elemente, die beim Anklicken der Combobox auf einmal angezeigt werden. Je größer die Zahl der Elemente, desto länger wird die Auswahlliste.

☞ Wenn diese Zahl kleiner ist als die "Gesamt"zahl der Elemente in der Liste, kann in der Liste geblättert werden.

  - "Dezimalzeichen": Wählen Sie, wie Dezimalwerte im Bild dargestellt werden:
    - **Auto-Format**: Dezimalwerte werden gemäß der von der Funktion SetDecimalPoints auf Seite SetDecimalPoints erstellten virtuellen Tabelle formatiert.
    - **Benutzerdefiniert**: Geben Sie die Anzahl der anzuzeigenden Dezimalstellen (z.B. 2) für alle Dezimalwerte ein.

- **Schrift**: Klicken Sie zum Öffnen eines Schriftendialogs Schriftarten auf Seite 269. Verwenden Sie diesen Dialog zum Ändern der Schrift für Listeneinträge.
- **Sicherheit**: Geben Sie eine Sicherheitsstufe für den Befehl ein (0 bis 255). Wenn sich ein Bediener anmeldet und nicht über die angegebene Sicherheitsstufe verfügt, wird der Befehl deaktiviert. Wenn sich ein Bediener anmeldet und über die angegebene Sicherheitsstufe verfügt oder wenn dieses Feld leer bleibt, bleibt das Steuerelement aktiv.
- **Bestätigung erforderlich**: Anklicken (aktivieren), um während der Laufzeit einen Bediener zum Bestätigen eines Befehls aufzufordern.
- **Übersetzung aktiv**: Durch Anklicken (Freigeben) wird die automatische Übersetzung der Combobox mit Hilfe des Übersetzungswerkzeugs auf Seite 437 freigegeben.
- **Sortieren**: Durch Anklicken (Aktivieren) wird der Inhalt der Liste in alphabetischer Reihenfolge angezeigt. Dieser Parameter ist nur verfügbar, wenn im Bereich "Datenquellen" der Typ **Array-Variable** ausgewählt ist.
- **E-Sign**: Wenn diese Option angewählt ist, wird der Benutzer vor dem Aktivieren der Listenauswahl zur Eingabe der elektronischen Signatur aufgefordert.

Bilder und Grafiken

- **Eingabe aktiv**: Wenn Sie diese Option aktivieren, können Sie zur Laufzeit nicht nur vorhandene Listeneinträge auswählen, sondern auch beliebige Werte eingeben.
- **Virtuelle Tastatur**: Für dieses Objekt wird bei der Eingabe von Werten eine virtuelle Tastatur verwendet. Vor dem Konfigurieren der Virtuellen Tastatur für dieses Objekt muss die Option "Virtuelle Tastatur" in den Projekteinstellungen im Register **Viewer** aktiviert sein.

Datenquellen

Verwenden Sie den Dialog "Datenquellen" zum Konfigurieren der Elemente/Bezeichnungen, die im Objekt Combobox angezeigt werden.

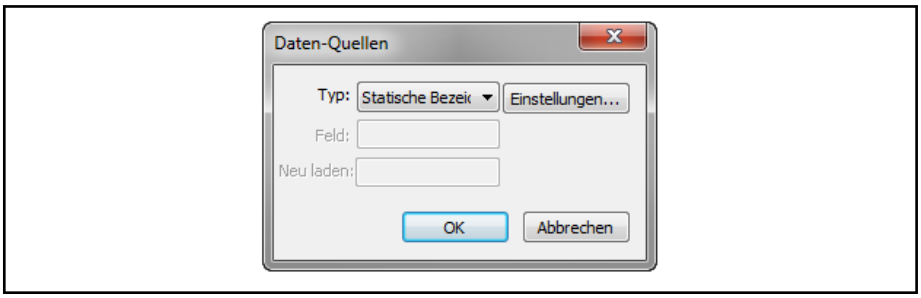


Abb.8-36: Dialog Datenquellen

- **Typ**: Wählen Sie den Typ der zu verwendenden Datenquelle aus und klicken Sie dann zum Konfigurieren der Quelle auf die Schaltfläche **Einstellungen...**. Jeder Quelltyp wird weiter unten genauer beschrieben.
- **Feld** (nur für Textdatei und Datenbank): Festlegen, aus welchem Feld bzw. welcher Spalte der Datenquelle gelesen werden soll.
- **Neu laden** (nur für Textdatei und Datenbank): Geben Sie einen Tag-Namen ein. Wenn sich der Wert des angegebenen Tags ändert, lädt die Combobox die Bezeichnungen erneut aus der Datenquelle.

Typ: Statische Bezeichnungen

Wenn **Typ** auf **Statische Bezeichnungen** gesetzt ist, können die folgenden "Einstellungen" konfiguriert werden:

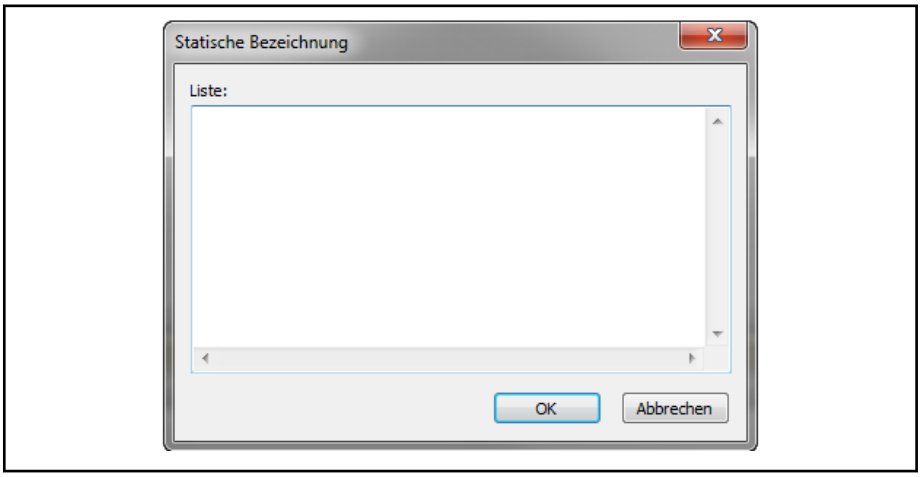


Abb.8-37: Dialog Statische Bezeichnungen

Geben Sie Ihre Listeneinträge – eine Bezeichnung pro Zeile – wie beim Bearbeiten einer einfachen Textdatei ein. Sie können als Listeneinträge auch Tags oder Ausdrücke verwenden, indem Sie diese in geschweifte Klammern setzen.

Zum Beispiel:

```
Tank#1 in {ArrayNameTag}.
```

Die Listeneinträge werden nicht sortiert und müssen in die Reihenfolge gebracht werden, in der sie während der Laufzeit angezeigt werden sollen. Die erste Zeile ist Position 0, die zweite Zeile ist Position 1 usw.

Typ: Array-Variable

Wenn **Typ** auf **Array-Variable** gesetzt ist, können die folgenden Einstellungen konfiguriert werden:

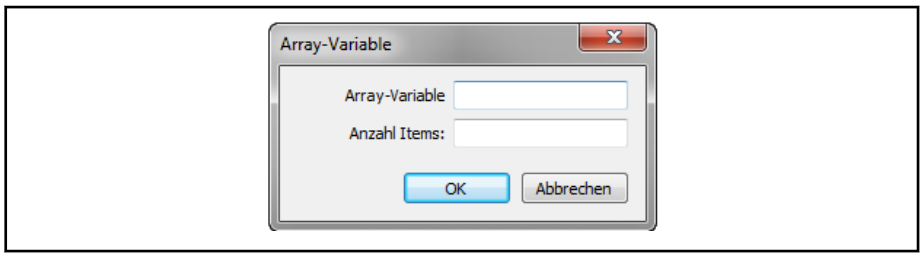


Abb.8-38: Dialog Array-Variable

- **Array-Variable**: Geben Sie den Namen eines Array-Variable auf Seite 125 vom Typ String auf Seite 124 ein, der die Elemente für die Combobox enthält.
- **Anzahl der Elemente**: Legen Sie fest, wie viele Array-Elemente in der Combobox angezeigt werden sollen. Die Combobox zählt "Array-Index 0" als das erste Element. Wenn also der Wert 4 eingegeben wird, zeigt die Combobox die Array-Elemente 0-4 an.

Typ: Textdatei

Wenn **Typ** auf **Textdatei** gesetzt ist, können die folgenden "Einstellungen" konfiguriert werden:

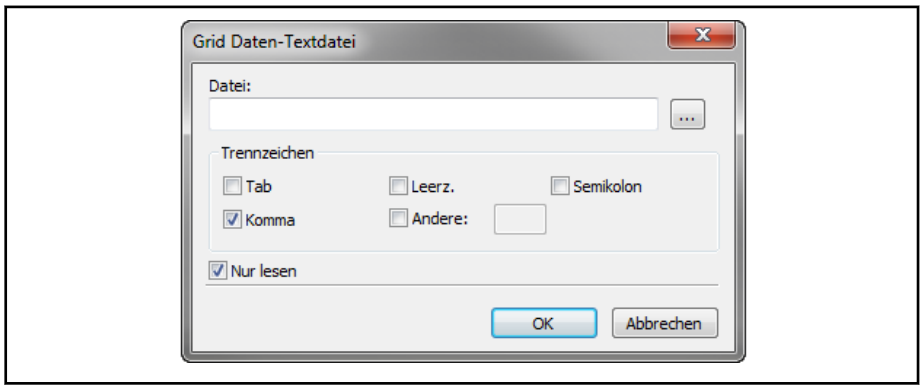


Abb.8-39: Rasterdaten – Dialog Textdatei

- **Datei**: Name der Quelltextdatei. Sie können entweder den Dateinamen und dessen Pfad eingeben oder zum Suchen auf die Schaltfläche "..." klicken. (Wenn die Datei in Ihrem Projektordner gespeichert ist, kann der Pfad im Namen entfallen.) Im Feld **Datei** können Tag-Namen in geschweiften Klammern eingegeben werden (z.B. **{TagName}**).
- **Trennzeichen**: Legen Sie den/die in der Daten Quelldatei verwendeten Feldbegrenzer fest. Wenn zum Beispiel die Daten aus einer CSV-Datei (comma separated values) gelesen werden, ist die Option **Komma** auszuwählen. Durch Auswählen der Option **Andere** und Eingabe des benutzerdefinierten Feldbegrenzers im Feld rechts daneben, kann sogar ein benutzerdefiniertes Trennzeichen ausgewählt werden.

Typ: Datenbank

Wenn **Typ** auf **Datenbank** gesetzt ist, können die folgenden "Einstellungen" konfiguriert werden:

Bilder und Grafiken

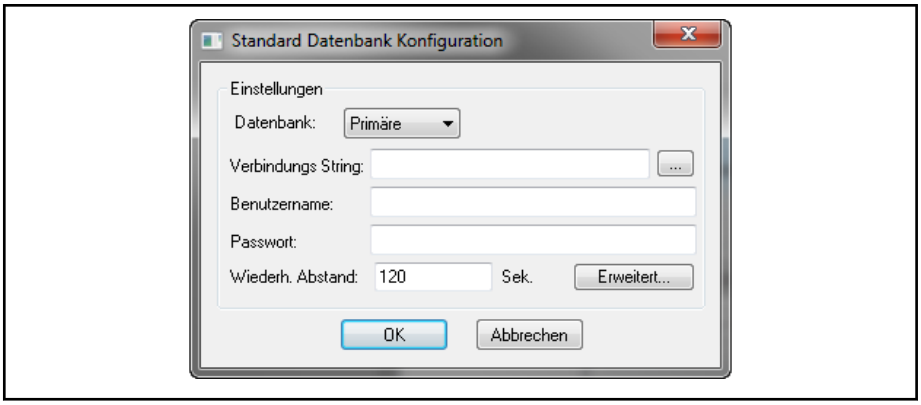


Abb.8-40: Dialog Datenbankkonfiguration

Weitere Informationen zu diesem Dialog finden Sie unter Dialog Datenbankkonfiguration auf Seite 502.

Objekt Listenfeld

Zum Einfügen eines Listenfelds klicken Sie im Werkzeugfenster **WinStudio Screen Tools** auf **Listenfeld**: Generell zeigt das "aktive" Listenfeldobjekt beim Ausführen eines Projekts eine Liste aller Meldungen an.

Wenn ein Bild nur ein Listenfeldobjekt und keine Texteingabefelder enthält, wird das Listenfeldobjekt automatisch aktiviert.

Wenn ein Bild mehrere Listenfeldobjekte und Texteingabefelder enthält, kann ein Listenfeldobjekt mit dem Cursor (Zeigegerät) oder der Tabulatortaste ausgewählt und aktiviert werden.

Sie können während der Laufzeit eine Meldung aus dem aktiven Listenfeld auswählen und den Wert der Meldung in einen Tag schreiben. (Wenn eine Liste zu lang für den sichtbaren Bereich eines Listenfeldobjekts ist, enthält das Objekt Rollbalken.)

Verwenden Sie die Option **Enter zur Auswahl erforderlich** im Dialog "Objekteigenschaften" zum Konfigurieren des Eingabeverhaltens der Liste:

- Aktivieren Sie die Option **Enter zur Auswahl erforderlich** und verwenden Sie zum Blättern durch die Meldungsliste die Tasten von der (virtuellen-) Tastatur, Listensteuerobjekte aus der **Bibliothek**, Zeigegeräte oder benutzerdefinierte Tasten mit der Funktion PostKeys() auf Seite 633. Verwenden Sie dann die Eingabetaste, um die Meldung auszuwählen und ihren Wert in den Schreib-Tag zu übernehmen. Mit Hilfe der Tasten ESC und Tab können Sie jederzeit die Eingabe abbrechen und zu der zuvor ausgewählten Meldung zurückkehren.
- Deaktivieren Sie die Option **Enter zur Auswahl erforderlich**, um den Wert einer ausgewählten (markierten) Meldung automatisch in den Schreib-Tag zu übernehmen.

Hinzufügen eines Listenfeldobjekts zu einem WinStudio-Bild:

1. Zum Einfügen eines Listenfelds klicken Sie im Werkzeugfenster **WinStudio Screen Tools** auf **Listenfeld**
2. Klicken Sie in das Bild und ziehen Sie, um ein Rechteck zu erstellen bzw. zu ändern. Die Höhe des Rechtecks und die Schriftgröße der dargestellten Texte bestimmen, wie viele Meldungen sichtbar sind. Die Breite des Listenfeldes bestimmt den sichtbaren Text je Zeile. Nach dem Erstellen des Listenfeldes können die Größe und der Schriftstil angepasst werden, um mehr Meldungen im verfügbaren Bereich anzuzeigen.

3. Mit einem Doppelklick auf das Objekt öffnet sich der Dialog "Objekteigenschaften".

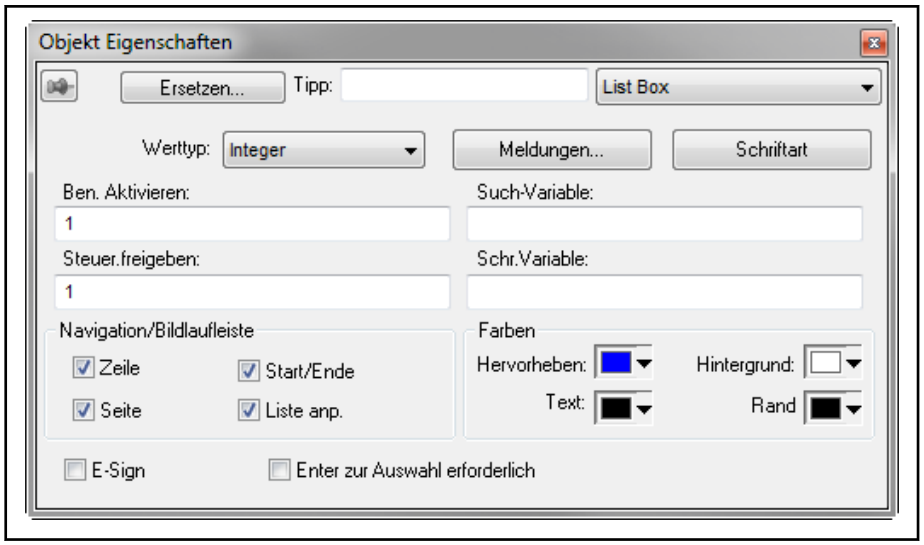


Abb.8-41: Objekteigenschaften: Listenfeld

Der Dialog "Objekteigenschaften" kann auch durch einen Rechtsklick auf das Listenfeldobjekt oder durch Markieren des Objekts, Drücken der Tasten Alt+Eingabe und Auswählen von **Eigenschaften** aus dem entsprechenden Kontextmenü geöffnet werden.

Dieser Dialog wird zur Festlegung der folgenden Parameter verwendet:

- Auswahlliste **Werttyp**: Wählen Sie einen der folgenden zum Indizieren der Meldungsliste verwendeten Variablentypen.
  - **Boolean**
  - **Integer** (Standard)
  - **LSB** (Least Significant Bit)

Weitere Informationen finden Sie in der Erläuterung der Spalte **Status** im "Dialog Meldungskonfiguration" auf Seite 213.

- **Meldungen**: Anklicken zum Öffnen von "Dialog Meldungskonfiguration" auf Seite 213.
- **Benutzereingabe aktivieren**: Geben Sie einen Tag, einen Ausdruck oder eine Zahl (ungleich null) ein, um eine Meldungsauswahl im Laufzeitsystem durch den Maschinenbediener zu erlauben. Der Standardwert ist "1" ("wahr" oder "freigegeben").
- **Steuerung freigeben**: Geben Sie einen Tag, einen Ausdruck oder eine Zahl ("ungleich null") ein, um die Funktion "Textauswahl" in der Laufzeitumgebung zu aktivieren.

  Der Standardwert ist "1" ("wahr" oder "freigegeben"). Die Textauswahl erfolgt über einen Tag im Feld "Such-Variable".

  WinStudio verwendet dabei die fortlaufenden Nummern in der Spalte "Wert" im "Dialog Meldungskonfiguration" auf Seite 213 zur Adressierung der Texte. Dabei repräsentiert jede Nummer in der Spalte "Wert" genau einem Text in der Spalte "Meldungen". Wenn die Steuerungsfunktion aktiviert ist, können die Texte ohne Benutzereingabe direkt aus dem Prozess angewählt werden.

Bilder und Grafiken

- **Such-Variable**: Geben Sie einen Tag vom Typ Integer oder Boolean ein, der auf eine ausgewählte Meldung zeigt, basierend auf dem Feld Meldungs-**Wert** (im "Dialog Meldungskonfiguration" auf Seite 213). Der Inhalt der Variablen referenziert über die Spalte "Wert" einen Text in der Spalte "Meldungen".

  Mit den Feldern **Steuerung freigeben** und **Benutzer aktivieren** kann gesteuert werden, ob dieser Tag vom Bediener oder von einem Prozess geändert werden kann.
- **Schreib-Variable** ("optional"): Geben Sie einen String-Tag ein, um den Wert der zuletzt ausgewählten Meldung zu erhalten. Wenn ein Bild mit einem Listenfeldobjekt geschlossen und wieder geöffnet wird, verwendet "WinStudio" diesen Tag-Wert zur Bestimmung der letzten in diesem Listenfeld ausgewählten Meldung.
- **E-Sign**: Wenn diese Option angewählt ist, wird der Benutzer bei der Verwendung dieses Objekts zur Eingabe der elektronischen Signatur aufgefordert.
- **Zeile**: Durch Anklicken (Aktivieren) werden Pfeile für Zeile auf und Zeile ab in den Rollbalken des Listenfeldobjekts integriert.
- **Seite**: Durch Anklicken (Aktivieren) werden Pfeile für Seite auf und Seite ab in den Rollbalken des Listenfeldobjekts integriert.
- **Start/Ende**: Durch Anklicken (Aktivieren) werden Pfeile für Anfang und Ende in den Rollbalken des Listenfeldobjekts integriert.
- **Liste anpassen**: Durch Anklicken (Aktivieren) kann die Meldungsliste endlos durchgeblättert werden.
- **Enter zur Auswahl erforderlich** : Durch Anklicken (Aktivieren) dieses Kontrollkästchens können Meldungen nur mit der Eingabetaste ausgewählt werden. Dies verhindert das Auswählen von Meldungen mit der Tab-Taste.
- Felder **Farbe**: Klicken Sie zum Öffnen des Dialogs "Farbe" oder des 16-Farben-Dialogs "Farbauswahl" auf ein Farbkästchen. In beiden Dialogen können Farben für das Listenfeldobjekt festgelegt oder geändert werden.
  - **Hervorheben**: Festlegen einer Farbe um eine Meldung hervorzuheben (Standard: blau).
  - **Text**: Festlegen einer Farbe für markierten Text (Standard: schwarz).
  - **Hintergrund**: Festlegen einer Farbe für den Hintergrund des Listenfelds (Standard: weiß).
  - **Rand**: Festlegen einer Farbe für den Rahmen des Listenfelds (Standard: schwarz).

Dialog Meldungskonfiguration

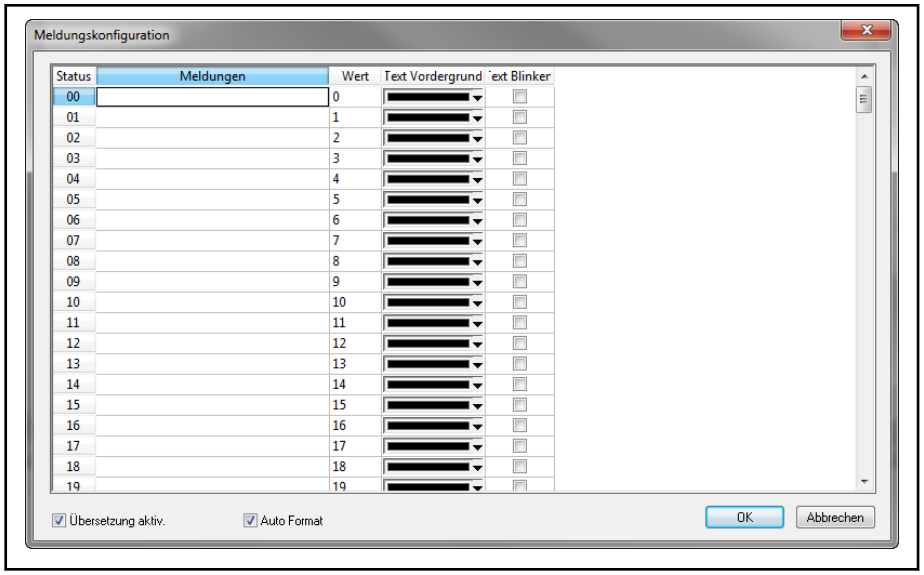


Abb.8-42: Dialog Meldungskonfiguration

Die Parameter auf dieser Registerkarte werden wie folgt verwendet:

- Spalte **Status** ("schreibgeschützt"): Dieses Feld dient der Anzeige der einzelnen indizierten Meldungen

  "WinStudio" nummeriert diese Spalte automatisch, je nachdem welche Einstellung in der Auswahlliste "WertTyp" ausgewählt ist:

  – **Boolean**: Liefert zwei gültige Zustände mit Bezeichnung 0 und 1
  – **Integer**: Liefert 255 gültige Zustände mit Bezeichnung 1 bis 255
  – **LSB**: Liefert 32 gültige Zustände (32 Bits in einem Integer-Wert) mit Bezeichnung 0 bis 31

- Spalte **Meldung**: Geben Sie die Zeichenkette ein, die im Listenfeldobjekt angezeigt werden soll. Die Meldung kann Tags enthalten, wenn diese in geschweifte Klammern gesetzt werden (z.B. "{"tagname"}").
- Spalte **Wert**: Jeder hier eingetragene Wert korrespondiert mit genau einem Text in der Spalte "Meldungen". Dadurch ist es möglich über jeden Wert einen Text zu adressieren. Wenn während der Laufzeitumgebung ein passender Wert in das Tag geschrieben wird, welches im Feld "Such Variable" parametriert ist, wird der dazugehörige Text automatisch ausgewählt und in der Anzeige des "Listen Objektes" hervorgehoben. Parallel dazu wird der Wert in das Tag geschrieben, welches im Feld "Schreib Variable" parametriert ist. Wenn im Feld "Wert Typ" LSB angegeben wurde, verwendet "WinStudio" die in der Spalte "Status" festgelegten Werte.
- **Text Vordergrund**: Anklicken zum Festlegen einer Farbe für den Vordergrund des Meldungstexts. Die Farbe wird nur angezeigt, wenn die Meldung nicht markiert ist.
- **Text blinkend**: Anklicken (aktivieren), damit eine markierte Meldung einmal pro Sekunde blinkt.
- **Schriftarten**: Anklicken zum Öffnen des Dialogs "Schriftart", in dem die Merkmale der Schriftart für die Meldungen (Stil, Größe usw.) geändert werden können.
- **Übersetzung aktivieren**: Durch Anklicken dieser Option wird der Text in die Übersetzungsdatei aufgenommen.

- **Auto-Format**: Wenn diese Option aktiviert ist und die Meldung einen Dezimalwert in geschweiften Klammern (z.B. `{1,2345}`) oder einen Tag vom Typ Real auf Seite 124 enthält (siehe **Meldung** oben), dann wird der Wert gemäß der von der Funktion SetDecimalPoints auf Seite SetDecimalPoints erstellten virtuelle Tabelle formatiert.

Objekt Smart Message

Klicken Sie im Werkzeugfenster **WinStudio Screen Tools** auf **Smart Message** zum Erstellen eines oder mehrerer intelligenter Meldungsobjekte, die beim Ausführen Ihres Projekts zum Anzeigen von Meldungen und Grafiken verwendet werden können. "WinStudio" stellt die folgenden "Smart Message - Objekttypen" zur Verfügung:

- **Meldungsanzeige**: Ermöglicht die Anzeige einer aus mehreren Meldungen innerhalb eines einzelnen Meldungsobjekts.
- **Multistatus Indikator**: Ermöglicht die Anzeige einer aus mehreren Meldungen innerhalb eines einzigen Meldungsobjekts wobei zusätzlich Bitmap-Grafiken zusammen mit den Meldungen angezeigt werden können.
- **Multistatus Schalter**: Ermöglicht die Anzeige von Meldungen und Bitmap-Grafiken. Dieses Objekt gleicht außerdem einem Multipositionsschalter, da hiermit inkrementell durch die Meldungen geschaltet werden kann, indem das Objekt während der Laufzeit angeklickt wird.

Diese "Smart Message - Objekttypen" unterscheiden sich aufgrund ihres unterschiedlichen Funktionsumfangs in der Konfiguration. Einen Parameter haben sie jedoch gemeinsam: Die Variable im Feld "Tag-Wert lesen" für die Meldungsauswahl.

Hinzufügen eines "Smart Message - Objektes" zu einem WinStudio-Bild:

1. Klicken Sie auf **Smart Message** und positionieren Sie die Maus im Bild-Editor.
2. Erstellen Sie ein Rechteck durch Klicken und Ziehen. Die Größe des Rechtecks und die Schriftgröße bestimmen, wieviel Text und in welcher Größe die Bitmap-Grafik angezeigt werden können. Später können die Größe des Rechtecks und die Schriftart angepasst werden, um das Anzeigen längerer Meldungen zu ermöglichen.
3. Mit einem Doppelklick auf das Objekt öffnet sich der Dialog "Objekteigenschaften".

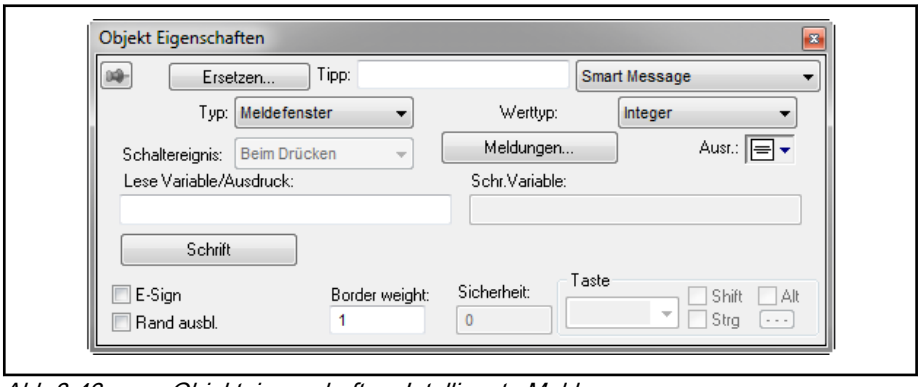


Abb.8-43: Objekteigenschaften: Intelligente Meldung

Dieser Dialog wird zur Festlegung der folgenden Parameter verwendet:

- **Typ**: Anklicken zur Auswahl des Objekttyps. Der Objekttyp bestimmt das Verhalten des Objekts während der Laufzeit und die von ihm unterstützten Funktionen:
  - **Meldungsanzeige** (Standard)

  - **Multistatus Indikator**
  - **Multistatus Schalter**
- Auswahlliste **Werttyp**: Wählen Sie die zum Indizieren der Meldungsliste verwendeten Werttypen aus:
  - **Boolean**: Liefert zwei gültige Zustände. Verwenden Sie diese Auswahl, wenn sie eine von zwei verschiedenen Meldungen auf Grundlage eines Booleschen Wertes (0 oder 1) anzeigen wollen.
  - **Integer** (Standard): Liefert 500 gültige Zustände. Verwenden Sie diese Auswahl, wenn sie verschiedene Meldungen auf Grundlage numerischer Werte aus einem Integer-Tag anzeigen wollen.
  - **LSB** (least significant bit): Liefert 32 gültige Zustände (32 Bits in einem Integer-Wert). Verwenden Sie diese Auswahl, wenn sie verschiedene Meldungen anzeigen wollen auf der Grundlage, welches Bit aus einem Integer-Tag gesetzt wurde. Wenn mehr als ein Bit aus dem Integer-Tag gleichzeitig gesetzt sind, wird die Meldung angezeigt, die mit dem niedrigstwertigen Bit verbunden ist.

☞ Wenn **Multistatus Schalter** als Typ für ein **Smart Message**-Objekt ausgewählt ist, können nur 16 verschiedene Meldungen mit dem Objekt verbunden werden, auch für **Integer-** oder **LSB-**Werte.

- **Tag/Ausdr. lesen**: Geben Sie einen Tag oder einen Ausdruck ein. Der Wert bestimmt, welche Meldung von dem Objekt während der Laufzeit angezeigt wird.
- **Tag schreiben** (optional und nur für **Multistatus Schalter** verfügbar): Geben Sie den Namen eines Projekt-Tags ein. Der Wert, der mit der aktuell angezeigten Meldung verbunden ist, wird in diesen Tag geschrieben.
- **Ausrichten**: Wählen Sie die Ausrichtung für den im "Smart Message - Objekt" angezeigten Text aus.
- Bereich **Taste** (Tastaturkürzel sind nur für **Multistatus Schalter** verfügbar): Tastaturkürzel zum Auswählen der nächsten Meldung mit einer Tastatur, wenn der Typ "Multistatus Schalter" ausgewählt ist. Diese Option ist besonders hilfreich wenn die Tastatur während der Laufzeit die einzige verfügbare physische Schnittstelle zur Interaktion mit der Bedienoberfläche ist.
- **Schalterereignis** Auswahlliste (nur verfügbar für **Multistatus Schalter**): Wählen Sie eine der folgenden Optionen zur Festlegung, wann die Meldung geändert wird:
  - **Beim Drücken**: Wechseln zur nächsten Meldung, wenn das Objekt angeklickt wird (Standard).
  - **Wenn gedrückt**: Kontinuierliches Wechseln zur nächsten Meldung, während die Maustaste auf dem Objekt gedrückt gehalten wird.
  - **Beim Loslassen**: Wechseln zur nächsten Meldung, wenn die Maustaste auf dem Objekt losgelassen wird.
- **E-Sign** (nur für **Multistatus Schalter** verfügbar): Wenn diese Option angewählt ist, wird der Benutzer vor dem Ausführen der Aktion zur Eingabe der elektronischen Signatur aufgefordert.
- **Sicherheit** (nur für **Multistatus Schalter** verfügbar): Für das Objekt bzw. die Animationen erforderliche Benutzerberechtigung.
- **Rand ausblenden**: Wenn diese Option angewählt ist, ist die Rahmenlinie für das Objekt unsichtbar.

- **Linienstärke**: Definiert die Stärke der um das Objekt gezeichneten Linie (der Rand).
- **Schrift**: Öffnet den Dialog Schriftarten, in dem die Schriftarteinstellungen für den im Objekt angezeigten Text konfiguriert werden können.
- **Meldungen…**: Öffnet den Dialog "Konfiguration", in dem die Meldungen für das Objekt wie folgt konfiguriert werden können:

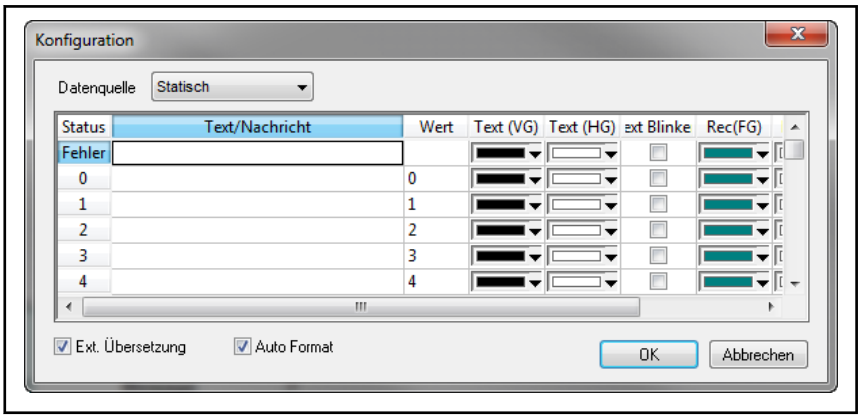


Abb.8-44: Dialog Konfiguration intelligente Meldung

  - **Datenquelle**: Die von dem Objekt angezeigten Meldungen können entweder direkt in der Tabelle eingetragen (**Statisch**) oder aus einer externen Textdatei eingelesen werden (**Textdatei**). Wenn als Datenquelle die Option **Statisch** ausgewählt ist, wird der Dialog "Konfiguration" wie oben dargestellt angezeigt und Sie können alle Einstellungen in der Tabelle vornehmen.

    Bei Auswahl **Textdatei** zeigt der Dialog "Konfiguration" ein Feld zur Eingabe von Pfad und Name der Datei an, aus der die Meldungen gelesen werden (Quelldatei).

    Weitere Informationen zur Formatierung der Textdatei finden Sie unter "Format Quelldatei" auf Seite 217.
  - **Übersetzung aktivieren**: Durch Anklicken dieser Option, wird der Text in die Übersetzungsdatei aufgenommen.
  - **Auto-Format**: Wenn diese Option aktiviert ist und die Meldung einen Dezimalwert in geschweiften Klammern (z.B. **{1,2345}**) oder einen Tag vom Typ Real auf Seite 124 enthält (siehe **Text/Meldung** unten), dann wird der Wert gemäß der von der Funktion SetDecimalPoints auf Seite SetDecimalPoints erstellten virtuelle Tabelle formatiert.

Die folgende Tabelle beschreibt die Bedeutung der mit jeder Meldung verbundenen Eigenschaften, unabhängig von der Datenquelle:

Eigenschaft	Beschreibung
Text/Meldung	Meldung (Text), die bei Auswahl während der Laufzeit angezeigt wird. Die Meldung kann Tags enthalten, wenn diese in geschweifte Klammern gesetzt werden (z.B. `{tagname}`).
Wert	Jeder Meldung muss ein eindeutiger Wert zugeordnet sein. Wenn während der Laufzeit ein passender Wert in das Tag geschrieben wird, welches im Feld "Tag lesen" parametriert ist, wird der dazugehörige Text automatisch ausgewählt und angezeigt. Gibt es zu einem Wert keine passende Textmeldung, wird während der Laufzeit die in der ersten Zeile (**Status** = **Fehler**) konfigurierte Meldung angezeigt. Wenn als Objekt-**Typ Multistatus Schalter** festgelegt ist, wird der Wert der aktuellen Meldung auch ein Tag geschrieben, sofern dieser im Feld "Tag Schreiben" konfiguriert ist.
Text (VG)	Vordergrundfarbe für die während der Laufzeit angezeigten Meldungen.

Eigenschaft	Beschreibung
Text (HG)	Hintergrundfarbe für die während der Laufzeit angezeigten Meldungen.
Text blinkend	Bei Aktivierung blinkt der Meldungstext während der Laufzeit.
Rec (VG)	Linienfarbe (Rahmen) für das Rechteck hinter der Meldung.
Rec (HG)	Hintergrundfarbe (Füllung) für das Rechteck hinter der Meldung.
Rec blinkend	Bei Aktivierung blinkt das Rechteck hinter der Meldung während der Laufzeit.
Grafikdatei	Ggf. Pfad und Name der Bitmap-Datei (*.BMP), die angezeigt wird, wenn die zugeordnete Meldung während der Laufzeit ausgewählt wird. Wenn kein Pfad angegeben wird, muss die Bitmap-Datei in Ihrem Projektordner gespeichert werden.
Transparenz	Wählen Sie die transparente Farbe in der Grafikdatei, wenn das Kontrollkästchen **Transparenz freig.** ausgewählt ist.
Transparenz aktiv	Wenn ausgewählt wird die im Feld **Transparenz** ausgewählte Farbe in der Grafikdatei als transparent festgelegt.

Die Eigenschaften **Grafikdatei**, **Transparenz** und **Transparenz aktiv** sind für den Typ **Meldungsanzeige** nicht verfügbar.

Sie können Daten aus diesem Dialog kopieren und in ein Excel-Arbeitsblatt einfügen und umgekehrt.

Format Quelldatei

Dieser Abschnitt beschreibt das von dem Objekt "SmartMessage" unterstützte Format der Textdatei, wenn als Datenquelle eine Textdatei konfiguriert ist. Der Hauptvorteil bei der Verwendung einer externen Textdatei anstelle von statischen Werten ist, dass diese die Flexibilität zum Ändern der Meldungen während der Laufzeit bietet, indem Sie zur Laufzeit den Namen der verwendeten Textdatei oder sogar den Inhalt der Textdatei ändern.

Die Textdatei muss im CSV-Format (comma separated values) erstellt werden, wobei das Kommazeichen (",") zum Trennen der Spalten (Daten) in jeder Zeile der Datei verwendet wird. Daher kann ein beliebiger CSV-Editor wie Microsoft Notepad oder Microsoft Excel zum Erstellen der CSV-Datei mit den Meldungen und deren Eigenschaften für das "SmartMessage - Objekt" verwendet werden.

Eine Beschreibung aller Eigenschaften einer Meldung finden Sie unter Object Smart Message auf Seite 214. Die Reihenfolge der Daten in der CSV-Datei ist in der folgenden Tabelle beschrieben:

Spalte Nr.	Eigenschaft	Standardwert
1	Text/Meldung	-
2	Wert	-
3	Text (VG)	0
4	Text (HG)	16777215
5	Text blinkend	0
6	Rec (VG)	8421376
7	Rec (HG)	16777215
8	Rec blinkend	0

Spalte Nr.	Eigenschaft	Standardwert
9	Grafikdatei	-
10	Transparenz	0
11	Transparenz aktiv	0

Beim Konfigurieren von Textmeldungen mit einem Kommazeichen als Teil der Meldung muss die gesamte Meldung zwischen Anführungszeichen konfiguriert werden (z.B. "Achtung, Motor abschalten"). Andernfalls wird das Komma als Datentrennzeichen und nicht als Teil der Meldung interpretiert.

- Die erste Zeile dieser Datei entspricht dem **Status** = **Fehler**. Anders ausgedrückt: Wenn zur Laufzeit, das konfigurierte Tag im Feld "Tag Lesen" einen Wert enthält, dem kein Text zugeordnet ist,, wird die in der ersten Zeile (**Status** = **Fehler**) konfigurierte Meldung angezeigt.

  Da der Fehlertext der ersten Zeile nicht durch einen bestimmten Wert ausgelöst wird, ist der Eintrag in Spalte "Wert" irrelevant. Die erste Zeile muss immer konfiguriert werden, ungeachtet des Objekttyps (auch für **Multistatus Schalter**).
- Die Spalten **Text/Meldung** und **Wert** sind Pflichtfelder. Alle anderen Spalten sind optional. Wenn hier keine Werte festgelegt werden, werden die Standardwerte verwendet (siehe Tabelle).
- Die Felder **Text(VG)**, **Text(HG)**, **Rec(VG)**, **Rec(HG)** und **Transparenz** können mit dem Code der damit verbundenen Farbe konfiguriert werden. Der Code kann direkt im Dezimalformat (z.B. 255) oder im Hexadezimalformat gemäß der Syntax #Wert (z.B. .#0000FF) eingegeben werden.
- Die Felder **Text blinkend**, **Rec blinkend** und **Freig. Transparenz** können mit Booleschen Werten "0" oder "1" (**0** = **Deaktiviert**; **1** = **Aktiviert**), oder mit den Schlüsselwörtern **FALSCH** oder **WAHR** (**FALSCH** = **Deaktiviert**; **WAHR** = **Aktiviert**) konfiguriert werden.

Beispiel:

```
Error Message,,0,16777215,1,8421376,16777215,1,error.bmp,0,0
Message Zero,0,0,16777215,0,8421376,16777215,0,open.bmp,65280,1
Message Ten,10,0,16777215,0,8421376,16777215,0,closed.bmp,65280,1
Message Twenty,20,0,16777215,0,8421376,16777215,0,,0,0
Message Thirty,30,0,16777215,0,8421376,16777215,0,,0,0
```



Der Editor des "SmartMessage"-Objekts kann bei Auswahl

Datenquelle = **Statisch**

zum Konfigurieren von Meldungen, Werten und Farben verwendet werden. Alternativ kann der Inhalt der Tabelle markiert und über die Windows-Zwischenablage (Copy&Paste) in ein Excel-Arbeitsblatt eingefügt werden.

Dann kann das Excel-Arbeitsblatt als CSV-Datei gespeichert werden (**Datei** ▸ **Speichern unter**) und in WinStudio durch Umschalten der Datenquelle auf "Text File" wieder eingebunden werden.

Diese Vorgehensweise bietet eine benutzerfreundliche Schnittstelle zum Konfigurieren der Farbcodes.

8.3.4 WinStudio-Bibliotheken

Symbole

Symbole – Allgemeines

Die **Bibliothek** ist ein visueller Browser für alle Symbole auf Seite 220, die zum Einfügen in ein WinStudio-Bild zur Verfügung stehen.

Zum Öffnen des Werkzeugfensters "WinStudio-Symbole" wählen Sie den Eintrag **WinStudio-Bibliothek** im Kontextmenü **WinStudio** ▸ **Werkzeuge** oder im Kontextmenü **Anwendersymbole**

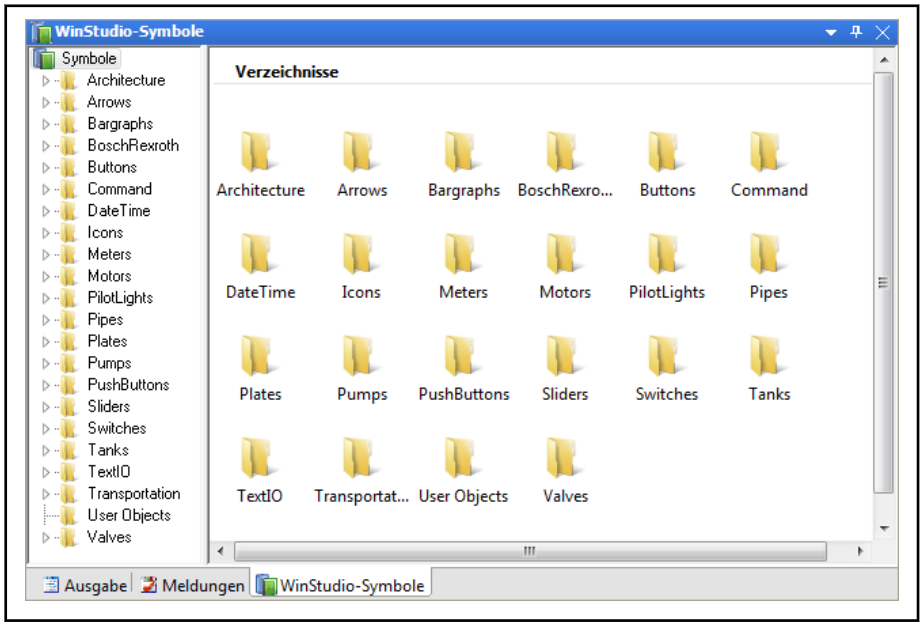


Abb.8-45: Fenster Symbolbibliothek

Das Werkzeugfenster "WinStudio-Symbole" bietet eine umfangreiche Auswahl von vordefinierten Symbolen. Diese sind in Ordnern nach Kategorie (z.B. Schaltflächen, Zähler, Tanks) sortiert .

Um ein Symbol auszuwählen und in ein WinStudio-Bild einzufügen, gehen Sie folgendermaßen vor:

1. Suchen Sie in der Bibliothek das gewünschte Symbol und klicken Sie dieses an

   Der Mauszeiger ändert sich, um anzuzeigen, dass ein Symbol selektiert ist
2. Kehren Sie zum WinStudio-Bild zurück, in das das Symbol eingefügt werden soll
3. Klicken Sie auf eine beliebige Stelle im Bild, um das gewählte Symbol zu platzieren
4. Bearbeiten Sie bei Bedarf die Objekteigenschaften auf Seite 178 des Symbols

Weitere Informationen unter Speichern von eigenen Projektsymbolen auf Seite 220.

Benutzerdefiniertes Symbol für andere Projekte zur Verfügung stellen

Benutzerdefinierte Symbole sind nur in dem Projekt verfügbar, in welchem sie ursprünglich erstellt und gespeichert wurden, d.h. der Ordner "Anwendersymbole" im Projektbaum ist spezifisch für dieses Projekt. Benutzerdefinierte Symbole können aber in den Ordner "User Objects" der WinStudio-Symbolbibliothek gesendet und damit für alle Projekte verfügbar gemacht werden:

Bilder und Grafiken

1. Wählen Sie das Symbol im Order "Anwendersymbole" aus, das für andere Projekte verfügbar gemacht werden soll.

   Wählen Sie im Kontextmenü dieses Elements den Eintrag **In WinStudio-Bibliothek übernehmen** aus

2. Das Symbol wird mit Hilfe dieser Funktion in das Verzeichnis "..\Users\All Users\WinStudio\Symbols" kopiert und erscheint nun im Ordner "User Objects"

Speichern von eigenen Projektsymbolen

Ein **Symbol** ist eine Gruppe aus einem oder mehreren Objekten, welches im Ordner "Anwendersymbole" gespeichert ist und so beliebig oft in Ihrem Projekt verwendet werden kann.

Bei jeder Verwendung eines Symbols wird im Wirklichkeit eine Kopie erstellt, die mit der Master-**Symbol**datei im Ordner "Symbole" verknüpft ist.

Diese verknüpften Kopien werden auch als „Instanzen“ des Symbols bezeichnet.

Wenn danach Veränderungen am Master-Symbol vorgenommen werden, werden diese Änderungen automatisch an jede verknüpfte Kopie in allen WinStudio-Bildern weitergegeben.

Jede verknüpfte Kopie des Master-Symbols kann durch Ergänzung von **Benutzerdefinierten Eigenschaften** individuell erweitert werden.

Wenn zum Beispiel ein Messgerät zur Anzeige von Tankfüllständen erstellt und als Master-Symbol gespeichert wird, kann jede Instanz individuell ergänzt werden:

Jede Instanz des Messgerätes kann für einen anderen Tank den Füllstand anzeigen.

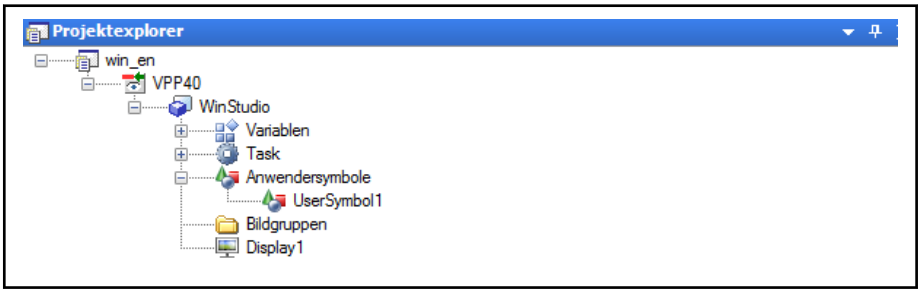


Abb.8-46: Ordner "Anwendersymbole"

Erstellen eines Master-Symbols

Erstellen eines Master-Symbols und Speichern im Ordner "Anwendersymbole":

1. Entwerfen Sie Ihr Symbol wie beim normalen Bearbeiten eines WinStudio-Bildes mit einer beliebigen Kombination aus statischen auf Seite 182 und aktiven Objekten auf Seite 190.

   Zum Beispiel drei Kontrollkästchen auf Seite 198 im einem rechteckigen Bereich auf Seite 185:

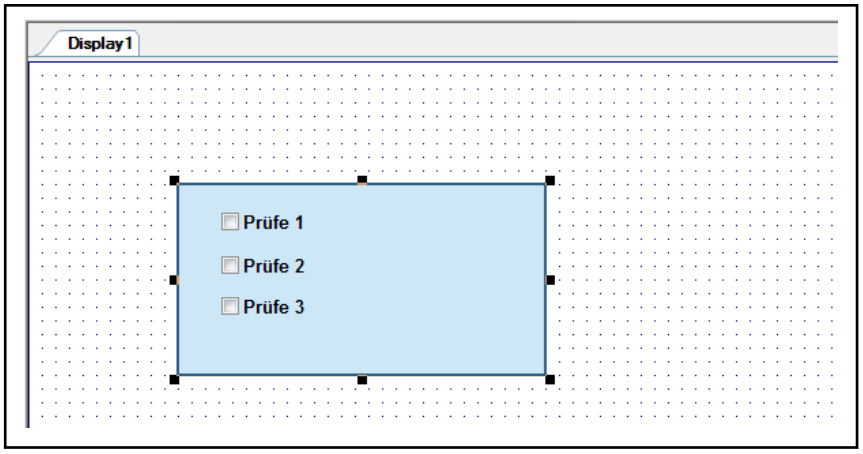


Abb.8-47: Hinzufügen von Objekten in einem WinStudio-Bild

2. Wählen Sie das/die Objekt(e) oder Gruppe auf Seite 259, die als Symbole gespeichert werden sollen.

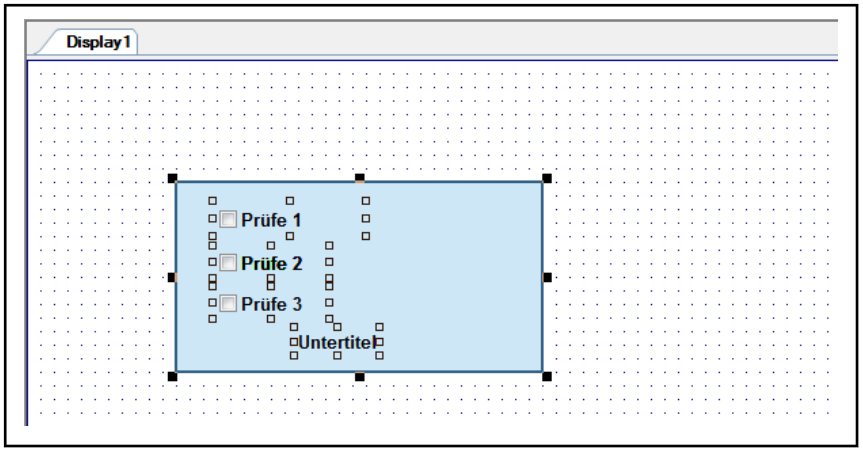


Abb.8-48: Auswahl der Objekte

Es ist nicht erforderlich, eine Gruppe aus zwei oder mehr Objekten zu erstellen, bevor diese als Symbol gespeichert werden. Die gemeinsame Speicherung von Objekten als Symbol führt diese Gruppierung aus.

Es gibt allerdings einen Anwendungsfall, er eine Gruppierung der Objekte erfordert:

Ein Symbol kann nur einen einzigen Tooltip anzeigen.

Wenn aus mehreren Objekten, ohne vorherige Gruppierung, ein Symbol generiert wird, können evtl. vorhandene Tooltips nicht angezeigt werden (in den Objekteigenschaften auf Seite 178).

Um einen Tooltip für das Symbol festzulegen, müssen erst die Objekte gruppiert und dann der Tipp für die Gruppe konfiguriert werden. Dieser Tipp bleibt erhalten, wenn die Gruppe als Symbol gespeichert wird.

3. Führen sie einen Rechtsklick auf die Auswahl aus und klicken Sie dann im Kontextmenü auf **Verlinktes Symbol erzeugen**.

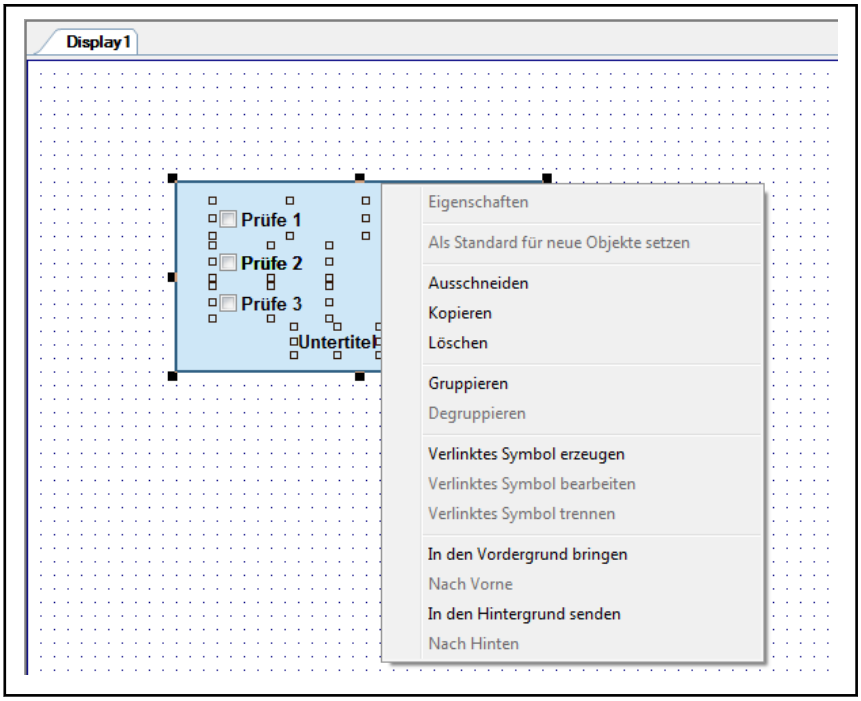


Abb.8-49: Erstellen eines verlinkten Symbols

4. Ein Standarddialog "Speichern unter" wird angezeigt und fordert zur Eingabe eines Dateinamens für das neue Symbol auf. Symboldateien (".sym") werden im Ordner **\Symbol** Ihres Projekts gespeichert.

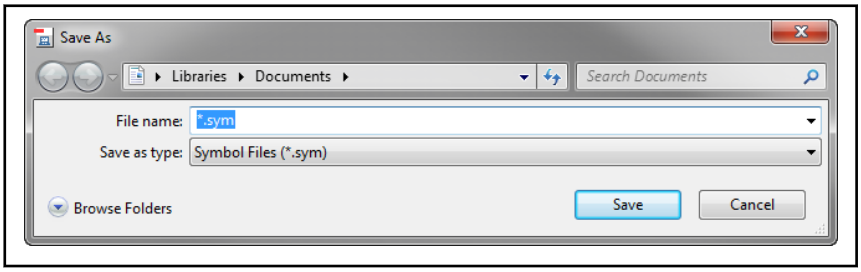


Abb.8-50: Speichern der Symboldatei test.sym

5. Klicken Sie auf **Speichern**, um die Datei zu speichern.

   Das Symbol erscheint nun im Ordner "Anwendersymbole" auf der Registerkarte Grafik auf Seite 170 des Projekt-Explorers.

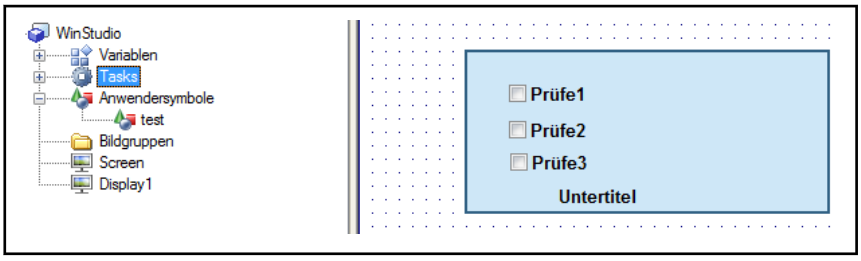


Abb.8-51: Symboldatei im Projekt-Explorer

Das Symbol kann jetzt in Ihrem Projekt wiederverwendet werden, aber in der aktuell gespeicherten Form wird jede Kopie die gleichen Eigenschaften haben.

Es müssen nun benutzerdefinierte Eigenschaften für das Symbol definiert werden, d.h. die Eigenschaften, die Sie bei jeder Verwendung des Symbols anpassen möchten.

Bearbeiten des Master-Symbols

Ein Master-Symbol kann nach dem Speichern bearbeitet werden, z.B. durch Hinzufügen oder Löschen von Objekten im Symbol oder durch Definieren von benutzerdefinierten Eigenschaften.

Bedenken Sie, dass alle am Master-Symbol vorgenommenen Änderungen automatisch an die verknüpften Kopien (Instanzen) im gesamten Projekt. weitergegeben werden. So wurde z.B. die Schriftfarbe der Unterschrift von schwarz auf blau geändert...

Es gibt eine Ausnahme:

Wenn der Tipp für ein Symbol geändert wird – wie unter "Erstellen eines Master-Symbols" auf Seite 220 beschrieben – dann zeigt sich diese Änderung nur an neuen Instanzen dieses Symbols. Bestehende Instanzen bleiben unverändert.

Bearbeiten eines Symbols:

1. Führen Sie im Ordner "Anwendersymbole" einen Doppelklick auf die Symboldatei aus.



Sie können auch auf eine beliebige Instanz des Symbols einen Rechtsklick ausführen und **Verknüpftes Symbol bearbeiten** aus dem Kontextmenü auswählen.

Die Symboldatei wird zum Bearbeiten in einem eigenen Fenster geöffnet.

Dieser Symbol-Editor funktioniert auf die gleiche Weise wie der WinStudio-Bild-Editor, außer dass jedes Objekt im Fenster zu einem Symbol gehört.

Wenn Objekte im Symbol-Editor hinzugefügt, verschoben oder gelöscht werden, kann dies die Größe oder Form des Symbols verändern und damit das Layout der WinStudio-Bilder beeinflussen, in denen es verwendet wird.

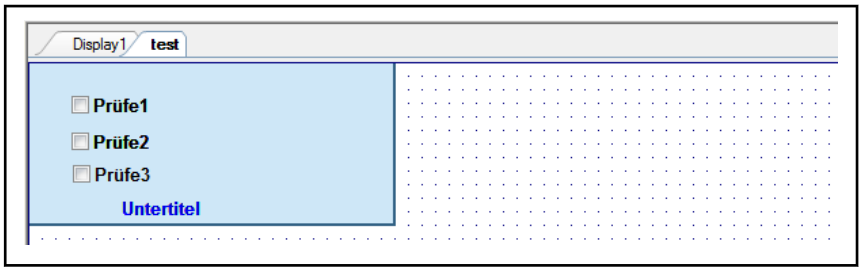


Abb.8-52: Symboldatei zum Bearbeiten geöffnet

Neben dem Hinzufügen, Verschieben oder Löschen von Objekten im Symbol können auch die Objekteigenschaften auf Seite 178 wie gewohnt bearbeitet werden.

Nicht immer ist es sinnvoll, dass alle Eigenschaften eines Symbols in allen Instanzen gleich ist.

In unserem Beispiel sollen die hinterlegten Tags auf Seite 116 und die Beschriftung der drei Kontrollkästchen für jede Instanz frei wählbar sein.

2. Wählen Sie das erste Objekt des Symbols und öffnen Sie dessen Objekteigenschaften. Zum Beispiel das erste Kontrollkästchen:

Bilder und Grafiken

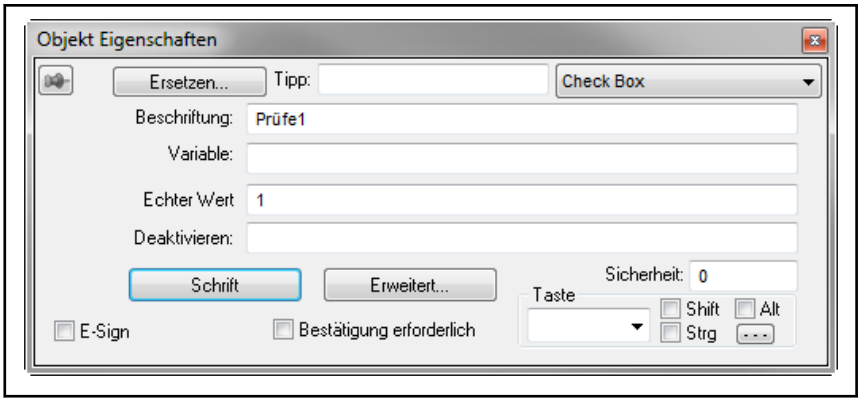


Abb.8-53: Objekteigenschaften für das erste Kontrollkästchen

3. In allen Feldern, in denen normalerweise Tags, Ausdrücke oder Werte konfiguriert werden, kann statt dessen eine benutzerdefinierte Eigenschaft definiert werden nach der Syntax:

   ```
   # [Kategorie.] Eigenschaft : [Wert]
   ```

   … wobei gilt:

   - **Kategorie** ist ein "optionaler" Name als Sammelbegriff für Eigenschaften, die zusammengefasst werden sollen, z.B. alle Bezeichnungen oder alle Kontrollkästchenwerte. Wenn Sie keine **Kategorie** für eine Eigenschaft festlegen, wird sie automatisch unter der Kategorie „Main“ aufgeführt.
   - **Eigenschaft** ist der Anzeigename zur Beschreibung einer Eigenschaft. Ein Anzeigename wird für jede Eigenschaft benötigt und muss immer einen Doppelpunkt ":" nachgestellt haben.
   - **Wert** ist ein "optionaler" Standardwert für die Eigenschaft.



Alle Syntaxregeln für Tags bzw. Ausdrücke gelten für "Wert", einschließlich Tag-Namen auf Seite 121, Zeiger auf Seite 131, Arrays auf Seite 125, Zeichenketten, numerische und Boolesche Werte und Skripting-Funktionen.

Im folgenden Beispiel soll der Tag, der den aktuellen Zustand des Kontrollkästchens beinhaltet, für jede Instanz des Master-Symbols individuell festgelegt werden.

Geben Sie im Feld **Variable** den Text **#Option1.Tag:** ein, wie gezeigt.

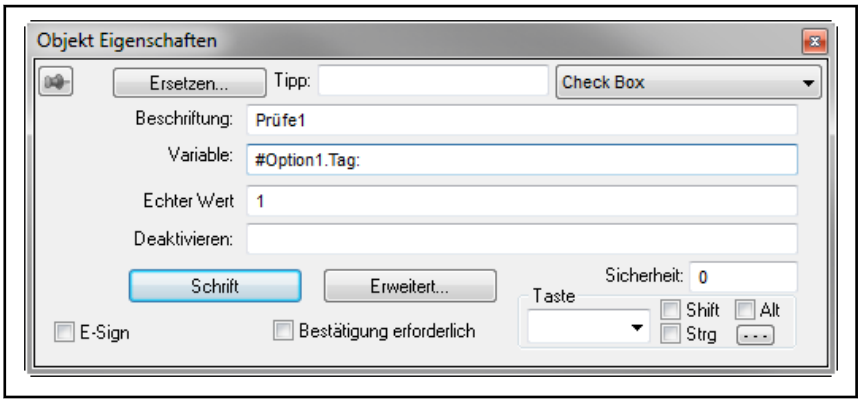


Abb.8-54: Definieren einer benutzerdefinierten Eigenschaft für das Tag-Feld

Der Eintrag "#Option1.Tag:" im Master- Symbol führt im Eigenschaftsdialog einer Symbol-Instanz zu folgender Darstellung:

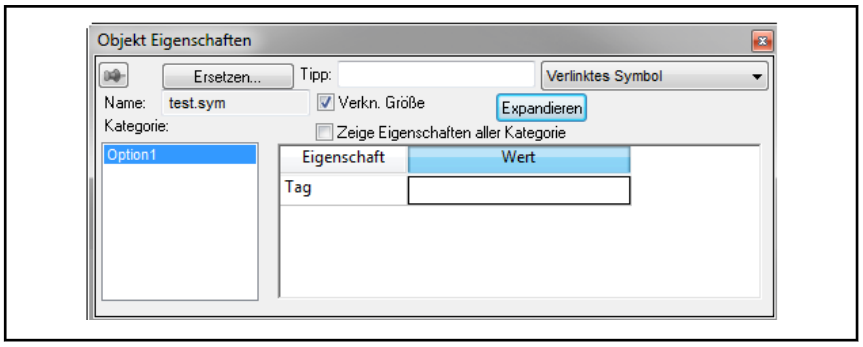


Abb.8-55: Benutzerdefinierte Eigenschaften in einem Symbol

In der Spalte "Wert" muss dann ein Tag eingetragen werden, der für diese Instanz den Status des Kontrollkästchens enthalten soll. Weitere Details siehe weiter unten.

4. Objekteigenschaftsfelder die normalerweise einfache Texte (Zeichenketten) erwarten, können durch dynamische Elemente (Platzhalter) in geschweiften Klammern ergänzt werden. In diesem Beispiel erfordert das Feld **Beschriftung** eine Zeichenkette. Geben Sie also wie gezeigt **{#Option1.Beschriftung:"Option 1"}** ein.

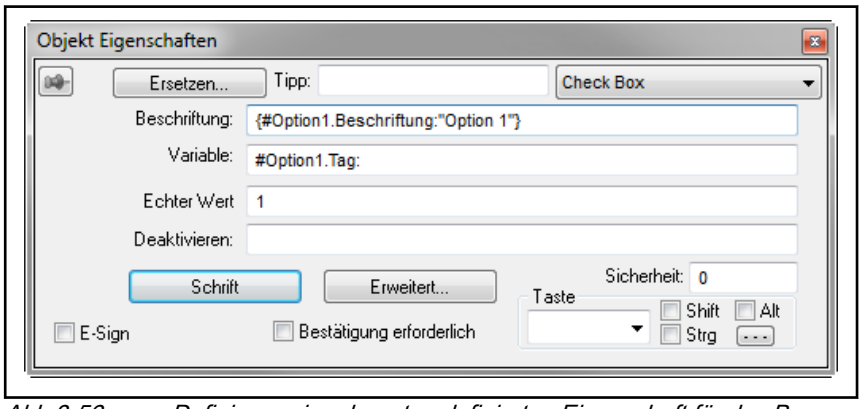


Abb.8-56: Definieren einer benutzerdefinierten Eigenschaft für das Beschriftungsfeld

Der Eintrag im Master- Symbol führt im Eigenschaftsdialog der Symbol-Instanz zu folgendem zusätzlichen Eintrag:

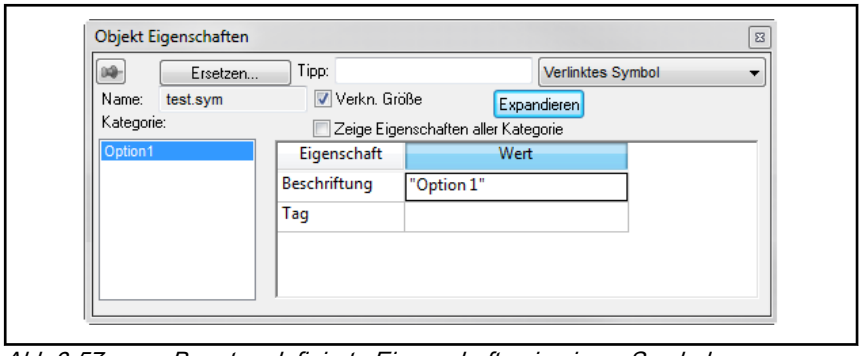


Abb.8-57: Benutzerdefinierte Eigenschaften in einem Symbol

5. Wiederholen Sie Schritte 2 bis 4 zur Parametrierung der restlichen benutzerdefinierten Eigenschaften in dem Master-Symbol. In diesem Bei-

spiel zeigt der Eigenschaftsdialog der Symbol-Instanz folgende Einträge:

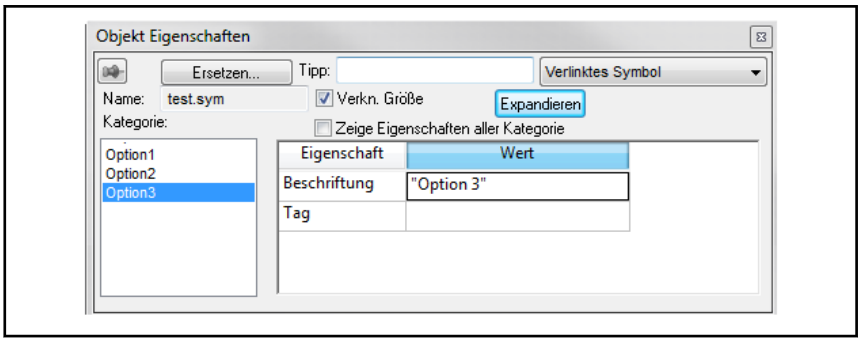


Abb.8-58: Benutzerdefinierte Eigenschaften in einem Symbol

Zeige **Eigenschaften aller Kategorien** erlaubt es alle Objekt-Eigenschaften des Symbols untereinander anzuzeigen.

6. Speichern Sie das Symbol und schließen Sie den Symbol-Editor.
7. Führen Sie nun "Anwendung verifizieren" aus dem Kontextmenü des Ordners "WinStudio" aus. Dadurch werden alle bestehenden Instanzen des Symbols in Ihrem Projekt aktualisiert.

Hinzufügen von ToolTips zu benutzerdefinierten Eigenschaften

Für jede im Master-Symbol verfügbare benutzerdefinierte Eigenschaft kann eine Beschreibung hinterlegt werden, die das Konfigurieren der Symbol-Instanzen vereinfacht.

Öffnen Sie das Symbol nach dem Erstellen mit dem Symbol-Editor, führen sie einen Rechtsklick in den Symbol-Editor aus (nicht auf das Symbol selbst) und wählen Sie **Symboleigenschaften bearbeiten** aus dem Kontextmenü.

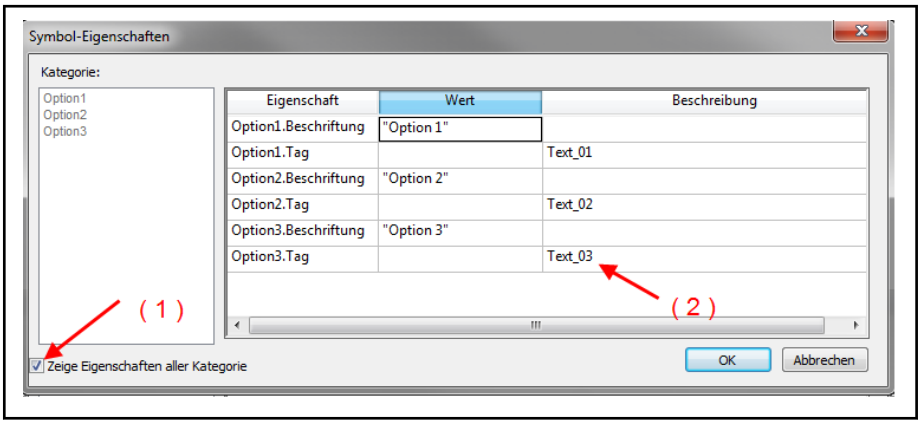


Abb.8-59: Symboleigenschaften bearbeiten

Mit dem Kontrollkästchen (1) **Zeige Eigenschaften aller Kategorien** können die Informationen zu allen Objekten unter einander angezeigt werden.

In der Spalte (2) **Beschreibung** wird jeweils ein Text eingetragen, der später als Tooltip verwendet werden soll.

Beim Einfügen und Konfigurieren einer Symbol-Instanz im Bild kann der Benutzer die Beschreibung als ToolTip lesen, indem er einfach den Mauszeiger über den Namen der Eigenschaft bewegt, wie im folgenden Bild dargestellt:

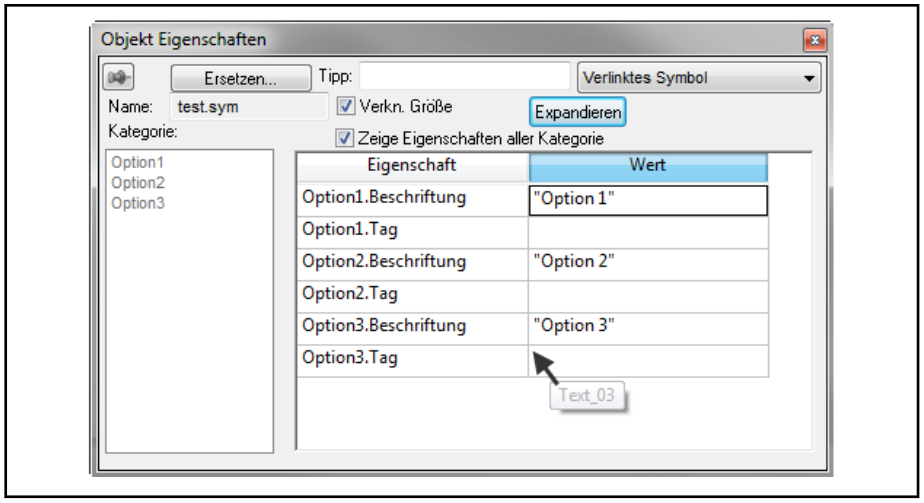


Abb.8-60: ToolTip mit Beschreibung der Eigenschaft

Passwortschutz für Symbole

Alle vom Benutzer erstellten Symbole können mit einem Passwort belegt werden, um zu verhindern, dass sie von anderen Benutzern bearbeitet oder analysiert werden. Schützen eines Symbols:

1. Führen Sie im Ordner "Anwendersymbole" einen Rechtsklick auf die gewünschte Symboldatei (**.sym**) aus und wählen Sie **Passwort ändern** aus dem Kontextmenü. Der Dialog "Schutz bearbeiten" wird angezeigt
2. Geben Sie Ihr Passwort in das Feld **Neues Passwort** ein
3. Geben Sie Ihr Passwort erneut in das Feld **Passwort bestätigen** ein
4. Klicken Sie auf **OK**

Danach wird das Passwort jedes Mal abgefragt, wenn sie versuchen, das Symbol zu bearbeiten oder für eine Instanz des Symbols die Verknüpfung aufzuheben.

Einfügen von Symbolen in ein WinStudio-Bild

Einfügen eines Symbols in ein Bild und Vervollständigen der benutzerdefinierten Symbol-Eigenschaften:

1. Öffnen Sie das gewünschte Bild aus dem Ordner Bilder auf Seite 170 oder fügen Sie ein neues Bild ein.
2. *Öffnen Sie die Symbolbibliothek auf einen der folgenden Wege:*
   - öffnen Sie das Kontextmenü **WinStudio ▸ Werkzeuge** und wählen "WinStudio-Bibliothek" aus oder
   - öffnen Sie das Kontextmenü **Anwendersymbole** und wählen "WinStudio-Bibliothek" aus oder
   - führen sie einen Rechtsklick im WinStudio-Bildeditor aus, in den das Symbol eingefügt werden soll, und klicken dann im Kontextmenü auf **Verknüpftes Symbol einfügen**
3. Wählen Sie das Symbol aus der Symbolbibliothek und klicken Sie dann in das WinStudio-Bild:

Bilder und Grafiken

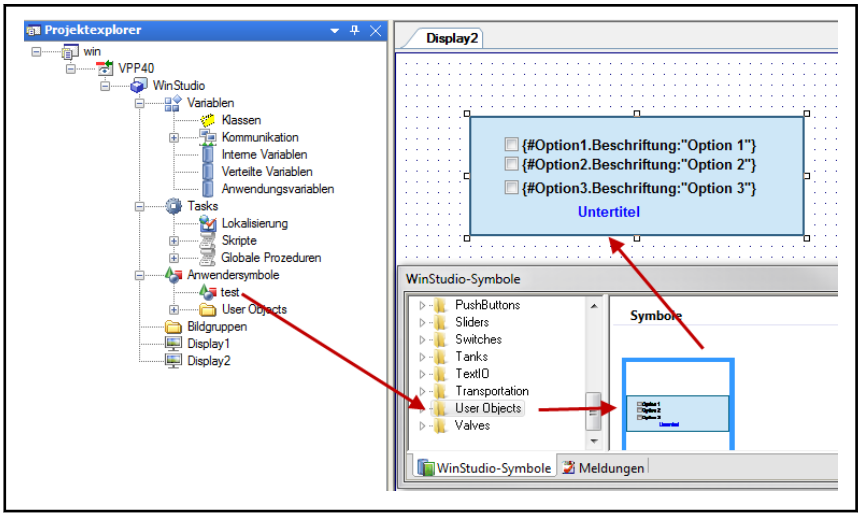


Abb.8-61: Symbol in einem BILD platzieren

Wenn das Symbol eingefügt ist, kann es wie alle anderen Bildobjekte manipuliert werden. Es kann an anderen Objekten ausgerichtet und verteilt auf Seite 257 werden und es können Animationen auf Seite 243 angewendet werden.

Zunächst müssen jedoch die benutzerdefinierten Eigenschaften für diese Instanz des Symbols fertig gestellt werden.

4. Öffnen Sie die Objekteigenschaften für das Symbol
5. Weisen Sie den Übergabeparametern "Tag" echte Prozessgrößen (Texte in Anführungszeichen bzw. Tag-Namen) zu.

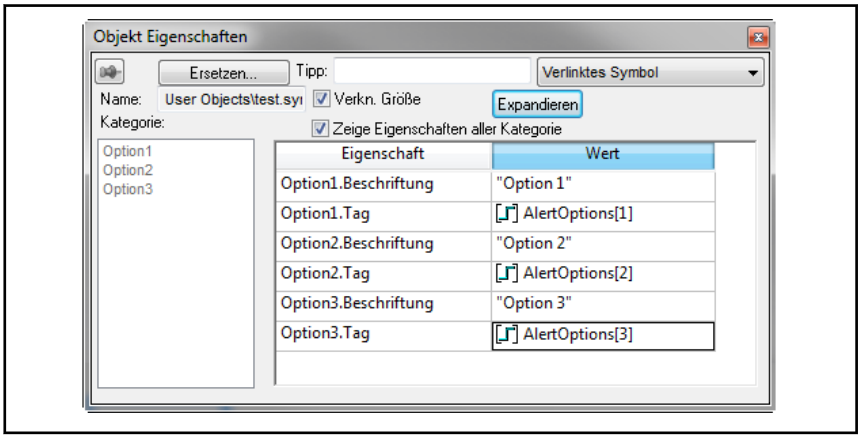


Abb.8-62: Fertiggestellte Eigenschaften für das Symbol

6. Klicken Sie zum Schließen des Dialogs "Symboleigenschaften" auf **OK** und schließen Sie dann den Dialog "Objekteigenschaften"

Die Platzhalter im Engineering {} werden zur Laufzeit durch die echten (zugewiesenen) Prozesswerte (Texte , Tag-Wert) ersetzt.

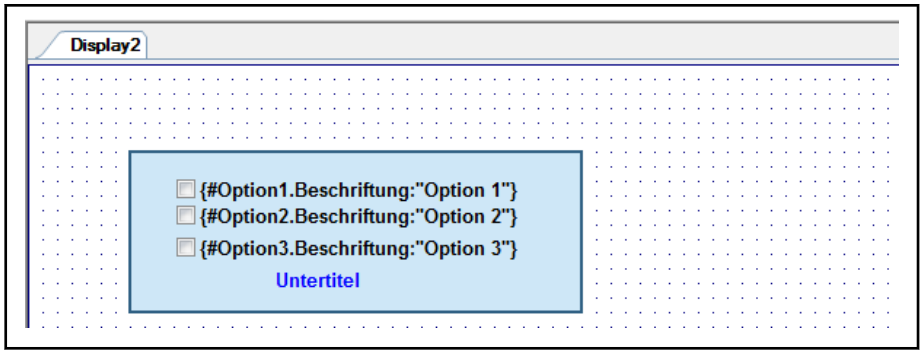


Abb.8-63: Symbol beim Bearbeiten

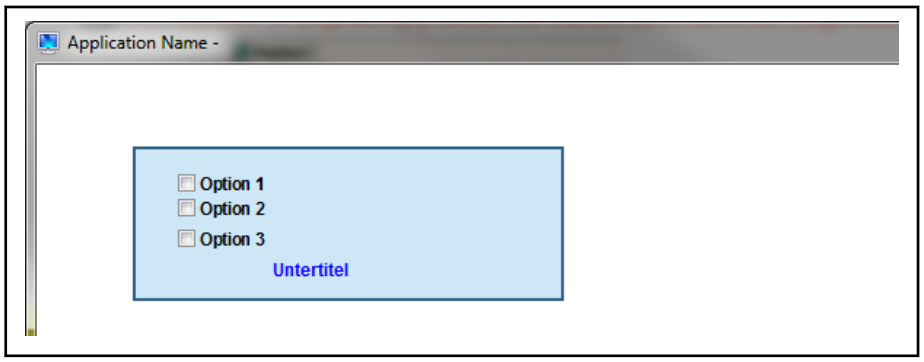


Abb.8-64: Symbol während der Laufzeit

Bedenken Sie, dass die zugewiesenen Prozesswerte für jede Instanz eines Master-Symbols unabhängig von allen anderen Instanzen dieses Master-Symbols sind.

Wenn aber an der Master-Symboldatei Änderungen vorgenommen werden, so werden diese Änderungen automatisch an alle Instanzen weitergegeben.

Active X-Control

Klicken Sie im Werkzeugfenster **WinStudio Screen Tools** auf **ActiveX-Control**. Es öffnet sich der Dialog "ActiveX-Control einfügen", mit dem Sie ActiveX-Elemente auswählen und in ein WinStudio-Bild einfügen können.

Wenn der Dialog geöffnet wird, enthält er eine Liste aller ActiveX-Elemente, die auf Ihrem PC registriert sind:

Bilder und Grafiken

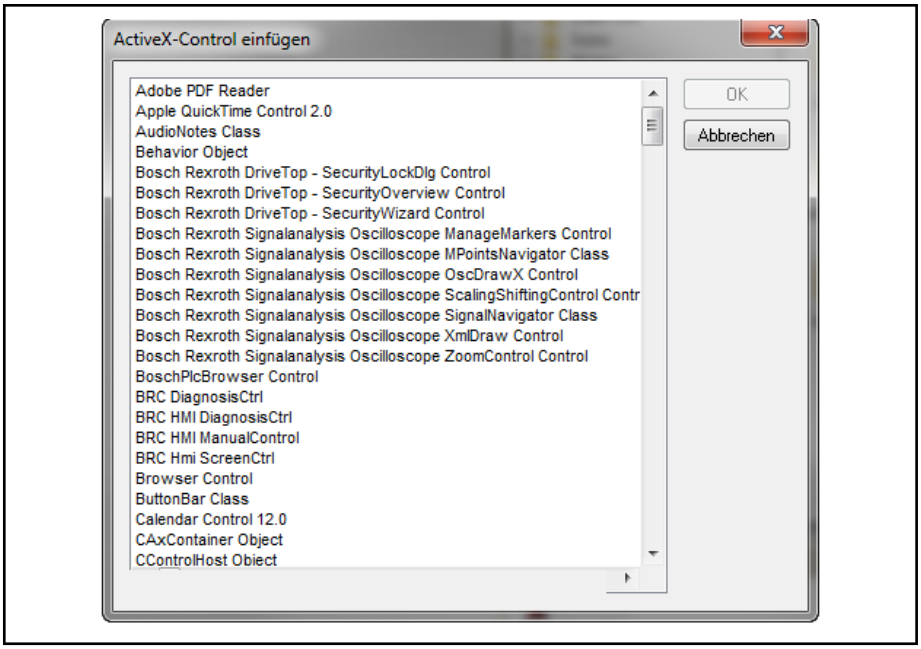


Abb.8-65: Dialog ActiveX-Steuerelemente einfügen

Wenn Sie in Ihrem Projekt ActiveX-Controls verwenden möchten, müssen die gleichen Controls auf Ihren Laufzeitstationen installiert und registriert sein.

Da die Funktion zum "Automatischen Herunterladen" und "Automatischen Registrieren" auf den Zielgeräten aus Sicherheitsgründen meist deaktiviert ist, sind die ActiveX Controls nach dem Projekt- Download nicht immer verfügbar.

Um ActiveX Controls trotz Sicherheitseinstellungen nutzen zu können, müssen diese manuell auf das Zielgerät übertragen und registriert werden. Wenn das ActiveX Control erstmalig registriert ist, kann ein Update mit dem Projekt durch eine Anpassung der Projektdatei (projekt_name.app) erreicht werden:

Wenn Sie das automatische Herunterladen von ActiveX-Steuerelementen trotzdem freigeben wollen, kann dies durch manuelles Bearbeiten Ihrer Projektdatei (**projekt_name.app**) zum Hinzufügen der folgenden Einstellungen erreicht werden:

Projektdatei "projekt_name.app"

```
[UsedControls]
EnableDownload=1
Count=Anzahl der Steuerelemente

[Used Control1]
CLSID= Klassen-ID des ActiveX-Controls
Version = Version des ActiveX-Controls
Codebase= URL der ActiveX-Control Datei oder der .CAB-Datei mit den ActiveX-Control Dateien
RegFile1= Name einer bestimmten .OCX oder .DLL Datei in der .CAB Datei; siehe unten
RegFilen= Name einer bestimmten .OCX oder .DLL Datei in der .CAB Datei; siehe unten
...
[UsedControln]
...
```

Die Einstellungen "CLSID" und "Version" werden für jedes ActiveX-Control benötigt und müssen zu ID und Version der Control-Datei(en) passen, mit denen die "Codebase" verknüpft ist.

Dadurch kann eine Laufzeitstation das ActiveX-Control gegen bereits registrierte Elemente prüfen. Wenn die Einstellungen nicht übereinstimmen, lädt die Laufzeitstation evtl. unnötigerweise das gleiche Steuerelement noch einmal herunter.

Wenn Ihnen die Einstellungen von "CLSID" und "Version" für ein ActiveX-Steuerelement nicht bekannt sind, können Sie diese im Registry-Schlüssel eines bereits installierten und registrierten Steuerelements finden. Suchen Sie in der Windows Registry nach der Steuerdatei **HKEY_CLASSES_ROOT \CSLID**.

URL für die Einstellung "Codebase" kann entweder absolut oder relativ zum „Start“-Verzeichnis des Webservers sein.

Zum Beispiel:

```
Codebase=http://"server_adresse"/AddOns/IndDateTimePick.ocx
      …oder…
Codebase=AddOns/IndDateTimePick.ocx
```

Die Einstellungen für "Regfile" werden nur benötigt, wenn "Codebase" mit einer .CAB-Datei (Container für mehreren ActiveX -Controls) verknüpft ist, dann muss für jedes enthaltene ActiveX-Control ein "Regfile" - Eintrag aufgenommen werden.

ActiveX-Controls sind Komponenten, die nach einer Norm entwickelt werden. Sie können in "Rexroth WinStudio" Anwenderbildern eingebunden und zur Laufzeit ausgeführt werden. ActiveX-Controls können die folgenden Schnittstellen zur Verfügung stellen:

- **Eigenschaften**: Variablen, deren Werte für Ihr Projekt gelesen und/oder geschrieben werden können (zum Beispiel Objektfarbe, Dateiname, URL usw.)
- **Methoden**: Funktionen des ActiveX-Objekts, die durch Ihr Projekt ausgelöst werden können (z.B. Dialog öffnen, Berechnung ausführen usw.)
- **Ereignisse**: Interne Meldungen, die das Ausführen von Ausdrücken im Ihrem Projekt auslösen können (z.B. Maus_Klick, Download_abgeschlossen usw.)

Die Namen der von jedem ActiveX-Objekt unterstützten Eigenschaften, Methoden und Ereignisse hängen von dessen Implementierung ab.

Es gibt zwei verschiedene Möglichkeiten für eine Schnittstelle zwischen Ihrem Projekt und den ActiveX-Steuerelementen:

- Durch Verwenden der ActiveX-Funktion

  XGet() auf Seite 763,

  XSet() auf Seite 765 und

  XRun() auf Seite 764

  oder
- durch Verwendung des Fensters "Objekteigenschaften" zum Konfigurieren des Objekts

Mit einem Doppelklick auf das ActiveX-Control öffnet sich der Dialog "Objekteigenschaften".

Bilder und Grafiken

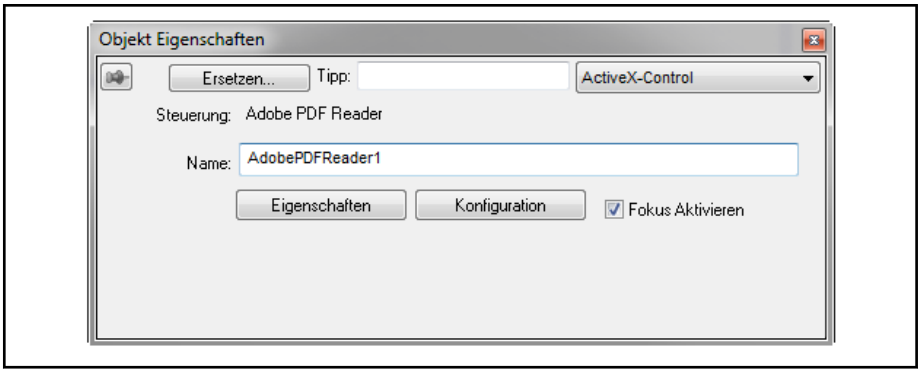


Abb.8-66: Objekteigenschaften: ActiveX-Steuerelement

Das Fenster "Objekteigenschaften" zeigt den Namen des ActiveX-Controls an. Im allgemeinen ist jedes ActiveX-Control entweder eine ***.dll** oder ein ***.ocx** Datei, die auf Ihrem lokalen Rechner registriert ist.

Sie müssen dem ActiveX-Control im Feld "Name" einen eindeutigen Namen (Alias) zuweisen (z.B. MyControl). Dieser Name wird für Funktionsaufrufe aus der Scripting-Sprache benötigt.

☞ Es dürfen niemals zwei ActiveX-Controls mit dem gleichen Namen im selben Bild konfiguriert werden.

Die Schaltfläche **Eigenschaften** öffnet das Standardfenster zum Konfigurieren der statischen Eigenschaften (falls vorhanden). Layout und Optionen in diesem Dialog sind abhängig von der Implementierung jedes ActiveX-Controls. Verwenden Sie diese Schnittstelle zum Festlegen von Eigenschaften, die während der Laufzeit nicht geändert werden sollten (fest konfigurierte Eigenschaften).

Die Schaltfläche **Konfiguration** im Fenster "Objekteigenschaften" öffnet einen Konfigurationsdialog für die folgenden Aufgaben:

- Tags mit Eigenschaften des ActiveX-Controls verbinden
- Methoden aus dem ActiveX-Control auslösen auf Grundlage von Tag-Änderungen und
- Skripte konfigurieren, die bei bestimmten Ereignissen im ActiveX-Control ausgeführt werden

Die folgenden Abschnitte beschreiben die Konfiguration dieser Schnittstellen.

Obwohl der Konfigurationsdialog die Liste aller Eigenschaften, Methoden und Ereignisse anzeigt, müssen nur die Elemente konfiguriert werden, die Sie für Ihre Anwendung benötigen.

Die in den folgenden Abschnitten verwendeten Screenshots stellen das ActiveX-Control "Acrobat 3D Office" dar.

Die Namen von Eigenschaften, Methoden und Ereignissen unterscheiden sich für jedes ActiveX-Control, die Konfigurationsschnittstelle ist aber gleich. Die nachfolgend beschriebene Vorgehensweise ist für alle Controls gleich.

Konfigurieren von Eigenschaften

Die Registerkarte "Eigenschaften" beinhaltet eine Tabelle mit einer Zeile für jede parametrierbare Eigenschaft:

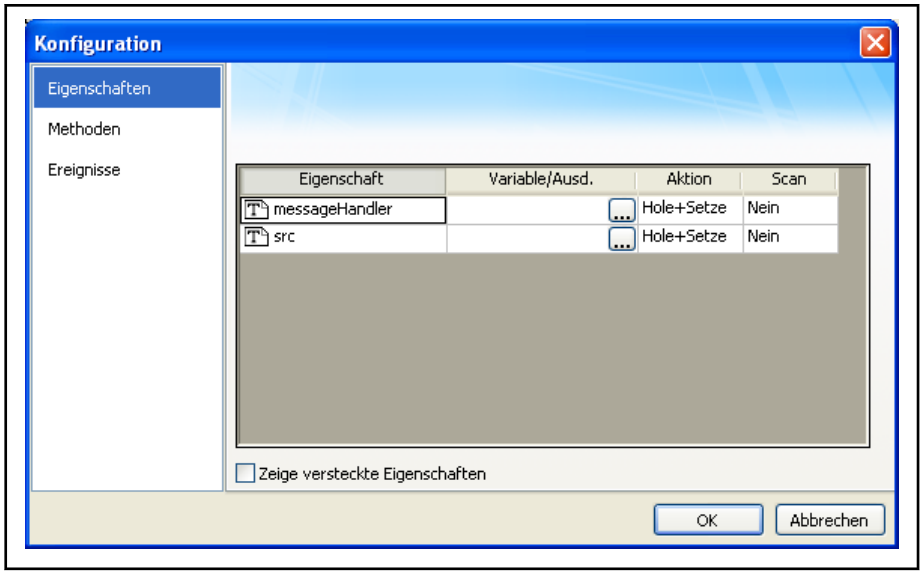


Abb.8-67: Konfigurationsdialog – Registerkarte Eigenschaften

- Spalte **Eigenschaft**: Bezeichnung aller in diesem ActiveX-Control verfügbaren Eigenschaften inkl. der jeweiligen Typkennung.

Symbol Eigenschaft	Typ
	Boolean
	Integer
	Real
	String

- Spalte **Variable/Ausdruck**: Der in diesem Feld konfigurierte Tag ist mit der entsprechenden Eigenschaft des ActiveX-Controls verbunden. Die Spalte "Aktion" bestimmt, ob der Wert dieses Tags in die ActiveX-Eigenschaft geschrieben wird oder ob der Wert der ActiveX-Eigenschaft in diesen Tag geschrieben wird (oder beides).

☞ Sie können in diesem Feld einen Ausdruck konfigurieren, wenn Sie das Ergebnis eines Ausdrucks in die Eigenschaft des ActiveX-Objekts schreiben wollen.

In diesem Fall kann der Wert der Eigenschaft allerdings nicht zu einem Tag zurückgelesen werden (es sei denn, die Funktion XGet() wird verwendet).

- Spalte **Aktion**: In dieser Spalte wird festgelegt ob der Parameter in Spalte "Variable/Ausdruck" aus dem Control gelesen oder in das Control geschrieben wird. Für den Fall, dass sowohl gelesen als auch geschrieben werden soll (Hole+Setze) muss die Reihenfolge der Aktionen festgelegt werden (siehe nachstehendeTabelle):

  Die Spalte "Aktion" wird automatisch auf "Setzen" gestellt, wenn der Ausdruck in Spalte "Variable/Ausdruck" nicht beschrieben werden kann.

Bilder und Grafiken

Aktion	Beschreibung
Holen	Liest den Wert der ActiveX-Eigenschaft und schreibt ihn in den im Feld **Tag/Ausdruck** konfigurierten Tag.
Setzen	Schreibt den Wert aus dem im Feld **Tag/Ausdruck** konfigurierten Tag oder Ausdruck in die ActiveX-Eigenschaft.
Holen+Setzen	Führt beide Aktionen aus (**Holen** und **Setzen**). Allerdings führt "WinStudio" beim Öffnen eines Bildes mit dem ActiveX-Control den Befehl **Holen** vor dem Ausführen eventueller **Setzen**-Befehle aus. D. h. wenn "WinStudio" das Bild öffnet, in dem das ActiveX-Control konfiguriert ist, wird der im Feld **Tag/Ausdruck** konfigurierte Tag mit dem Wert aus der ActiveX-Eigenschaft aktualisiert.
Setzen+Holen	Führt beide Aktionen aus (**Holen** und **Setzen**). Allerdings führt "WinStudio" beim Öffnen eines Bildes mit dem ActiveX-Objekt den Befehl **Setzen** vor dem Ausführen eventueller **Holen**-Befehle aus. D. h. wenn "WinStudio" das Bild öffnet, in dem das ActiveX-Element konfiguriert ist, wird die ActiveX-Eigenschaft mit dem Wert des im Feld **Tag/Ausdruck** konfigurierten Tags aktualisiert.

- Spalte **Scan**: Definiert die Aufrufmethode zum Holen von Werten aus den ActiveX-Eigenschaften, gemäß der folgenden Tabelle:

Scan	Beschreibung
Nein	Der Wert der ActiveX-Eigenschaft wird gelesen und in den im Feld Tag/Ausdruck konfigurierten Tag geschrieben, aber erst dann, wenn das Bild mit dem ActiveX-Objekt geöffnet ist und wenn das ActiveX-Objekt eine Meldung zum Aktualisieren dieses Tags an "WinStudio" sendet.
Immer	"WinStudio" liest kontinuierlich den Wert der ActiveX-Eigenschaft und aktualisiert den im Feld "Variable/Ausdruck" konfigurierten Tag mit diesem Wert.

☞ Manche ActiveX-Controls senden automatisch eine Meldung an ihren Container (z.B. an die Runtime-Umgebung), um anzuzeigen, dass sich ein Wert geändert hat und dieser gelesen (Holen) werden kann.

Andere ActiveX-Controls unterstützen das Versenden von Meldungen bei Werteänderungen jedoch nicht.

In diesem Fall können die aktualisierten Werte der ActiveX-Eigenschaften nur durch kontinuierlichen Aufruf dieser Werte aus dem ActiveX-Steuerelemente erlangt werden (Scan=Immer).

Konfigurieren von Methoden

Die Registerkarte Methoden stellt eine Tabelle mit den folgenden Feldern zur Verfügung:

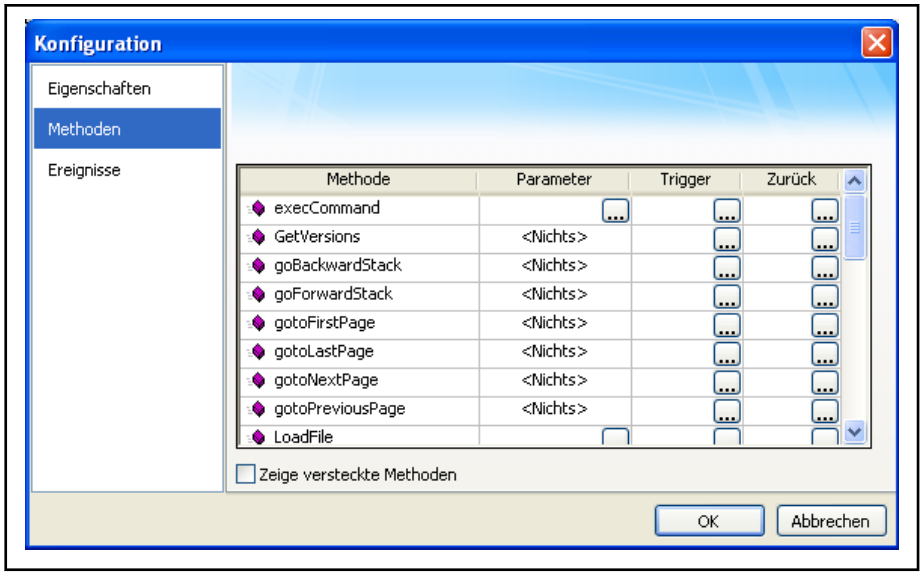


Abb.8-68: Konfigurationsdialog – Registerkarte Methoden

- Spalte **Methode**: Führt alle von diesem ActiveX-Objekt verfügbaren Methoden auf
- Spalte **Parameter**: Die in dieser Spalte konfigurierten Tags sind mit den Übergabeparametern einer Methode verknüpft.

  Wenn eine Methode keine Parameter unterstützt, wird der Text <Keine> im Parameterfeld angezeigt.

  Wenn die Methode mehr als einen Übergabeparameter erwartet, kann für jeden Parameter ein Tag, getrennt durch ein Komma (",") eingegeben werden.

  Zum Beispiel **TagA , TagB , TagC**.

  Bei Ausführung der Methode wird entweder der Wert der Tags in die Parameter der Methode geschrieben (Eingabeparameter) oder das ActiveX schreibt nach Ausführung der Methode den Wert der Parameter in die Tags (Ausgabeparameter).



Durch einen Klick auf die Schaltfläche Durchsuchen (...) wird eine Liste der von der Methode unterstützten Parameter angezeigt, die mit einem Tag verbunden werden können.

- Spalte **Trigger**: Wenn der in dieser Spalte konfigurierte Tag seinen Wert ändert, wird die entsprechende Methode des ActiveX-Controls ausgeführt
- Spalte **Zurück**: Der in dieser Spalte konfigurierte Tag erhält den Rückgabewert der Methode (sofern vorhanden)

Konfigurieren von Ereignissen

Die Registerkarte Ereignisse stellt eine Tabelle mit den folgenden Feldern zur Verfügung:

Bilder und Grafiken

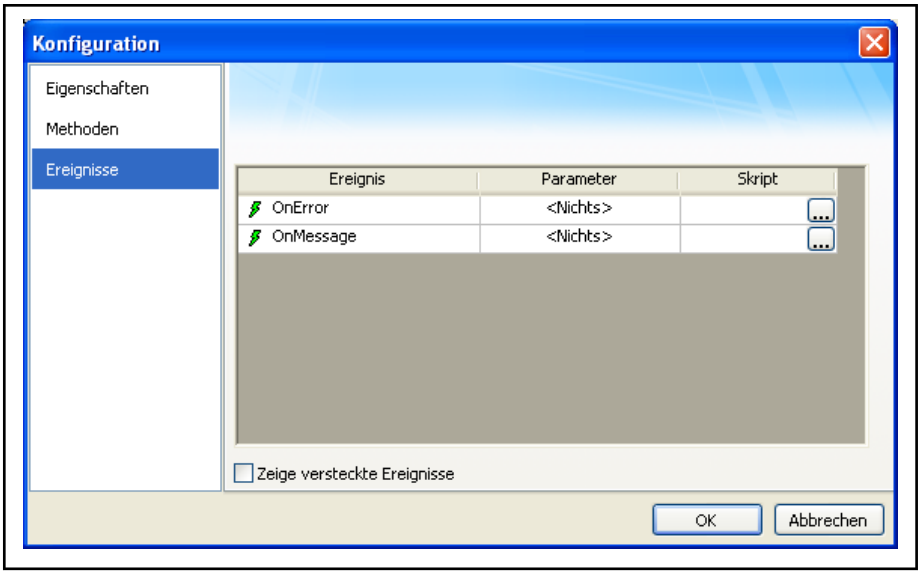


Abb.8-69: Konfigurationsdialog – Registerkarte Ereignisse

- Spalte **Ereignis**: Führt alle von diesem ActiveX-Objekt verfügbaren Ereignisse auf.
- Spalte **Parameter**: Die in dieser Spalte konfigurierten Tags sind mit den Übergabeparametern eines Ereignisses verknüpft.

  Wenn ein Ereignis keine Parameter unterstützt, wird der Text <Keine> im Parameterfeld angezeigt.

  Wenn ein Ereignis mehr als einen Übergabeparameter erwartet, kann für jeden Parameter ein Tag, getrennt durch ein Komma (",") eingegeben werden.

  Zum Beispiel **TagA , TagB , TagC**.

  Beim Auftreten des Ereignisses wird entweder der Wert der Tags in die Parameter des Ereignisses geschrieben (Eingabeparameter) oder die Parameterwerte werden in die Tags geschrieben (Ausgabeparameter).



Durch einen Klick auf die Schaltfläche Durchsuchen (...) wird eine Liste der von dem Ereignis unterstützten Parameter angezeigt, die mit einem Tag verbunden werden können.

- Spalte **Skript**: Das in diesem Feld konfigurierte Skript wird ausgeführt, wenn das Ereignis durch das ActiveX-Steuerelement ausgelöst wird.



Durch einen Klick auf die Schaltfläche Durchsuchen (...) wird der Programmeditor geöffnet. Das Programm kann in VBScript oder in der "Rexroth WinStudio" eigenen Programmsprache erstellt werden.

.NET-Controls

.NET-Komponenten werden auf der Basis des Microsoft .NET Frameworks entwickelt, das als Standard für modulare Programmiertechnologien dient. Da WinStudio ein .NET-Container ist, können Sie in Ihren WinStudio-Bildern .NET-Komponenten einbinden und ausführen. Die tatsächlichen Funktionen einer .NET-Komponente sind in einem **.NET-Control** enthalten, das die Konfigurationsdialoge zur Verfügung stellt.

.NET-Komponenten beinhalten die folgenden Schnittstellen:

- **Eigenschaften**: Variablen, deren Werte in Ihr Projekt gelesen und/oder geschrieben werden können (zum Beispiel Objektfarbe, Dateiname, URL usw.)
- **Methoden**: Funktionen aus der .NET-Komponente, die durch Ihr Projekt ausgelöst werden können (z.B. Dialog öffnen, Berechnung ausführen usw.)
- **Ereignisse**: Interne Meldungen, die das Ausführen von Ausdrücken in Ihrem Projekt auslösen können (z.B. Maus_Klick, Download_abgeschlossen usw.)

.NET Controls können sich in ihrer Darstellung und Funktion unterscheiden, je nachdem welche Eigenschaften, Methoden oder Ereignisse konfiguriert werden.

HINWEIS

Stellen Sie bei der Verwendung von .NET-Komponenten in Ihrem Projekt sicher, dass das Zielsystem (Laufzeitstation) die gleichen Komponenten unterstützen kann und dass diese korrekt installiert und registriert sind.

Auswählen und Platzieren eines .NET-Controls

Auswählen und Einfügen eines .NET-Controls in ein WinStudio-Bild:

1. Klicken Sie im Werkzeugfenster **WinStudio Screen Tools** auf **.NET Control**.
2. Wenn der Dialog ".NET-Framework-Komponenten" geöffnet wird (wie in der folgenden Abbildung), enthält er eine Liste aller .NET-Komponenten, die auf Ihrem Rechner registriert sind.

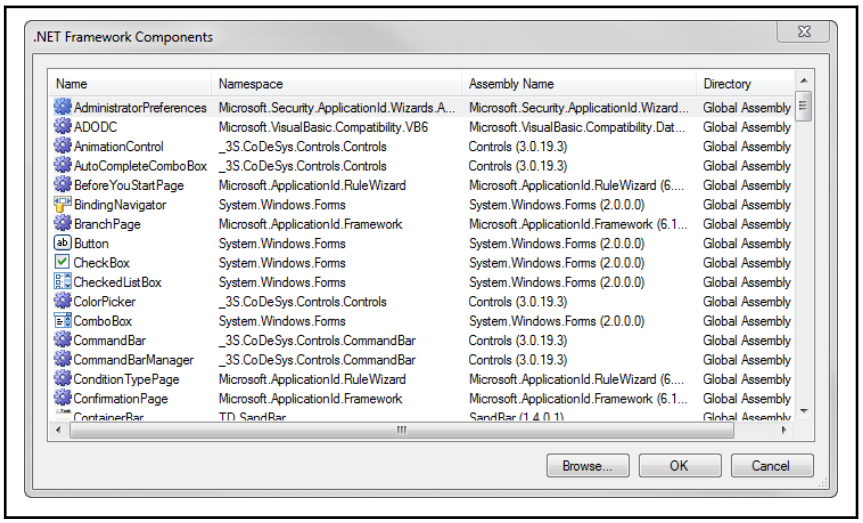


Abb.8-70: Dialog .NET-Framework-Komponenten

3. Wählen Sie eine Komponente aus der Liste und klicken Sie dann auf **OK**, um sie im Bild zu platzieren. Zum Suchen einer nicht-registrierten Komponente auf Ihren Rechner können Sie auch auf die Schaltfläche **Durchsuchen...** / **Browse...** klicken.

Registrierte .NET-Komponenten werden üblicherweise im folgenden Verzeichnis gespeichert:

C:\WINDOWS\Microsoft.NET\Framework

Nicht registrierte .NET Controls können der Auswahlliste im Dialog ".NET-Framework-Komponenten" hinzugefügt werden, indem folgender Eintrag in der Datei "projekt_name.APP" ergänzt wird:

[Execution Environment]

DotNetControlPath=OptionalPath

Zum Beispiel:

[Execution Environment]

DotNetControlPath=C:\DOTNET CONTROLS BACKUP

Danach führt der Dialog ".NET-Framework-Komponenten" alle registrierten Komponenten "und" alle im angegebenen Verzeichnis gefundenen Komponenten auf.

4. Standardmäßig wird ein neues .NET-Control in der oberen linken Ecke des Bildes platziert. Klicken Sie auf das Objekt und ziehen Sie es an die gewünschte Stelle.
5. Wenn das Objekt platziert ist, öffnen Sie mit einem Doppelklick dessen Dialog "Objekteigenschaften".

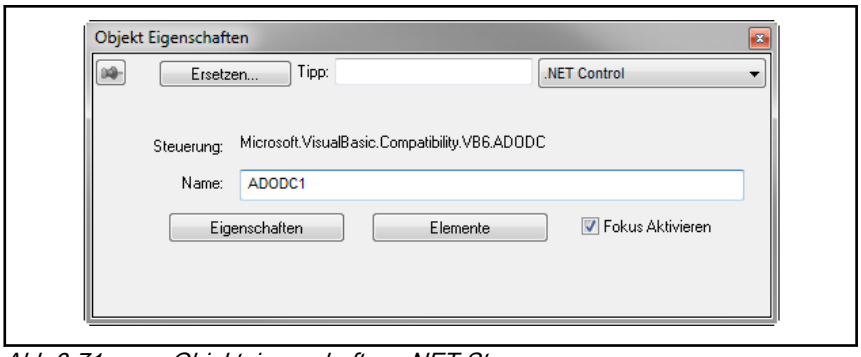


Abb.8-71: Objekteigenschaften: .NET Steuerung

Der Dialog "Objekteigenschaften" zeigt den Namen der .NET-Komponente an. Der Komponente muss im Feld **Name** ein eindeutiger Name (Alias) zugewiesen werden (z.B. Kontrollkästchen1). Dieser Name und wird bei der Verwendung der Skriptsprachen (VBScript auf Seite 797 und Integrierte Scriptsprache auf Seite 545) zum Referenzieren der Komponente verwendet.

Es dürfen niemals zwei .NET-Controls mit dem gleichen Namen im selben WinStudio-Bild konfiguriert werden.

Die Schaltfläche **Eigenschaften** öffnet das Standardfenster zum Konfigurieren der statischen Eigenschaften (falls vorhanden). Layout und Optionen in diesem Dialog sind abhängig von der Implementierung jeder .NET-Komponente. Verwenden Sie diese Schnittstelle zum Festlegen von Eigenschaften, die während der Laufzeit nicht geändert werden sollten (fest konfigurierte Eigenschaften).

Die Schaltfläche **Elemente** im Dialog "Objekteigenschaften" öffnet zusätzliche Dialoge für folgende Aufgaben:

- Tags mit Eigenschaften der .NET-Komponente verbinden
- Methoden aus der .NET-Komponente über eine Tag-Änderung auslösen

- Skripte konfigurieren, die beim Auftreten von Ereignissen aus der .NET-Komponente ausgeführt werden sollen.

Die folgenden Abschnitte beschreiben die Konfiguration dieser Schnittstellen.



Obwohl der Dialog "Elemente" eine Liste aller Eigenschaften, Methoden und Ereignisse anzeigt, müssen nur die Elemente konfiguriert werden, die Sie für Ihr Projekt benötigen.

Die im Folgenden verwendeten Abbildungen stellen das .NET Control "Kontrollkästchen" dar. Obwohl die Namen von Eigenschaften, Methoden und Ereignissen sich je nach .NET Control unterscheiden, ist die Konfigurationsschnittstelle für alle .NET-Komponenten gleich. Die hier beschriebenen Elemente (Eigenschaften, Methoden, Ereignisse) gelten für alle .NET Controls.

Konfigurieren von Eigenschaften

Die Registerkarte "Eigenschaften" stellt eine Tabelle mit den folgenden Feldern zur Verfügung:

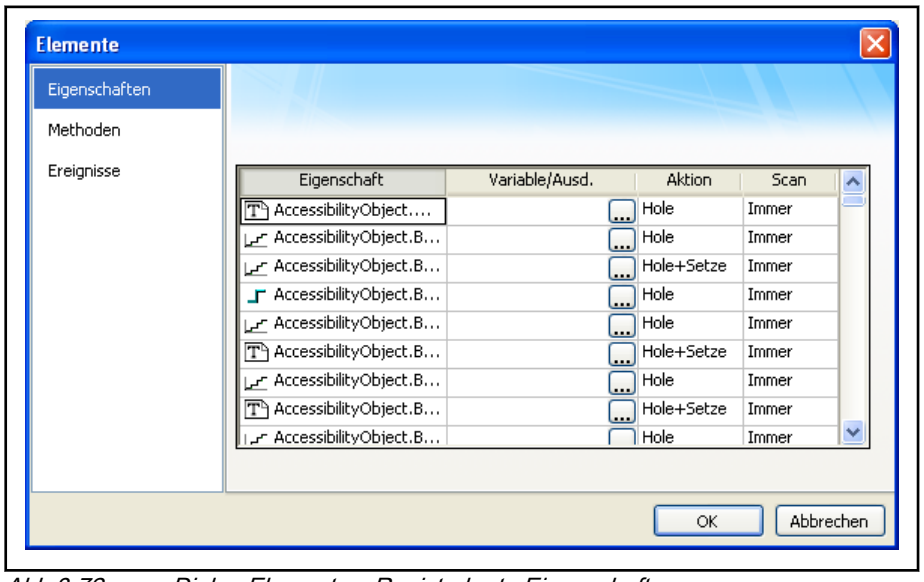


Abb.8-72: Dialog Elemente – Registerkarte Eigenschaften

- Spalte **Eigenschaft**: Führt alle von der .NET-Komponente verfügbaren Eigenschaften auf und gibt deren Typen an:

Symbol Eigenschaft	Typ
	Boolean
	Integer
	Real
	String

- Spalte **Variable/Ausdruck**: Der in diesem Feld konfigurierte Tag ist mit der entsprechenden Eigenschaft der .NET-Komponente verbunden. Die Spalte Aktion bestimmt, ob der Wert dieses Tags in die Eigenschaft geschrieben wird oder ob der Wert der Eigenschaft in diesen Tag geschrieben wird (oder beides).
- Spalte **Aktion**: In dieser Spalte wird festgelegt ob der Parameter in Spalte "Variable/Ausdruck" aus dem Control gelesen oder in das Control geschrieben wird, gemäß der folgenden Tabelle:

> Wenn der Wert der Eigenschaft schreibgeschützt ist (kann von Ihrem Projekt nicht überschrieben werden), so wird das Feld **Aktion** automatisch auf **Holen** gestellt.

Aktion	Beschreibung
Holen	Liest den Wert der Eigenschaft und schreibt ihn in den im Feld **Variable/Ausdruck** konfigurierten Tag.
Setzen	Schreibt den Wert aus dem im Feld **Variable/Ausdruck** konfigurierten Tag oder Ausdruck in die Eigenschaft.
Holen+Setzen	Führt beide Aktionen aus (**Holen** und **Setzen**). Aber: WinStudio führt beim Öffnen eines Bildes mit der .NET-Komponente den Befehl **Holen** vor dem Ausführen eventueller **Setzen**-Befehle aus, d. h. wenn "WinStudio" das Bild öffnet, in der die .NET-Komponente konfiguriert ist, wird der im Feld **Variable/Ausdruck** konfigurierte Tag mit dem Wert aus der Eigenschaft aktualisiert.
Setzen+Holen	Führt beide Aktionen aus (**Holen** und **Setzen**). Aber: WinStudio führt beim Öffnen eines Bildes mit der .NET-Komponente den Befehl **Setzen** vor dem Ausführen eventueller **Holen**-Befehle aus, d. h. wenn "WinStudio" das Bild öffnet, in der die .NET-Komponente konfiguriert ist, wird die Eigenschaft mit dem Wert des im Feld **Variable/Ausdruck** konfigurierten Tags aktualisiert.

- Spalte **Scan**: Definiert die Aufrufmethode zum Holen von Werten aus den Eigenschaften. Für .NET-Komponenten führen alle Eigenschaften den Scan standardmäßig **Immer** aus. "WinStudio" ruft kontinuierlich den Wert der Eigenschaft ab und aktualisiert den im Feld **Tag/Ausdruck** konfigurierten Tag mit diesem Wert.

Konfigurieren von Methoden

Die Registerkarte "Methoden" stellt eine Tabelle mit den folgenden Feldern zur Verfügung:

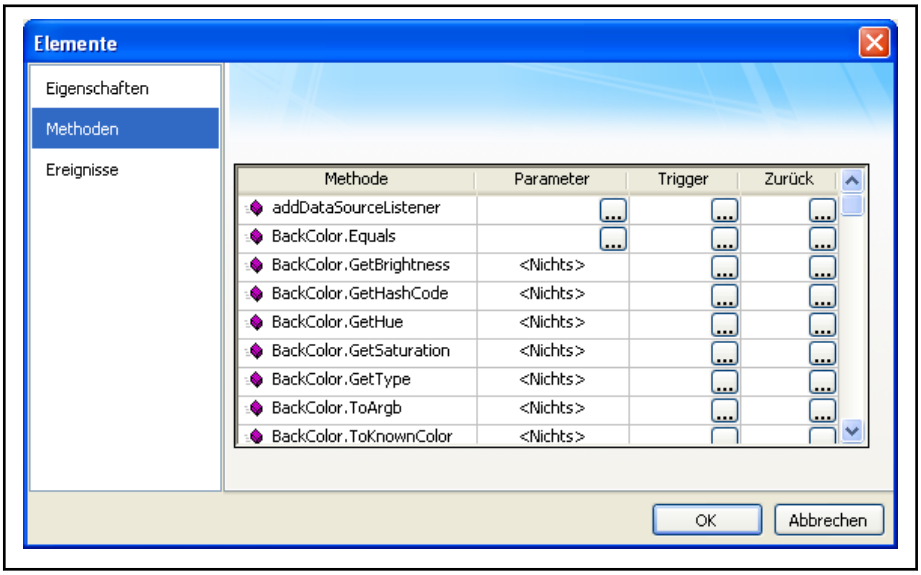


Abb.8-73: Dialog Elemente – Registerkarte Methoden

- Spalte **Methode**: Führt alle von dieser .NET-Komponente verfügbaren Methoden auf.
- Spalte **Parameter**: Die in dieser Spalte konfigurierten Tags sind mit den Übergabeparametern einer Methode verknüpft. Wenn eine Methode keine Übergabeparameter unterstützt, wird der Text <Keine> im Parameterfeld angezeigt. Wenn eine Methode mehr als einen Übergabepara-

meter erwartet, kann für jeden Parameter ein Tag, getrennt durch ein Komma (",") eingegeben werden.

Zum Beispiel **TagA , TagB , TagC**.

Durch einen Klick auf die Schaltfläche Durchsuchen (...) wird eine Liste der von der Methode unterstützten Parameter angezeigt, die mit einem Tag verbunden werden können.

Bei Ausführung der Methode wird entweder der Wert der Tags in die Übergabe-Parameter der Methode geschrieben (Eingabeparameter) oder das .NET-Controls schreibt nach Ausführung der Methode den Wert der Übergabe-Parameter in die Tags (Ausgabeparameter).

- Spalte **Trigger**: Wenn der in dieser Spalte konfigurierte Tag seinen Wert ändert, wird die entsprechende Methode des .NET-Controls ausgeführt.
- Spalte **Zurück**: Der in diesem Feld konfigurierte Tag erhält den Rückgabewert der Methode (sofern vorhanden)

Konfigurieren von Ereignissen

Die Registerkarte "Ereignisse" stellt eine Tabelle mit den folgenden Feldern zur Verfügung:

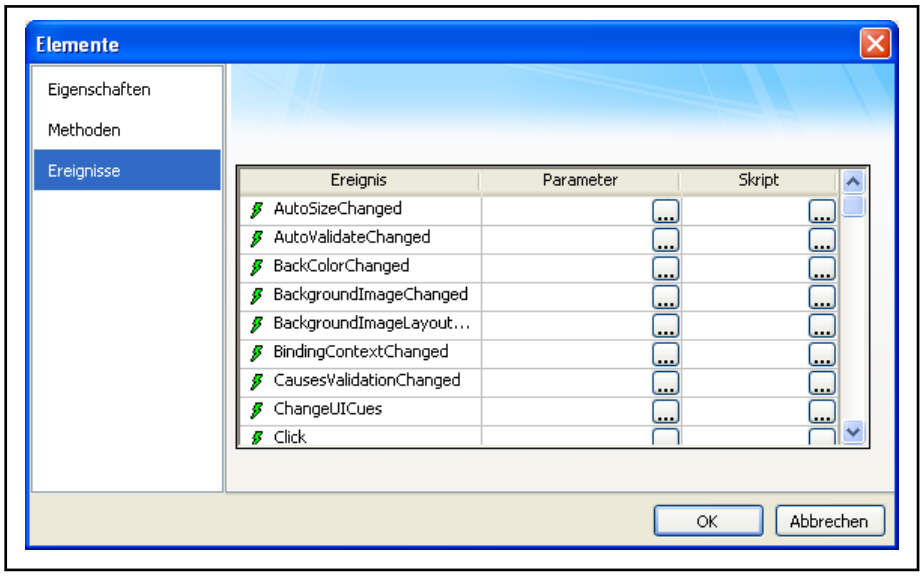


Abb.8-74: Dialog Elemente – Registerkarte Ereignisse

- Spalte **Ereignis**: Führt alle von diesem .NET-Control verfügbaren Ereignisse auf.
- Spalte **Parameter**: Die in dieser Spalte konfigurierten Tags sind mit den Übergabeparametern der Ereignisse verknüpft. Wenn ein Ereignis keine Übergabeparameter unterstützt, wird der Text <Keine> im Parameterfeld angezeigt. Wenn ein Ereignis mehr als einen Übergabeparameter erwartet, kann für jeden Parameter ein Tag, getrennt durch ein Komma (",") eingegeben werden.

  Zum Beispiel **TagA , TagB , TagC**.



Durch einen Klick auf die Schaltfläche Durchsuchen (...) wird eine Liste der von dem Ereignis unterstützten Parameter angezeigt, die mit einem Tag verbunden werden können.

Bei Auftreten des Ereignisses wird entweder der Wert der Tags in die Übergabe-Parameter der Methode geschrieben (Eingabeparameter)

oder das .NET Control schreibt nach dem Auftreten des Ereignisses den Wert der Übergabe-Parameter in die Tags (Ausgabeparameter).

- Spalte **Skript**: Das in diesem Feld konfigurierte Skript wird ausgeführt, wenn das Ereignis durch das .NET-Control ausgelöst wird.



Durch einen Klick auf die Schaltfläche Durchsuchen (...) wird der Programmeditor geöffnet. Das Programm kann in VBScript oder in der "Rexroth WinStudio" eigenen Programmsprache erstellt werden.

Hinzufügen einer Verknüpfung zu einer externen Grafikdatei

Mit dem Bibliotheksobjekt "Grafik" können externe Grafikdateien als Verknüpfung in ein WinStudio Bild eingefügt werden. Dadurch ist es möglich während der Laufzeitumgebung das Bild auszutauschen.

Die Grafikdatei kann im Projekt oder an beliebiger Stelle gespeichert sein:

- Wenn die Grafikdatei mit den restlichen Projektdateien auf das Zielsystem herunter geladen werden soll, muss sie im Projektordner gespeichert werden, z.B.

  `[…]\Projekt1\VEP40\user\config\`

- Wenn die Datei an anderer Stelle auf dem Zielsystem oder im Netzwerk gespeichert ist, wird automatisch der absolute Pfad der Verknüpfung gespeichert



Wenn Sie anstelle einer Verknüpfung zu einer externen Datei die Grafik direkt in das WinStudio-Bild einbinden möchten und wenn das Bild während der Laufzeit nicht geändert werden muss, kann statt dessen das Bildobjekt Bitmap auf Seite 189 verwendet werden.

1. Im Werkzeugfenster **WinStudio Screen Tools** klicken Sie auf **Grafik**.

   Ein Dialog "Öffnen" wird angezeigt

2. Verwenden Sie den Dialog zum suchen und auswählen der Grafikdatei und klicken Sie dann auf **OK**.

   Die Grafik wird dem WinStudio-Bild als Verknüpfung hinzugefügt

3. Führen Sie einen Doppelklick auf das Grafikobjekt aus.

   Der Dialog "Objekteigenschaften: Grafikverknüpfung" wird angezeigt

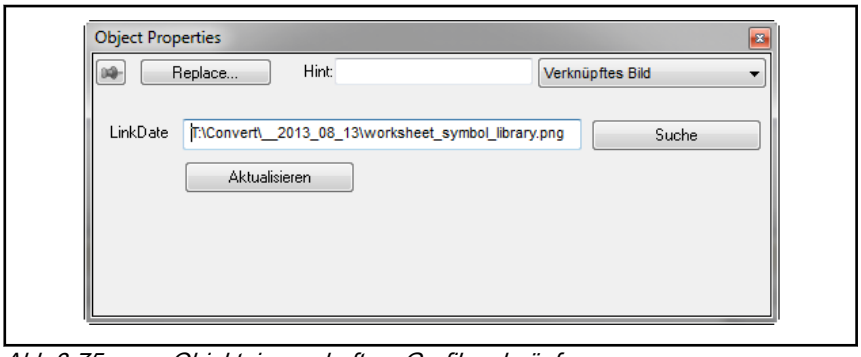


Abb.8-75: Objekteigenschaften: Grafikverknüpfung

4. Prüfen Sie im Feld **Dateiverknüpfung** die Verknüpfung zur Grafikdatei.

   Wenn die Grafikdatei im Projektordner gespeichert ist, dann ist die Verknüpfung relativ zu diesem Ordner. Wenn die Grafikdatei an anderer

Stelle gespeichert ist, dann ist die Verknüpfung ein vollständiger Dateipfad

5. Wenn Sie die Verknüpfung während der Projektlaufzeit ändern möchten, geben Sie als Verknüpfung einem String-Tag (z.B.. **{tagname}**) ein. Die Verknüpfung wird dann während der Projektlaufzeit bei jeder Wertänderung des String-Tags aktualisiert.

   Der Tag-Wert muss das gleiche Format haben wie eine normale Verknüpfung: ein relativer Dateipfad für eine im Projektordner gespeicherte Datei oder ein absoluter Dateipfad für eine Datei, die an anderer Stelle auf dem Zielsystem oder im Netzwerk gespeichert ist
6. Schließen Sie den Dialog "Objekteigenschaften"

8.3.5 Anwenden von Animationen auf Bildobjekte

Anwenden von Animationen auf Bildobjekte, Allgemeines

Verwenden Sie die Symbolleiste "WinStudio-Dynamikfunktionen" um Animationen auf ein Bildobjekt oder eine Objektgruppe anzuwenden. Mit Animationen können Objekteigenschaften während der Laufzeit mit Hilfe von Tags verändert werden. Mit einigen Animationen können auch Befehle ausgeführt oder Werte (Sollwerte) in die Tags eingefügt werden.



Abb.8-76: Symbolleiste "WinStudio-Dynamikfunktionen"

Ein Dynamikobjekt wird nur dann aktiv (farbig) dargestellt, wenn das angewählte Bildobjekt bzw. Bildgruppe diese Animation unterstützt (z.B. Textein-/-ausgabe für Textfelder). Wenn man die Maus über ein aktives WinStudio-Dynamikobjekt bewegt, wird die Funktion im Klartext angezeigt.

Befehls-Animation

Symbol: 

Klicken Sie in der Symbolleiste **WinStudio-Dynamikfunktionen** auf **Befehl** , um die Animationen zu einem ausgewählten Objekt oder einer Objektgruppe hinzuzufügen. Diese Dynamikfunktion ermöglicht das Auslösen einer Aktion bzw. das Ausführen von Programmfunktionen per Mausklick während der Laufzeit. Mit einem Doppelklick auf das Objekt werden die Objekteigenschaften angezeigt.

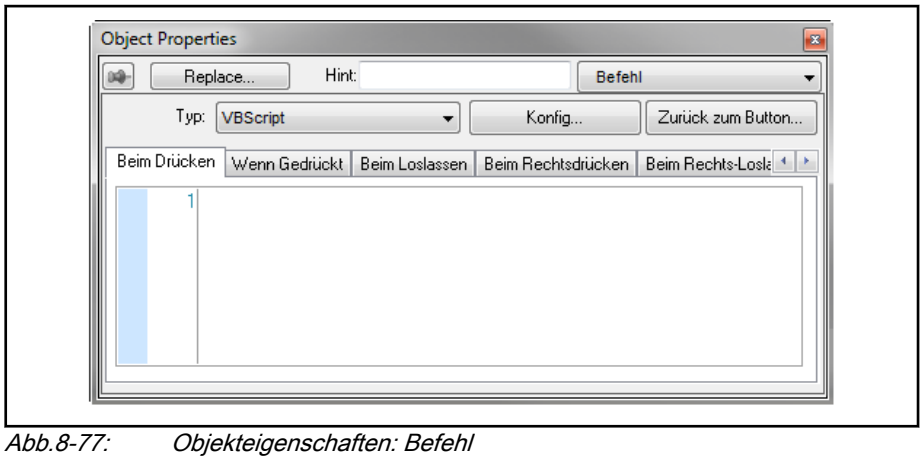


Abb.8-77: Objekteigenschaften: Befehl

Die Konfigurationsdialog stellt für jedes vom Objekt unterstützte Ereignis eine eigene Registerkarte zur Verfügung. Bitte beachten Sie, dass für die gleiche Dynamikfunktion mehr als ein Ereignis gleichzeitig konfiguriert werden kann:

- **Beim Drücken**: Führt den Befehl bzw. das Skript einmal aus, wenn der Benutzer das Objekt mit der linken Maustaste anklickt.
- **Wenn gedrückt**: Für den Befehl bzw. das Skript kontinuierlich aus, während die Maustaste gedrückt ist. Die Ausführungsdauer (in Millisekunden) für den Befehl bzw. das Skript wird im Feld "Rate" festgelegt. Eine Ausnahme bildet die Option VBScript, die schnellstmöglich ausgeführt wird.
- **Beim Loslassen**: Führt den Befehl bzw. das Skript einmal aus, wenn der Benutzer das Objekt mit der linken Maustaste loslässt.
- **Beim Rechtsdrücken**: Führt den Befehl bzw. das Skript einmal aus, wenn der Benutzer das Objekt mit der rechten Maustaste anklickt.
- **Beim Rechts-Loslassen**: Führt den Befehl bzw. das Skript einmal aus, wenn der Benutzer das Objekt mit der rechten Maustaste loslässt.
- **Bei Doppelklick**: Führt den Befehl bzw. das Skript einmal aus, wenn der Benutzer auf das Objekt einen Doppelklick mit der linken Maustaste ausführt .



- Das Laufzeitprojekt behandelt Touchscreen-Aktionen genau wie Mauszeigeroptionen.
- Die Ereignisse "Beim Rechtsdrücken", "Beim Rechts-Loslassen" und "Bei Doppelklick" werden in Projekten, die auf Zielsystemen mit Windows CE ausgeführt werden, nicht unterstützt
- Beim Erstellen eines Projekts für ein Touchscreen-Gerät muss beachtet werden, dass Ereignisse "Beim Rechtsdrücken" und "Beim Rechts-Loslassen" auf solchen Geräten nicht ausgelöst werden können.

- Auswahlfeld **Typ**: Diese Einstellung bestimmt die Art der Aktion, die beim Auftreten des Ereignisses ausgeführt werden soll. Beachten Sie, dass für jedes Ereignis (z.B. beim Drücken, beim Loslassen) eine eigene Typ-Auswahl getroffen werden kann. Dadurch kann die gleiche Dynamikfunktion für verschiedene Ereignisse auf unterschiedliche Arten konfiguriert werden. Es werden folgende Typen unterstützt:

Type	Beschreibung
Built-in Language	Ermöglicht das Konfigurieren eines Skripts mit der eingebauten Sprache von "WinStudio". Wenn dieser Typ ausgewählt wird, kann der Benutzer für jedes Ereignis in der Spalte **Ausdruck** bis zu zwölf Ausdrücke konfigurieren. Wenn das Ereignis ausgelöst wird, werden die Ausdrücke nacheinander ab der ersten bis zur letzten Zeile ausgeführt. Das Ergebnis jedes Ausdrucks wird in den ggf. in der Spalte **Variable** konfigurierten Tag geschrieben. Weitere Informationen finden in Anhang: WinStudio-interne Skriptsprache auf Seite 545
VBScript	Ermöglicht das Konfigurieren eines Skripts mit der Standardsprache VBScript. Wenn dieser Typ ausgewählt wird, kann der Benutzer im VBScript-Editor ein Skript für die **Befehlsanimation** konfigurieren. Weitere Informationen zur VBScript-Sprache finden Sie in Übersicht VBScript auf Seite 797
Bild öffnen	Ermöglicht das Konfigurieren der **Befehlsanimation** zum Öffnen einer bestimmtes Bildes, wenn das Ereignis während der Laufzeit ausgelöst wird. Dieser Typ ist gleichwertig mit der Funktion Öffnen auf Seite 604. Sie können entweder den Bildnamen in das Feld **Bild öffnen** eingeben oder ein Bild über den Browse-Dialog auswählen. Außerdem kann in diesem Feld ein String-Tag in geschweiften Klammern {TagName} eingegeben werden. Wenn das Ereignis ausgeführt wird, versucht das Projekt, das genannte Bild zu öffnen.

Type	Beschreibung
Bild schließen	Ermöglicht das Konfigurieren der **Befehlsanimation** zum Schließen eines bestimmten Bildes, wenn das Ereignis während der Laufzeit ausgelöst wird. Dieser Typ ist gleichwertig mit der Funktion Schließen auf Seite 603. Sie können entweder den Bildnamen in das Feld **Bild schließen** eingeben oder ein Bild über den Browse-Dialog auswählen. Außerdem kann in diesem Feld ein String-Tag in geschweiften Klammern {TagName} eingegeben werden. Wenn das Ereignis ausgeführt wird, versucht das Projekt, das genannte Bild zu schließen.
Variable setzen	Ermöglicht das Konfigurieren der **Befehlsanimation** zum Setzen einer Variablen, wenn das Ereignis während der Laufzeit ausgelöst wird. Sie können entweder den Tag-Namen in das Feld **Variable setzen** eingeben oder ein Tag über den Browse-Dialog auswählen. Wenn das Ereignis ausgeführt wird, schreibt das Projekt den Wert 1 in die konfigurierte Variable.
Variable zurücksetzen	Ermöglicht das Konfigurieren der **Befehlsanimation** zum Zurücksetzen einer Variablen, wenn das Ereignis während der Laufzeit ausgelöst wird. Sie können entweder den Tag-Namen in das Feld **Variable zurücksetzen** eingeben oder ein Tag über den Browse-Dialog auswählen. Wenn das Ereignis ausgeführt wird, schreibt das Projekt den Wert 0 in die konfigurierte Variable.
Variable toggeln	Ermöglicht das Konfigurieren der **Befehlsanimation** zum Umschalten einer Variablen, wenn das Ereignis während der Laufzeit ausgelöst wird. Sie können entweder den Tag-Namen in das Feld **Variable umschalten** eingeben oder ein Tag über den Browse-Dialog auswählen. Wenn das Ereignis ausgeführt wird, schaltet das Projekt den Wert des in diesem Feld konfigurierten Tags um.

- Schaltfläche **Konfig**: Öffnet eine erweiterte Ansicht der Objekteigenschaften mit zusätzlichen Konfigurationseinstellungen.
- **Zurück zu Schaltfläche**: Klicken, um zu den Objekteigenschaften des zugrundeliegenden Schaltflächenobjekts zurückzukehren.

Dialog Konfiguration

Erweiterter Objekteigenschaftsdialog mit zusätzlichen Konfigurationseinstellungen...

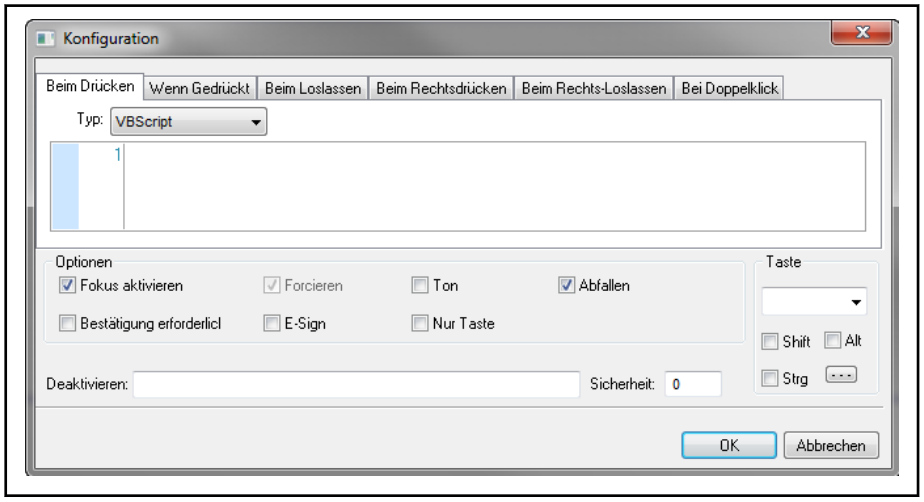


Abb.8-78: Dialog Konfiguration

Die Ereignis-Registerkarten (z.B. "Beim Drücken", "Während" usw.) und die Auswahlliste **Typ** sind die gleichen wie im Dialog "Objekteigenschaften". Die zusätzlichen Konfigurationseinstellungen sind allen Ereignissen gemeinsam:

- Bereich "Optionen":
  - **Fokus aktivieren**: Wenn diese Option angewählt ist, kann das Objekt, auf das die Befehlsanimationen angewendet wurde, während der Laufzeit den Eingabefokus erhalten und dadurch bei Navigation mit der <Tab>-Taste selektiert werden

Bilder und Grafiken

- **Forcieren**: Wenn diese Option aktiviert ist, lösen alle Wertzuweisungen zu Variablen Änderungsereignisse aus, auch wenn der Wert des entsprechenden Tags sich nicht ändert. Wenn zum Beispiel ein Tag den Wert 0 hat und die Befehlsanimation den gleichen Wert 0 in diesen Tag schreibt, erkennen alle anderen Tasks im Laufzeitprojekt nach Ausführung der Animation, dass dieser Tag eine Änderung erfahren hat (auch wenn dies nicht der Fall war). Diese Option ist hilfreich, um sicherzustellen, dass durch Tag-Änderungen veranlasste Aktionen (z.B. Schreiben bei Tag-Änderung von einem Kommunikationstreiber) nach dem Ausführen der Befehlsanimation ausgelöst werden.

  Bei mit "Rexroth WinStudio v7.x" oder früheren Versionen erstellten Projekten ist die Option **Forcieren** standardmäßig aktiviert und kann nicht deaktiviert werden.
- **Ton**: Wenn diese Option aktiviert ist, wird nach dem Ausführen des Befehls ein kurzer Piep-Ton ausgegeben. Diese Option ist hilfreich, um dem Benutzer durch eine akustische Rückmeldung anzuzeigen, dass der Befehl ausgeführt wurde. Er zeigt allerdings nicht an, ob die durch die Befehlsanimation ausgelöste Aktion erfolgreich war oder nicht.
- **Abfallen**: Wenn diese Option aktiviert ist, wird das Ereignis „Beim Loslassen“ ausgeführt, wenn der Cursor (oder der Finger) aus dem Objektbereich gezogen wird (egal, ob die Schaltfläche gelöst wurde oder nicht). Diese Option ist hilfreich, um sicherzustellen, dass nach einem Ereignis „Beim Drücken“ immer das Ereignis „Beim Loslassen“ ausgeführt wird, auch wenn der Benutzer die Maus vor dem Lösen der Taste aus dem Objektbereich zieht.
- **Bestätig erforderlich**: Wenn diese Option aktiviert ist, muss der Benutzer vor dem Ausführen des Befehls eine Bestätigungsabfrage quittieren. Diese Option ist hilfreich zur Verringerung der unbeabsichtigten Auslösevorgänge von kritischen Ereignissen während der Laufzeit.
- **E-Sign**: Wenn diese Option angewählt ist, wird der Benutzer vor dem Ausführen des Befehls zur Eingabe der elektronischen Signatur aufgefordert
- **Nur Taste**: Wenn diese Option aktiviert ist, kann der Benutzer Befehle "nur" mit Tastaturkürzeln ausführen (im unten beschriebenen Bereich "Taste" konfiguriert)

- **Deaktivieren**: Deaktiviert die Benutzeraktion, wenn das Ergebnis des in diesem Feld konfigurierten Ausdrucks WAHR ist (Wert ungleich 0)
- **Sicherheit**: Sicherheit auf Seite 407 benötigte Zugangsstufe die ein Bediener mindestens haben muss, um zur Laufzeit die Befehlsanimation ausführen zu dürfen
- Bereich **Taste**: Tastaturkürzel für das Auslösen der Ereignisse "Beim Drücken", "Während gedrückt" und "Beim Loslassen" mit Hilfe einer Tastatur. (Die Verwendung dieses Tastaturkürzel entspricht einem Klick mit der linken Maustaste.) Diese Option ist besonders hilfreich, wenn die Tastatur während der Laufzeit die einzige verfügbare physische Schnittstelle zur Interaktion mit der Bedienoberfläche ist.
  - **Umschalten**, **Strg** oder **Alt**: Klicken zum Erstellen einer Tastaturkombination, d.h. die Taste Umschalten, Strg bzw. Alt müssen zusammen mit der in der Auswahlliste festgelegten Taste gedrückt werden.

– Öffnen Sie den Dialog "Tasten ändern" durch Klicken auf die Such-Schaltfläche (...), um Ihre Tastaturkombinationen zu verändern. Für Tastaturkombinationen kann die Position der Taste Umschalt, Strg oder Alt auf der Tastatur mit **Links**, **Rechts** oder **Links oder rechts** festgelegt werden. Bei Auswahl von **Links oder rechts** wird der Befehl jedes Mal ausgeführt, wenn eine dieser Tasten in Verbindung mit der in der Auswahlliste festgelegten Taste gedrückt wird.

Hyperlink-Animation

Symbol: 

Klicken Sie in der Symbolleiste **WinStudio Dynamikfunktionen** auf **Hyperlink**, um einen Hyperlink zu einem ausgewählten Objekt oder einer Objektgruppe hinzuzufügen.

Hyperlinks ermöglichen dem Anwender während der Laufzeit einen Standardbrowser mit einer vorgegebenen URL zu öffnen.

Mit einem Doppelklick auf das Objekt öffnet sich der Dialog "Objekteigenschaften".

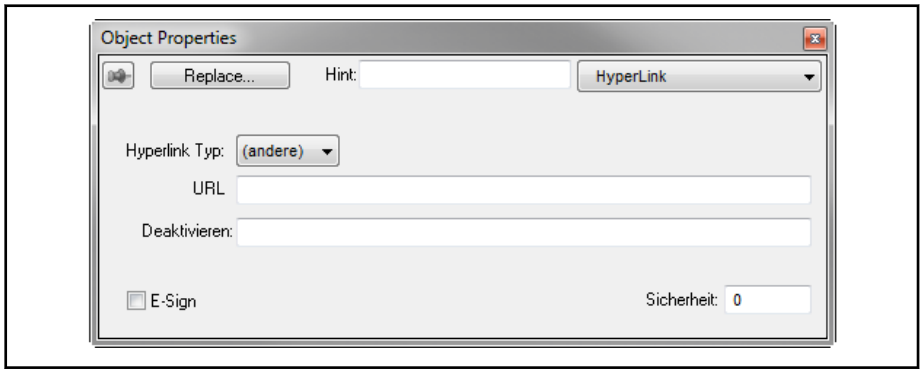


Abb.8-79: Objekteigenschaften: Hyperlink

Dieser Dialog wird zur Festlegung der folgenden Parameter verwendet:

- **Hyperlink-Typ**: Wählen Sie aus der Auswahlliste ein URL-Protokoll aus. Das Projekt verwendet dieses Protokoll beim Laden einer URL..
- **URL**: Geben Sie die URL-Adresse ein, die geladen werden soll.

Es ist nicht erforderlich, den Protokolltyp im Feld URL einzugeben.

Wenn ein Protokolltyp aus der Liste **Hyperlink-Typ** ausgewählt wird, fügt das Projekt der URL-Adresse automatisch den Protokoll-Präfix hinzu.

- **Deaktivieren**: Durch Eingabe eines Werts größer Null in diesem Feld wird die Hyperlink-Befehlsanimation für das/die ausgewählte/n Objekt(e) deaktiviert.
- **E-Sign**: Wenn diese Option angewählt ist, wird der Benutzer vor dem Ausführen der Animation zur Eingabe der elektronischen Signatur aufgefordert.
- **Sicherheit**: Geben Sie in diesem Feld einen Wert zur Festlegung einer Sicherheitsstufe für das Objekt bzw. die Objekte ein, wie unter Projektsicherheit auf Seite 407 definiert. Wenn ein Benutzer sich anmeldet, aber

nicht über die erforderliche Sicherheitsstufe verfügt, deaktiviert das Projekt den Hyperlink-Befehl für das Objekt bzw. die Objekte.

Balkendiagramm-Animation

Symbol: 

Klicken Sie in der Symbolleiste **WinStudio Dynamikfunktionen** auf **Balkendiagramm**, um einem ausgewählten Objekt Balkendiagramm-Eigenschaften hinzuzufügen. Führen Sie dann einen Doppelklick auf das Objekt aus, um den Dialog "Objekteigenschaften" zu öffnen.



Abb.8-80: Objekteigenschaften: Balkendiagramm

Im Dialog "Objekteigenschaften" werden die folgenden Parameter festgelegt:

- **Variable/Ausdruck**: Geben Sie einen Tag oder Ausdruck an, der als Balkenhöhe bzw -Breite angezeigt werden soll. Sie können auch durch Klicken auf die Schaltfläche  nach einem bestehenden Tag oder Ausdruck durchsuchen.
- **Minimaler Wert**: Geben Sie in diesem Feld eine numerische Konstante oder einen Tag-Wert ein, um den Mindestwert für die Berechnung der Höhe (für vertikaler Balken) oder Breite (für horizontale Balken) der Balken festzulegen.
- **Maximaler Wert**: Geben Sie in diesem Feld eine numerische Konstante oder einen Tag-Wert ein, um den Maximalwert für die Berechnung der Höhe (für vertikaler Balken) oder Breite (für horizontale Balken) der Balken festzulegen.

  Wenn in diesem Feld kein Wert eingegeben wird, öffnet die Anwendung einen Dialog zur Auswahl bzw. zum Anlegen eines Tags.



"WinStudio" erlaubt auch die Eingabe von Konstanten in Feldern mit Tags bzw. numerischen Werten. Konstante Werte (gekennzeichnet durch das #-Zeichen) sind gleichwertig mit numerischen Werten. Die Textkonstante kann sowohl als Beschreibungstext als auch zur Festlegung von generischen Werten genutzt werden.

Zum Beispiel: **#Name:100**.

Wobei der Wert ("100") nach dem Doppelpunkt (":") der generische Wert und "Name" nur ein Mnemonik für die Konstante ist und nicht zur Datenbank hinzugefügt wird.

- **Vordergrundfarbe:** Zum Festlegen einer Füllfarbe des Balkens klicken Sie auf die Combobox.

- **Richtung**: Klicken Sie auf das Optionsfeld **Vertikal** oder **Horizontal** zum Festlegen der Richtung des Balkendiagramms.
- **Orientierung**: Klicken Sie auf das Optionsfeld **Oben**, **Zentriert** oder **Unten** zum Festlegen der Lage der Maximal- und Minimalwerte beim Zeichnen des Balkens.

Animation Text-Daten-Verknüpfung

Symbol: 

Klicken Sie in der Symbolleiste **WinStudio Dynamikfunktionen** auf **Text-Daten-Verknüpfung**, um die Eigenschaften für Eingabe- oder Ausgabetext der Animation zu einem ausgewählten Textobjekt, Seite190 hinzuzufügen.

Die Dynamikfunktion "Text-Daten-Verknüpfung" ermöglicht die Darstellung und Änderung von Tag-Werten über die Bedienoberfläche während der Laufzeit.

☞ Diese Animation kann nur auf Textobjekte angewendet werden, die ein oder mehrere **#**-Zeichen enthalten.

Jedes **#** steht für ein Eingabe- bzw. Ausgabezeichen. **#**-Zeichen können mit regulärem Text im gleichen Textobjekt kombiniert werden – zum Beispiel **MeineBezeichnung#####** oder **$###.##**.

Es muss unbedingt beachtet werden, dass das Laufzeitprojekt immer die höchstwertigen Ziffern eines numerischen Werts angezeigt, ungeachtet der Zahl oder Platzierung der **#**-Zeichen im Text. Das bedeutet, wenn nicht genügend **#**-Zeichen zum Anzeigen des Werts vorhanden sind, wird dieser je nach Format des Werts umgewandelt (wie durch die unten beschriebene Option **Fmt** festgelegt):

- Beim Dezimalformat wird die Anzahl der Dezimalstellen durch die Position des Dezimaltrennzeichens im **###**-Text bestimmt. Wenn aber auf der linken Seite des Dezimaltrennzeichens nicht genügend **#**-Zeichen zum Anzeigen des gesamten Werts vorhanden sind, wird der Dezimalwert durch die Ganzzahl überschrieben.

  Wenn zum Beispiel versucht wird, den Wert 112,64 in **#,##** anzuzeigen, wird "112" ausgegeben.
- Wenn bei Hex- und Binärformaten mehr **#**-Zeichen vorhanden sind, als zum Anzeigen des Werts erforderlich, füllt das Laufzeitprojekt den Wert mit führenden Nullen auf. Wenn weniger Zeichen als benötigt vorhanden sind, wird der Wert einfach verkürzt.
- Bei Auto-Format ignoriert das Laufzeitprojekt die Anzahl der **#**-Zeichen und zeigt den gesamten numerischen oder String-Wert an.

  Dezimalwerte werden mit allen Vor-und Nachkommastellen angezeigt, ungeachtet der Platzierung des Dezimaltrennzeichens im **###**-Text. Bei außerordentlich großen Werten oder langen Zeichenketten kann dies das Layout Ihrer Bilder stören.

Mit einem Doppelklick auf das Objekt öffnet sich der Dialog "Objekteigenschaften". Dieser Dialog wird zur Festlegung der folgenden Parameter verwendet:



Abb.8-81: Objekteigenschaften: Text-Daten-Verknüpfung

- **Variable/Ausdruck**

  Geben Sie in das Textfeld folgendes ein:

  – Name eines Tags, dessen Wert angezeigt oder eingegeben werden soll

  – Ausdruck, dessen Werte angezeigt werden soll

  Sie können auch mit der Schaltfläche  die "Objektsuche" öffnen, um einen bestehenden Tag oder Ausdruck auszuwählen.

☞ Wenn der konfigurierte Tag bzw. Ausdruck ungültig ist, werden während der Laufzeit Platzhalterzeichen (**###**) angezeigt.

- **Format**: Auswahl, wie ein numerischer Wert des angegebenen Tags oder Ausdrucks formatiert und auf dem Bildschirm angezeigt wird. Die verfügbaren Optionen sind u.a. **Dezimal**, **Hexadezimal**, **Binär** und **Auto**. Bei Auswahl von **Auto** wird der Wert gemäß der von der Funktion DezimalzeichenSetzen auf Seite 681 erstellten virtuelle Tabelle formatiert. Die Formateinstellung ändert nicht das Ablageformat des dargestellten Tags oder Ausdrucks.

  Zum Beispiel: **Variable/Ausdruck** ist auf einen Tag vom Typ Integer auf Seite 124 gesetzt, **Eingabe freigegeben** ist aktiviert und **Format** ist auf **Hexadezimal** eingestellt. Es kann ein neuer Wert in Hexadezimalformat eingegeben werden, er wird aber in Ihrer Projektdatenbank als Integer abgespeichert
- **Eingabe aktivieren**: Diese Option anklicken (aktivieren), um Anwendereingaben für den angegebenen Tag zu erlauben. Diese Option abwählen (deaktivieren), wenn der Wert zur Laufzeit nur angezeigt, aber nicht geändert werden soll.
- **Zurück zum Text**: Klicken, um zu den Objekteigenschaften des zugrundeliegenden Textobjekts zurückzukehren
- **Minimaler Wert**: Geben Sie einen Mindestwert für den mit diesem Textobjekt verbundenen Tag ein. Der Benutzer kann keine niedrigeren Werte eingeben
- **Maximaler Wert**: Geben Sie einen Maximalwert für den mit diesem Textobjekt verbundenen Tag ein. Der Benutzer kann keine größere Werte eingeben

- **Passwort**: Diese Option anklicken (aktivieren), wenn das Eingabefeld z.B. für eine Passworteingabe verwendet wird und deshalb alle Zeichen durch ein " *" ersetzt werden sollen
- **Bestätigung erforderlich**: Diese Option anklicken (aktivieren), damit der Benutzer während der Laufzeit alle Werteingaben bestätigen muss
- **Autom. Größe**: Diese Option anklicken (aktivieren), um die Größe des Textobjekts automatisch an die Textlänge anzupassen. Diese Option ist nicht verfügbar, wenn **Eingabe aktiviert** angewählt ist (siehe oben)
- **E-Sign**: Wenn diese Option angewählt ist, wird der Benutzer vor dem Ändern des Tag-Werts zur Eingabe der elektronischen Signatur aufgefordert
- **Virtuelle Tastatur**: Für dieses Objekt wird bei der Eingabe von Werten eine virtuelle Tastatur verwendet. Vor dem Konfigurieren der Virtuellen Tastatur für diese Schnittstelle muss die Option "Virtuelle Tastatur" im Register "Viewer" in den Eigenschaften des Ordners "WinStudio" aktiviert werden
- **Deaktivieren**: Geben Sie in diesem Feld einen Wert größer null ein, um die Eingabefunktion zu deaktivieren
- **Sicherheit** Geben Sie in diesem Feld einen Wert zur Festlegung der Sicherheitsstufe ein, die ein Bediener mindestens haben muss, um zur Laufzeit einen Wert editieren zu dürfen. (siehe auch Sicherheit auf Seite 407)

Farb-Animation

Symbol: 

Klicken Sie in der Symbolleiste **WinStudio Dynamikfunktionen** auf **Farbumschlag**, um die Animation zu einem ausgewählten Objekt hinzuzufügen.

Die Farbanimation ermöglicht das Ändern der Farbe eines Objekts während der Laufzeit, auf Grundlage des Werts eines Tags bzw. Ausdrucks.

Mit einem Doppelklick auf das Objekt öffnet sich der Dialog Objekteigenschaften.



Abb.8-82: Objekteigenschaften: Farben

Dieser Dialog wird zur Festlegung der folgenden Parameter verwendet:

- **Typ**: Legt den Modus der Farmumschaltung fest:
  - **durch Grenze**: Bei Auswahl dieses Typs können bis zu vier Grenzwerte (Änderungslimit) sowie eine Farbe für jeden Grenzwert festgelegt werden. Wenn der Wert des im Feld Tag/Ausdruck konfigurierten Tags bzw. Ausdrucks den Grenzwert erreicht, wird die für

den entsprechenden Grenzwert festgelegte Farbe auf das Objekt angewendet.

– **durch Farbe**: Bei Auswahl dieses Typs ein beliebiger Farbcode, welcher dargestellt werden soll, einem Tag im Feld Tag/Ausdruck übergeben werden. Mit diesem Code kann jede von ihrem Gerät unterstützte Farbe auf das Objekt angewendet werden.

Sie können die Funktion RGB-Farbe auf Seite 679 im Feld Variable/Ausdruck konfigurieren, wenn unter **Typ** der Eintrag "durch Farbe" ausgewählt ist. Dadurch kann die Farbe gemäß den RGB-Codes konfiguriert werden. Unter Farbschnittstelle auf Seite 265 finden Sie eine Tabelle mit den Codes für häufig verwendete Farben.

- **Variable/Ausdruck**: Geben Sie den Namen des zu überwachenden Tags bzw. Ausdrucks ein. Wenn für die Einstellung Typ der Eintrag "durch Grenzwert" ausgewählt ist, vergleicht "WinStudio" das Ergebnis des Tags bzw. Ausdrucks mit den festgelegten Grenzwerten, um die korrekte Farbe für das gewählte Objekt zu bestimmen. Wenn "durch Farbe" ausgewählt ist, bestimmt der Wert des hier parametrierten Tags die Farbe des Objekts (Farbcode).
- **Änderungsgrenzwert**: Geben Sie einen Grenzwert (numerische Konstante oder Tag) für den Farbwechsel an. Die Zahlen müssen in aufsteigender Reihenfolge konfiguriert werden (von oben nach unten). Durch Klicken auf die Schaltfläche "Weitere Farben" können bis zu 16 verschiedene Grenzwerte für die Farbanimation konfiguriert werden.

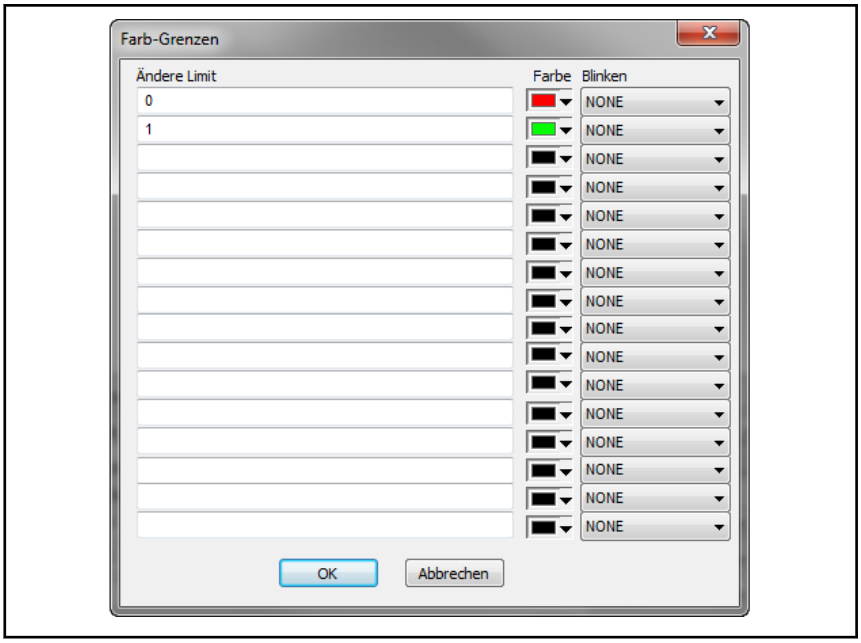


Abb.8-83: Dialog Farbgrenzwerte

- **Farbe**: Klicken Sie auf die Schaltfläche, um jedem Grenzwert für Farbänderung eine Farbe zuzuweisen.
- **Blinken**: Wählen Sie in der Auswahlliste, ob die Farbe bei Erreichen des Grenzwertes blinkt, und wie schnell.

Die folgenden Felder werden automatisch deaktiviert (grau hinterlegt), wenn für **Typ** der Eintrag "durch Farbe" gewählt ist: **Änderungsgrenzwert**, **Farbe** und **Blinken**.

Positions-Animation

Symbol: 

Mit der Positions-Animation kann ein Objekt während der Laufzeit horizontal und/oder vertikal verschoben werden.

Klicken Sie in der Symbolleiste **WinStudio Dynamikfunktionen** auf **Position**, um die Animation zu einem Objekt hinzuzufügen.

Mit einem Doppelklick auf das Objekt öffnet sich der Dialog "Objekteigenschaften".

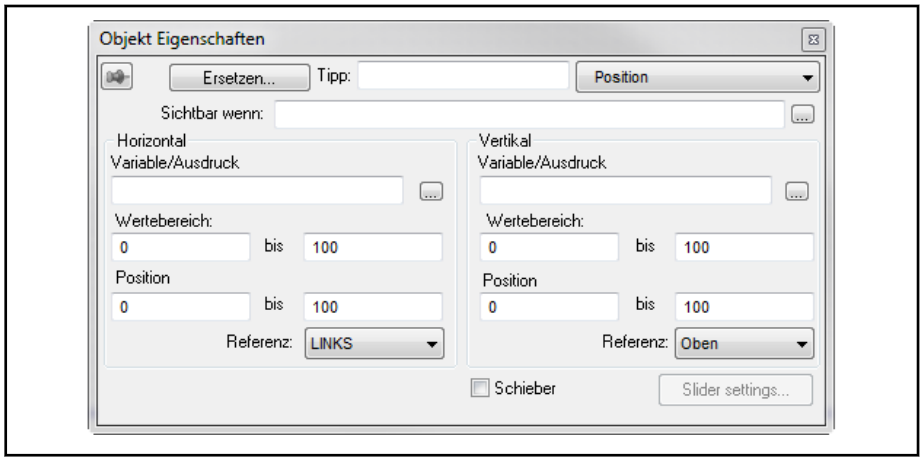


Abb.8-84: Objekteigenschaften: Position

Konfigurieren Sie mit Hilfe dieses Dialogs die folgenden Eigenschaften:

- **Sichtbar wenn**: Konfigurieren Sie in diesem Feld einen Tag bzw. Ausdruck um festzulegen, wann das Objekt angezeigt wird. Wenn das Feld leer ist oder der Tag bzw. Ausdruck WAHR ist (d.h. alle Werte ungleich null), wird das Objekt angezeigt. Wenn der Wert FALSCH ist (d.h. 0), wird das Objekt nicht angezeigt. Das Feld kann also leer gelassen werden, um das Objekt immer anzuzeigen.
- **Schieber**: Diese Option anklicken (aktivieren), um das Objekt wie einen Schieber zu bedienen, d.h. das Objekt kann über das Bild gezogen werden, um Werte an die konfigurierten Tags übergeben. (Siehe "Bewegen" weiter unten.) Klicken Sie auf die Schaltfläche **Schieber-Einstellungen** zum Konfigurieren zusätzlicher Einstellungen (Aktivierung /Deaktivierung und Zugriffsrechte).
- Bereiche "Horizontal" und "Vertikal" : Durch Konfigurieren dieser Einstellungen wird bestimmt, wie das Objekt sich innerhalb des WinStudio-Bildes bewegt:
  - **Variable** / **Ausdruck**: Die Bedeutung dieser Felder ist abhängig davon, ob die Option **Schieber** aktiviert ist (siehe oben):
    - Wenn die Option **Schieber** nicht aktiviert ist, wird in diesem Feld entweder einen Tag "oder" ein Ausdruck konfiguriert. Der aktuelle Wert des Tags bzw. Ausdrucks bestimmt die aktuelle Position des Objekts. Wenn sich der Wert verändert, wird das Objekt im Bild bewegt.

– Wenn die Option **Schieber** aktiviert ist, wird in diesem Feld nur ein Tag vom Typ Integer oder Real zugelassen. Der Tag erhält den Wert, der der aktuellen Position des Objekts entspricht. Wenn der Benutzer das Objekt bewegt, ändert sich der Wert.

Der Positionswert für eine "horizontale" Verschiebung erhöht sich, wenn das Objekt nach rechts bewegt wird und verringert sich, wenn das Objekt nach links bewegt wird. Der Positionswert für eine "vertikale" Verschiebung erhöht sich, wenn das Objekt nach unten bewegt wird und verringert sich, wenn das Objekt nach oben bewegt wird.

– **Wertebereich**: Geben Sie den Minimal- und Maximalwert für den Tag bzw. Ausdruck ein. Wenn der tatsächliche Wert diesen Bereich verlässt, wird der Wert ignoriert und statt dessen der Grenzwert verwendet.

– **Position**: Geben Sie Werte ein, um festzulegen, wie weit (in Pixel) sich das Objekt von seiner Ausgangsposition entfernen kann. Die Ausgangsposition entspricht „0,0“. Bei Werten größer null kann das Objekt nach rechts und unten bewegt werden, bei Werten kleiner null kann das Objekt nach links und oben bewegt werden. Während der Laufzeit ist die Position des Objekts proportional zu dem Wert des Tags bzw. Ausdrucks in seinem Gültigkeitsbereich. Wenn zum Beispiel die **Position** "0" bis "100" ist und der **Bereich** "0" bis "10" ist, dann bewegt jedes Inkrement in dem Wert das Objekt um zehn Pixel. Dies gilt für horizontale und vertikale Bewegungen.

– **Referenz**

  Wählen Sie einen Bezugspunkt auf dem Objekt. Die folgende Tabelle zeigt, wie die von Ihnen ausgewählten Werte für "Horizontal" und "Vertikal" zusammenwirken:

	LINKS	MITTE	RECHTS
OBEN			
MITTE			
UNTEN			

Der **Bezugspunkt** ist nur dann von Bedeutung, wenn dem gleichen Objekt die Größenanimation auf Seite 254 hinzugefügt wurde. Die Position des Objekts basiert immer auf diesen Bezugspunkt, ungeachtet der Größe oder Form des Objekts.

Animation Größenänderung

Symbol: 

Die Animation "Größenänderung" ermöglicht das Vergrößern oder Verkleinern eines Objekts während der Laufzeit.

Klicken Sie in der Symbolleiste **WinStudio-Dynamikfunktionen** auf **Größenänderung**, um die Animation zu einem Objekt hinzuzufügen.

Mit einem Doppelklick auf das Objekt öffnet sich der Dialog "Objekteigenschaften".

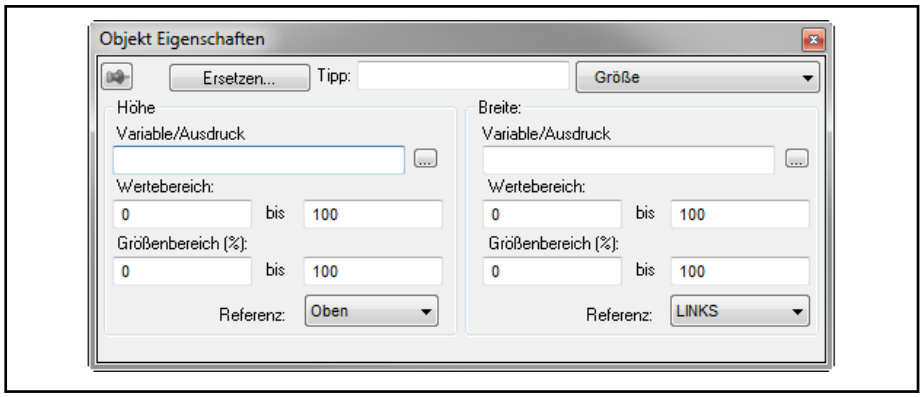
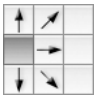
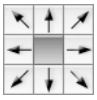
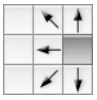
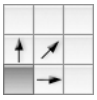
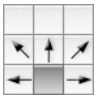
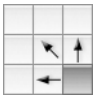


Abb.8-85: Objekteigenschaften: Größe

Konfigurieren Sie mit Hilfe dieses Dialogs die folgenden Eigenschaften:

- **Variable/Ausdruck**: Geben Sie die Tags zur Bestimmung von **Höhe** und **Breite** des Objekts ein. Lassen Sie eines der Felder leer, wenn Sie keine Größenänderung in dieser Dimension wünschen
- **Wertebereich**: Geben Sie den Minimal- und Maximalwert für den/die festgelegten Tag(s) ein. Wenn der tatsächliche Wert eines Tags diesen Bereich verlässt, wird der Wert ignoriert und statt dessen der Grenzwert verwendet
- **Größenbereich (%)**: Geben Sie den Minimal- und Maximalwert für die Objektgröße ein. Der Mindestwert kann bis zu 0% betragen (wodurch das Objekt effektiv unsichtbar wird) und der Maximalwert kann beliebig groß sein. Die Originalgröße des Objekts beträgt 100%, wenn es im Bild-Editor gezeichnet wird. 200% wäre also die doppelte Originalgröße usw.

  Während der Laufzeit ist die Größe des Objekts proportional zu dem Wert des Tags in seinem Gültigkeitsbereich. Wenn zum Beispiel die **Größe (%)** "0" bis "100" ist und der **Bereich** "0" bis "10" ist, dann vergrößert jedes Inkrement das Objekt um 10 %. Dies gilt für Höhe und Breite
- **Referenz**: Wählen Sie einen Bezugspunkt zur Bestimmung der Richtungen für die Größenänderung des Objekts. Die folgende Tabelle zeigt, wie die von Ihnen ausgewählten Werte für Höhe und Breite zusammenwirken:

	LINKS	MITTE	RECHTS
OBEN			
MITTE			
UNTEN			

Bilder und Grafiken

Animation Drehen

Symbol: 

Klicken Sie in der Symbolleiste **WinStudio-Dynamikfunktionen** auf **Drehung**, um die Animation zu einem Objekt Linie auf Seite 182, Offenes Polygon auf Seite 183, Geschlossenes Polygon auf Seite 184 oder Bitmap auf Seite 189 hinzuzufügen.

Mit einem Doppelklick auf das Objekt öffnet sich der Dialog "Objekteigenschaften".

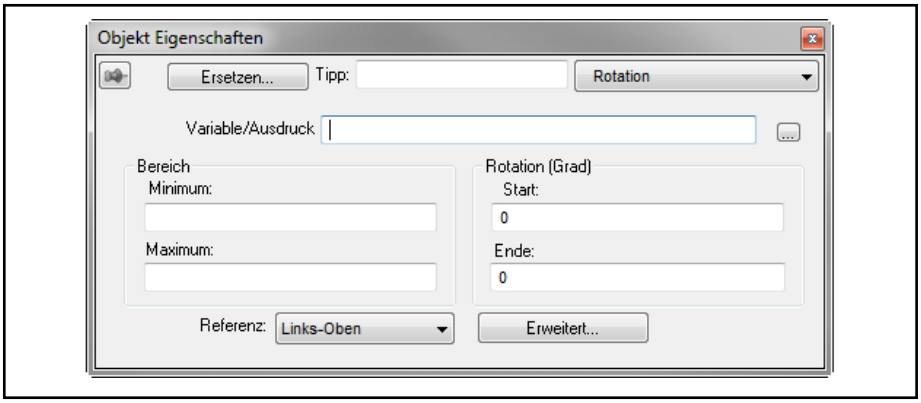


Abb.8-86: Objekteigenschaften: Drehungsanimation

Dieser Dialog wird zur Festlegung der folgenden Parameter verwendet.

- **Variable/Ausdruck**: Geben Sie einen Tag-Namen oder Ausdruck zum Aktivieren der Drehungsanimation ein. Der Wert aus **Variable/Ausdruck** bestimmt die tatsächliche Drehung das Objekt; der Grad der Drehung ändert sich mit dem Wert.
- "Wertebereich": Geben Sie die zulässigen **Minimal-** und **Maximal-**Werte für **Variable/Ausdruck** ein. Werte unter dem Minimum und über dem Maximum werden ignoriert.
- "Rotation (Grad)": Geben Sie die **Start-** und **End-**Positionen des Objekts ein (in Grad). Die tatsächliche Drehung ist proportional zu dem Wert von **Variable/Ausdruck** im "Wertebereich". Ein Objekt kann bis zu 360 Grad gedreht werden und dreht standardmäßig im Uhrzeigersinn.



Eine Drehungsanimation hat zum Beispiel die folgenden Einstellungen: **Minimum** ist 0, **Maximum** ist 100, **Start** ist 0 und **Ende** ist 180. Wenn der aktuelle Wert von **Variable/Ausdruck** 50 beträgt (d.h. in der Mitte zwischen Minimum und Maximum), dann beträgt die tatsächliche Drehung des Objekts 90 Grad (d.h. in der Mitte zwischen Start und Ende). Ein Wert von 25 entspricht 45°, ein Wert von 75 entspricht 135° usw.

- **Referenz**: Wählen Sie einen der folgenden Punkte als Drehpunkt aus, um den sich das Objekt dreht:
  - **Links oben**: Obere linke Ecke des Objekts.
  - **Links unten**: Untere linke Ecke des Objekts.
  - **Zentriert**: Mittelpunkt des Objekts.
  - **Rechts oben**: Obere rechte Ecke des Objekts.
  - **Rechts unten**: Untere rechte Ecke des Objekts.

Der Drehpunkt kann über einen Offset (Button Erweitert) wie unten beschrieben genau positioniert werden.

- **Erweitert**: Klicken Sie die Schaltfläche an, um den Dialog für "Weitere Einstellungen" zu öffnen, in denen die folgenden Einstellungen konfiguriert werden können:

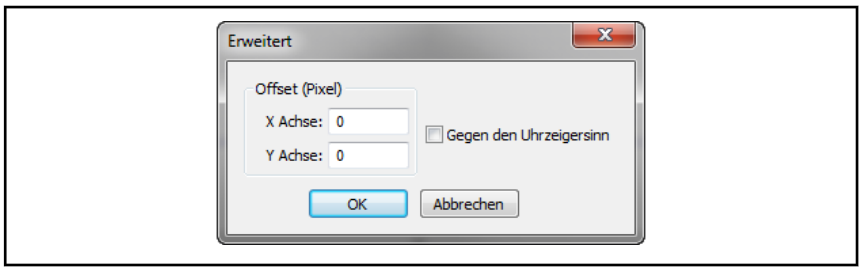


Abb.8-87: Objekteigenschaften: Drehungsanimation – Dialog Weitere Einstellungen

  - "Offset (Pixel)": Geben Sie die Anzahl der Pixel ein, um die die **Referenz** (d.h. der Drehpunkt) auf der **X-Achse** und/oder **Y-Achse** verschoben werden soll.
  - **Gegen den Uhrzeigersinn**: Diese Option anklicken (aktivieren), um das Objekt gegen den Uhrzeigersinn zu drehen.

8.4 Formatierung von Bildobjekten

8.4.1 Formatierung von Bildobjekten, Allgemeines

Die Symbolleisten "WinStudio Ausrichtungsfunktionen" und "WinStudio Objekteigenschaften" bieten zahlreiche Funktionen zum Ausrichten, Gruppieren und zur Gestaltung von Bildobjekten an.



Abb.8-88: Symbolleiste "Ausrichtungsfunktionen"



Abb.8-89: Symbolleiste "Objekteigenschaften"



Einige Funktionen in den Symbolleisten sind nur verfügbar, wenn ein oder mehrere Objekte im WinStudio-Bild ausgewählt wurden.

Wenn man die Maus über ein aktives "Werkzeug" bewegt, wird die Funktion im Klartext angezeigt.

Die Werkzeuge sind in die folgenden Gruppen gegliedert:

- **Anordnen**: Anordnen von Objekten in einem WinStudio-Bild, einschließlich
  - In den Vordergrund und In den Hintergrund auf Seite 258
  - Nach Vorne und Nach Hinten, eine Ebene auf Seite 259
  - Objekte gruppieren und Gruppierung aufheben auf Seite 259
  - Ausrichten, Zentrieren und Verteilen auf Seite 260
  - Drehen und Spiegeln auf Seite 262
- Position anpassen auf Seite 263
- **Größe**: Gleiche Breite / Gleiche Höhe für ausgewählte Objekte auf Seite 263 eines Bildobjekts in einem WinStudio-Bild

- **Stil**: Ändern der Farbgebung von Bildobjekten
  - – Füllfarbe auf Seite 264 eines Bildobjekts
  - – Linienfarbe auf Seite 265 eines Bildobjekts
- **Farbe einer Komponente bearbeiten**: Farbschnittstelle auf Seite 265
- **Schriftarten**: Ändern der Schriftarten auf Seite 269 eines Bildobjekts.

8.4.2 "In den Vordergrund" und "In den Hintergrund"

"WinStudio" weist jedem Bildobjekt eine eindeutige Identifikationsnummer (ID-Nr.) zu. Diese ID-Nummern beginnen immer mit null und gehen bis zur Gesamtzahl der Objekte in einem WinStudio-Bild.

Durch Klicken auf ein Objekt wird dessen ID-Nummer in der Statuszeile auf Seite 96 angezeigt.

"WinStudio" verwendet ID-Nummern, um zu bestimmen, ob ein Objekt vor oder hinter einem anderen Bildobjekt angezeigt wird. Objekte mit niedrigeren ID-Nummern werden hinter Objekten mit höheren ID-Nummern angezeigt.

Mit den folgenden Werkzeugen für Objektebenen können ausgewählte Objekte vor oder hinter andere Bildobjekte gebracht werden.

Diese Werkzeuge können nur mit einem einzelnen, markierten Objekt oder einer Gruppe auf Seite 259 von Objekten, nicht aber auf mehrere ausgewählte Objekte angewendet werden. Wenn mehrere Einzelobjekte ausgewählt und vor oder hinter ein anderes Objekt bewegt werden, dann behalten die ausgewählten Objekte ihre ursprüngliche Anzeigereihenfolge bei.

Objekt in den Hintergrund blenden

Klicken Sie in der Symbolleiste **WinStudio-Ausrichtungsfunktionen** auf  , **Objekt in den Hintergrund blenden**, um ein oder mehrere markierte Objekte hinter alle anderen Objekte im Bild zu verschieben. "WinStudio" weist dem Objekt die niedrigste ID-Nummer zu und bewegt das Objekt hinter alle anderen Objekte im Bild.

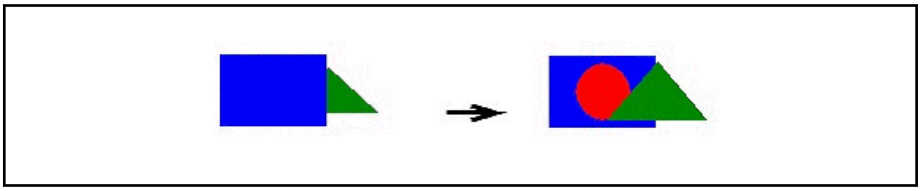


Abb.8-90: Objekte in den Hintergrund verschieben

Alternativ können Sie einen Rechtsklick auf das Objekt ausführen und **In den Hintergrund senden** aus dem Kontextmenü des Objekts auswählen.

Objekt in den Vordergrund blenden

Klicken Sie auf das Werkzeug  , **Objekt in den Vordergrund blenden**, um ein oder mehrere markierte Objekte vor alle anderen Objekte im Bild zu verschieben. "WinStudio" weist dem Objekt die höchste ID-Nummer zu und bewegt das Objekt vor alle anderen Objekte im Bild.

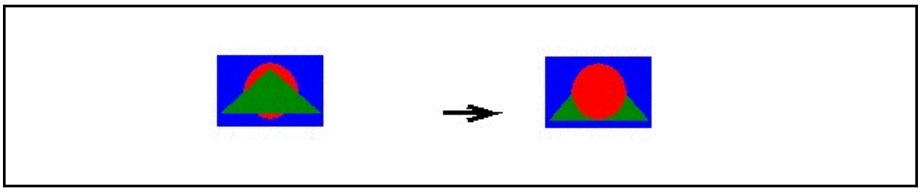


Abb.8-91: Objekte in den Vordergrund verschieben

Alternativ können Sie einen Rechtsklick auf das Objekt ausführen und **In den Vordergrund bringen** aus dem Kontextmenü des Objekts auswählen.

8.4.3 "Nach Vorne" und "Nach Hinten", eine Ebene

"WinStudio" weist jedem Bildobjekt eine eindeutige Identifikationsnummer (ID-Nr.) zu. Diese ID-Nummern beginnen immer mit null und gehen bis zur Gesamtzahl der Objekte im Bild. Durch Klicken auf ein Objekt wird dessen ID-Nummer in der Statuszeile auf Seite 96 angezeigt.

"WinStudio" verwendet ID-Nummern, um zu bestimmen, ob ein Objekt vor oder hinter einem anderen Objekt im Bild angezeigt wird. Objekte mit niedrigeren ID-Nummern werden hinter Objekten mit höheren ID-Nummern angezeigt. Mit den folgenden Werkzeugen für Objektebenen können ausgewählte Objekte vor oder hinter andere Bildobjekte gebracht werden.

☞ Diese Werkzeuge können nur mit einem einzelnen, markierten Objekt oder einer Gruppe auf Seite 259 von Objekten verwendet werden. Diese Werkzeuge können nicht auf mehrere ausgewählte Objekte angewendet werden. Wenn mehrere Einzelobjekte ausgewählt und vor oder hinter ein anderes Objekt bewegt werden, dann behalten die ausgewählten Objekte ihre ursprüngliche Anzeigereihenfolge bei.

Öffnen Sie das Kontextmenü eines gewählten Objekts und klicken Sie auf den Eintrag **Nach Hinten**, um das markierte Objekt bzw. die Gruppe um eine Ebene hinter das nächste Objekt im Bild zu verschieben. "WinStudio" weist dem markierten Objekt die nächstkleinere verfügbare ID-Nummer nach dem Objekt zu, hinter das es bewegt wurde.

Öffnen Sie das Kontextmenü eines gewählten Objekts und klicken Sie auf den Eintrag **Nach Vorne**, um das markierte Objekt bzw. die Gruppe um eine Ebene vor das nächste Objekt im Bild zu verschieben. "WinStudio" weist dem markierten Objekt die nächsthöhere verfügbare ID-Nummer nach dem Objekt zu, vor das es bewegt wurde.

8.4.4 "Objekte gruppieren" und "Gruppierung aufheben"

Symbole der Symbolleiste "Ausrichtungsfunktionen":

Verwenden Sie folgenden Werkzeuge, um zwei oder mehr markierte Objekte zu gruppieren bzw. um die Gruppierung aufzuheben.

☞ Alle Objekte mit Animationen auf Seite 243 und Gruppen aus Symbolobjekten (was die meisten Symbole und Bibliotheksobjekte einschließt) verfügen über mehrfache Objekt- und Dynamikeigenschaften.

Über die Auswahlliste im Dialog "Objekteigenschaften" kann auf diese verschiedenen Dialoge und Eigenschaften zugegriffen werden.

Objekte gruppieren

Klicken Sie in der Symbolleiste **WinStudio-Ausrichtungsfunktionen** auf  , **Objekte gruppieren**, um mehrere markierte Objekte zu einem einzigen Objekt zusammenzufassen, was die Handhabung von Objekten vereinfacht. (Im Dialog "Objekteigenschaften" kann auf jeden Teil der Gruppe zugegriffen werden.)

☞ Alternativ können Sie einen Rechtsklick auf eines der markierten Objekte ausführen und **Gruppieren** aus dem Kontextmenü auswählen.

Bilder und Grafiken

Gruppierung aufheben

Klicken Sie in der Symbolleiste **WinStudio-Ausrichtungsfunktionen** auf , **Gruppierung aufheben** um ein gruppiertes Objekt in seine einzelnen Bestandteile aufzuteilen.

Alternativ können Sie einen Rechtsklick auf das Objekt ausführen und **Degruppieren** aus dem Kontextmenü des Objekts auswählen.

Ein komplexes gruppiertes Objekt kann aus mehreren Gruppen gruppierter Objekte bestehen (bezeichnet als "Untergruppen"). Dementsprechend kann es erforderlich sein, zum vollständigen Aufheben der Gruppierung eines komplexen Objekts die Gruppierungen für alle Untergruppen aufzuheben.

8.4.5 "Ausrichten", "Zentrieren" und "Verteilen"

Wenn eine Reihe von Objekten (zwei oder mehr) markiert werden, können diese Objekte aufgrund der Position des zuletzt markierten Objekts ausgerichtet werden. Das zuletzt ausgewählte Objekt erhält daher eine spezielle Markierung (ausgefülltes Markierungssymbol).

In allen Abbildungen ist das Rechteck das zuletzt markierte Objekt.

Verwenden Sie die folgenden Werkzeuge aus der Symbolleiste **WinStudio-Ausrichtungsfunktionen** um eine Reihe von Objekten auszurichten.

Ein Beispiel ist in der jeweils folgenden Abbildung dargestellt:

Linksbündig

Durch Klicken auf das Werkzeug , **Linksbündig**, werden alle markierten Objekte an der linken Kante des zuletzt markierten Objekts ausgerichtet.

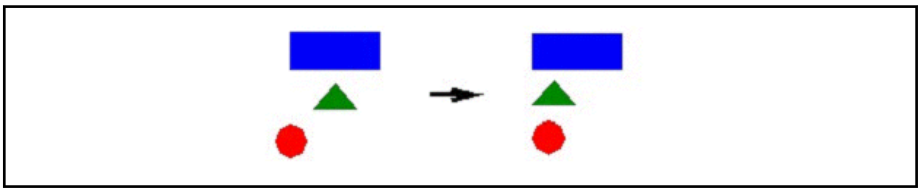


Abb.8-92: Objekte linksbündig ausrichten

Rechtsbündig

Durch Klicken auf das Werkzeug , **Rechtsbündig**, werden alle markierten Objekte an der rechten Kante des zuletzt markierten Objekts ausgerichtet.

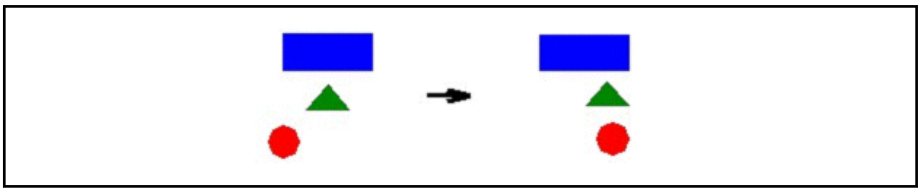


Abb.8-93: Objekte rechtsbündig ausrichten

Oben ausrichten

Durch Klicken auf das Werkzeug , **Oben ausrichten**, werden alle markierten Objekte an der Oberkante des zuletzt markierten Objekts ausgerichtet.

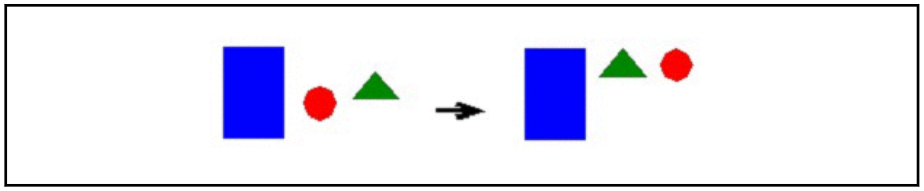


Abb.8-94: Objekte oben ausrichten

Unten ausrichten

Durch Klicken auf das Werkzeug , **Unten ausrichten**, werden alle markierten Objekte an der Unterkante des zuletzt markierten Objekts ausgerichtet.

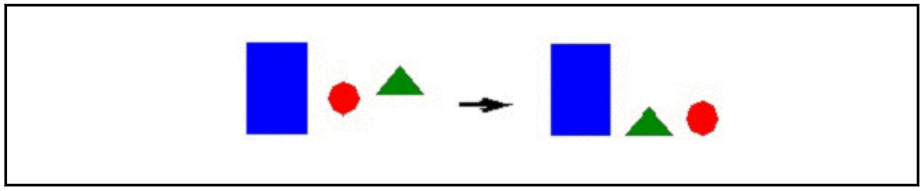


Abb.8-95: Objekte unten ausrichten

Vertikal zentrieren

Durch Klicken auf das Werkzeug , **Vertikal zentrieren**, werden alle markierten Objekte an der vertikalen Mittellinie des zuletzt markierten Objekts ausgerichtet.

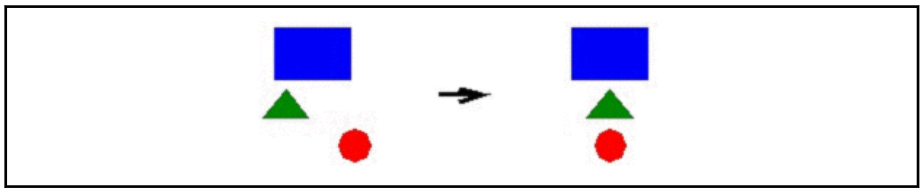


Abb.8-96: Objekte vertikal zentrieren

Horizontal zentrieren

Durch Klicken auf das Werkzeug , **Horizontal zentrieren**, werden alle markierten Objekte an der horizontalen Mittellinie des zuletzt markierten Objekts ausgerichtet.

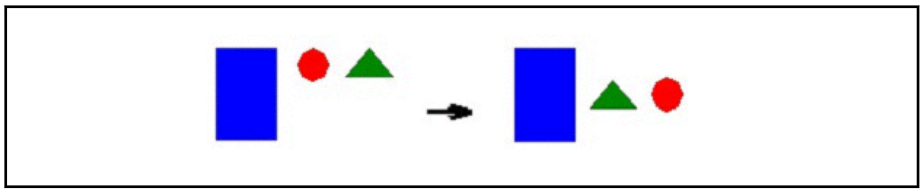


Abb.8-97: Objekte horizontal zentrieren

Horizontal verteilen

Durch Klicken auf das Werkzeug ,**Horizontal verteilen**, wird der gleiche horizontale Abstand zwischen einer Reihe von Objekten (drei oder mehr) eingefügt.

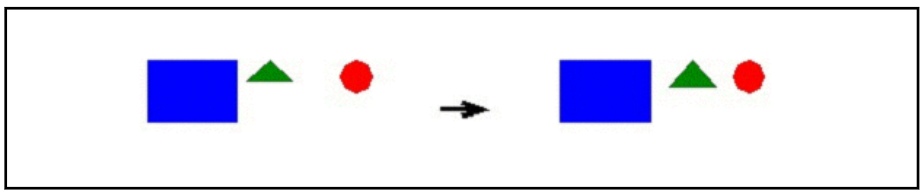


Abb.8-98: Objekte horizontal verteilen

Vertikal verteilen

Durch Klicken auf das Werkzeug , **Vertikal verteilen**, wird der gleiche vertikale Abstand zwischen einer Reihe von Objekten (drei oder mehr) eingefügt.

Bilder und Grafiken

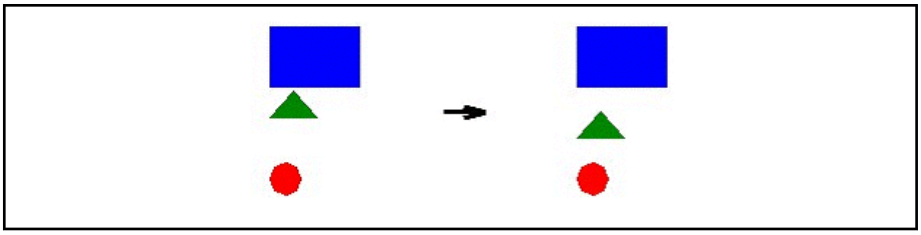


Abb.8-99: Objekte vertikal verteilen

Unter Umständen verschieben die Verteilungswerkzeuge das zuletzt markierte Objekt (mit ausgefüllten Ziehpunkten) um einige wenige Pixel, um alle Objekte gleichmäßig zu verteilen.

8.4.6 "Drehen" und "Spiegeln"

"Drehen"

Verwenden Sie die folgenden Werkzeuge aus der Symbolleiste **WinStudio-Ausrichtungsfunktionen** um Objekten zu drehen und zu spiegeln .

Drehen

Durch Klicken auf das Werkzeug , **Drehen** wird das markierte Objekt um 90 Grad im Uhrzeigersinn gedreht (eine Vierteldrehung):

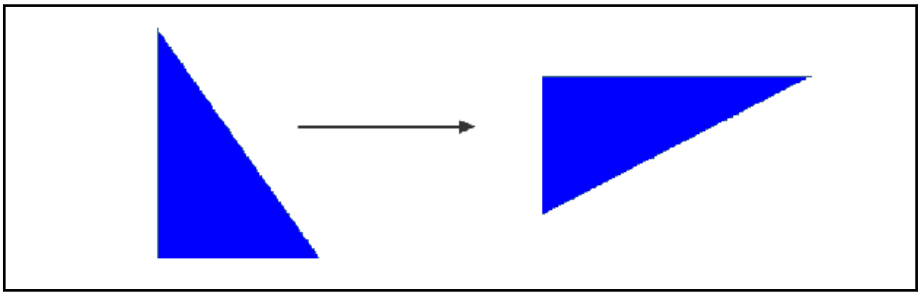


Abb.8-100: Objekte drehen

Dieses Werkzeug kann nur mit einem einzelnen markierten Objekt oder einer Gruppe auf Seite 259 von Objekten verwendet werden. Dieses Werkzeug kann nicht auf mehrere ausgewählte Objekte angewendet werden.

"Vertikal spiegeln"

Durch Anklicken des Werkzeugs **Vertikal spiegeln** wird das markierte Objekt vertikal umgekehrt. Das Objekt wird um eine gedachte vertikale Mittellinie gedreht, bis es ein Spiegelbild des Ursprungsobjekts ist:

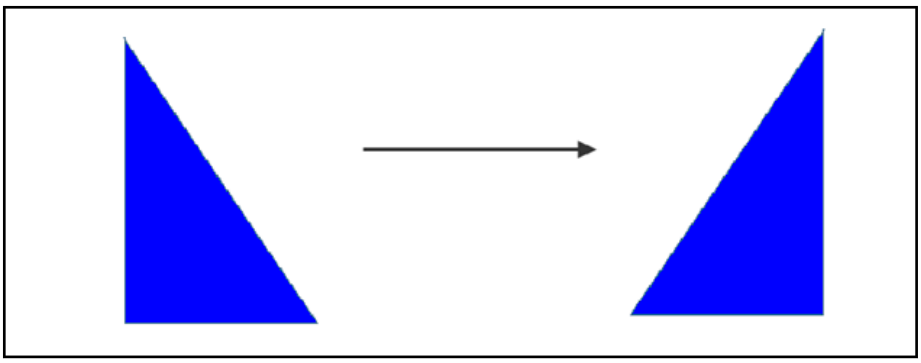


Abb.8-101: Objekte vertikal spiegeln

Dieses Werkzeug kann nur mit einem einzelnen markierten Objekt oder einer Gruppe auf Seite 259 von Objekten verwendet werden. Dieses Werkzeug kann nicht auf mehrere ausgewählte Objekte angewendet werden.

"Horizontal spiegeln"

Durch Anklicken des Werkzeugs , **Horizontal spiegeln** wird das markierte Objekt horizontal umgekehrt. Das Objekt wird um eine gedachte horizontale Mittellinie gedreht, bis es ein Spiegelbild des Ursprungsobjekts ist.

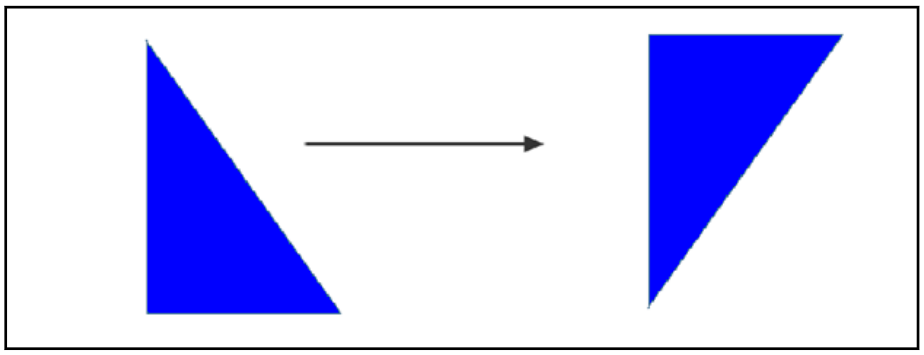


Abb.8-102: Objekte horizontal spiegeln

Dieses Werkzeug kann nur mit einem einzelnen markierten Objekt oder einer Gruppe auf Seite 259 von Objekten verwendet werden. Dieses Werkzeug kann nicht auf mehrere ausgewählte Objekte angewendet werden.

8.4.7 Position anpassen

Das Verschieben eines Bildobjekts durch Markieren und Ziehen kann ungenau sein – vor allem, wenn „Am Raster ausrichten“ aktiviert ist. Daher können die Positionierungswerkzeuge zum genauen Anpassen der Position (Nachjustieren) eines Objekts verwendet werden.

1. Klicken Sie auf ein Bildobjekt, um es zu markieren.
2. Verwenden Sie bei gedrückter <UMSCH>-Taste die Pfeiltasten zum Anpassen (Nachjustieren) der horizontalen und vertikalen Position des Objekts. Die Schrittweite ergibt sich aus der Rasterweite des Gitternetzes.

Siehe auch:

- Statuszeile auf Seite 96
- Rastereinstellungen auf Seite 181

8.4.8 "Gleiche Breite" / "Gleiche Höhe" für ausgewählte Objekte

Verwenden Sie die folgenden Funktionen in der Symbolleiste "WinStudio-Ausrichtungsfunktionen" um die Größe von Bildobjekten zu ändern:

Gleiche Breite für ausgewählte Objekte

Durch Klicken auf das Werkzeug , **Gleiche Breite für ausgewählte Objekte**, wird das zuerst markierte Objekt auf die Breite des zweiten gedehnt / gestaucht.

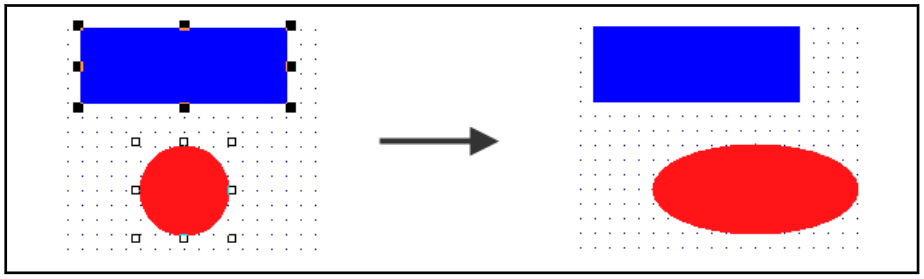


Abb.8-103: Objekte auf gleiche Breite dehnen

Bilder und Grafiken

Gleiche Höhe für ausgewählte Objekte

Durch Klicken auf das Werkzeug  , **Gleiche Höhe für ausgewählte Objekte**, wird das zuerst markierte Objekt auf die Höhe des zweiten gedehnt / gestaucht.

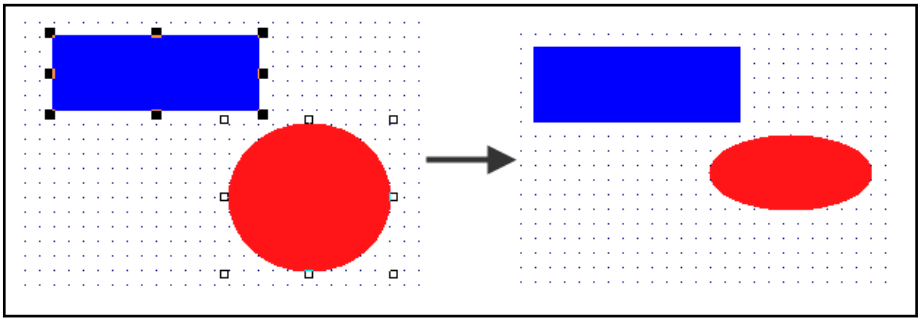


Abb.8-104: Objekte auf gleiche Breite dehnen



Mit **Breite anpassen** und **Höhe anpassen** kann eine Ellipse in einen Kreis oder ein Rechteck in ein Quadrat geändert werden. Es darf aber nur ein Objekt markiert sein.

Zur Größenanpassung von Objekten können auch der Mauszeiger oder die Pfeiltasten verwendet werden. Wenn ein Objekt (oder eine Objektgruppe) mit dem Mauszeiger markiert wird, erscheinen an allen Ecken sowie an den Mittelpunkten jeder Kante Ziehpunkte. Diese Ziehpunkte können wie folgt verwendet werden:

- **Zum Vergrößern eines Objekts** ziehen Sie einen Ziehpunkt in die Richtung, in der Sie die Objektgröße verändern wollen. Durch Ziehen an einem Kantenziehpunkt wird die Größe des Objekts nur in eine Richtung geändert (nur Höhe oder nur Breite). Durch Ziehen an einem Eckziehpunkt wird die Größe des gesamten Objekts geändert (Höhe und Breite). Beim Ziehen an einem Eckziehpunkt bleibt das Seitenverhältnis standardmäßig unverändert. Zur freien Größenänderung des Objekts halten Sie beim Ziehen am Ziehpunkt die Umschalttaste gedrückt.
- **Um die Größe eines Offenen auf Seite 183 oder Geschlossenen Polygons auf Seite 184** zu ändern, markieren und gruppieren auf Seite 259 Sie das Polygon (Kontextmenü Gruppieren).. Dann kann die Größe des Polygons wie bei einem normalen Objekt geändert werden.



Wenn man die Funktion zur Größenanpassung auf gruppierte Objekte (z.B. Verlinktes Symbol, Gruppe von Bildelementen) anwendet, werden alle in der Gruppe enthaltenen Einzelelemente in die gleiche Richtung und um das selbe Maß geändert.

Siehe auch:

- Statuszeile auf Seite 96

8.4.9 Füllfarbe

Durch Anklicken des Werkzeugs  , **Füllfarbe**, Symbolleisten "WinStudio Objekteigenschaften", wird die Standard-Füllfarbe für die folgenden Objekte festgelegt:

- Objekt Geschlossenes Polygon auf Seite 184
- Objekt Ellipse auf Seite 188
- Objekt Abgerundetes Rechteck auf Seite 187

- Objekt Rechteck auf Seite 185



Um Entwicklungszeit zu sparen, können mehrere Objekte (alle in der obenstehenden Liste genannten) markiert und **Füllfarbe** angewendet werden, um gleichzeitig für alle Objekte eine Standard-Füllfarbe festzulegen.

Siehe auch:

- **Farbe einer Komponente bearbeiten**: Farbschnittstelle auf Seite 265

8.4.10 Linienfarbe

Durch Anklicken des Werkzeugs  , **Linienfarbe**, Symbolleisten "WinStudio Objekteigenschaften", kann eine Linienfarbe für markierte Objekte oder eine Standardfarbe für neue Objekte festgelegt werden, einschließlich der folgenden:

- Objekt Linie auf Seite 182
- Objekt Offenes Polygon auf Seite 183
- Objekt Geschlossenes Polygon auf Seite 184
- Objekt Rechteck auf Seite 185
- Objekt Abgerundetes Rechteck auf Seite 187
- Objekt Ellipse auf Seite 188

Durch Anklicken des Werkzeugs **Linienfarbe** wird der Dialog "Linienauswahl" angezeigt. Verwenden Sie diesen Dialog zum Festlegen von Linienarten und -farben für die markierten Objekte.



Um Entwicklungszeit zu sparen, können mehrere der voran stehenden Objekte markiert und **Linienfarbe** angewendet werden, um gleichzeitig für alle Objekte eine Linienfarbe festzulegen.

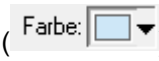
Siehe auch:

- **Farbe einer Komponente bearbeiten**: Farbschnittstelle auf Seite 265

8.4.11 Farbschnittstelle

Mit der Farbschnittstelle können Sie die Farbe einer Komponente bearbeiten.

Farbauswahl

1. Klicken Sie auf das Symbol (

) im Menüpunkt "Eigenschaften" eines Grafikobjektes.
2. Wenn sich die Popup-Box öffnet, können Sie eine der zehn Standardfarben oder sechzig auf einander abgestimmten Designfarben auswählen.

Bilder und Grafiken

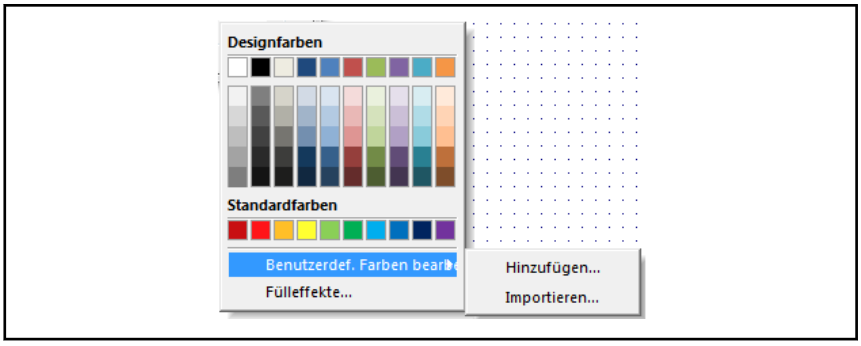


Abb.8-105: Popup-Box Hauptfarben

Die gewählte Farbe wird auf die bearbeitete Komponente angewendet.

Die Box erlaubt den Übergang zu

- Benutzerdefinierte Farben bearbeiten: "Hinzufügen, Standard- und Benutzerdefinierte Farben" auf Seite 266
- Benutzerdefinierte Farben bearbeiten: "Importieren" auf Seite 267
- "Fülleffekte" auf Seite 267

Hinzufügen, Standard- und Benutzerdefinierte Farben

1. Klicken Sie auf **Benutzerdefinierte Farben bearbeiten**, wenn Sie eine andere Farbe anwenden wollen. Die Auswahl kann im Bereich der Registerkarte **Standard** durch Anklicken des gewünschten Farbsechsecks oder mittels Pipette (Schaltfläche Auswahl / Select) erfolgen.



Die Pipette erlaubt die Übernahme einer beliebigen Farbe der IndraWorks-Umgebung.

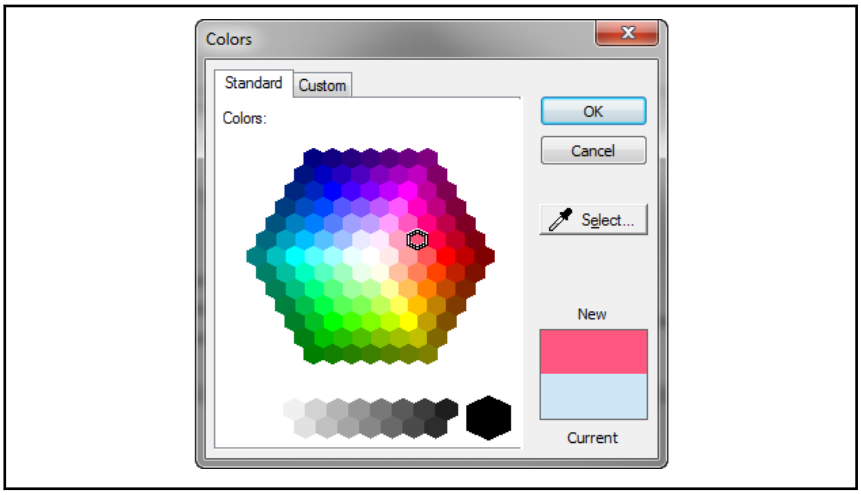


Abb.8-106: Popup-Box Standardfarben

2. Klicken Sie auf die Registerkarte **Benutzerdefiniert**.

   Die benutzerdefinierte Farbe kann durch Vorgabe des HSL-Farbcodes (Hue, Sat, Lum = Farbton, Sättigung, Helligkeit) oder des RGB-Farbcodes (rot, grün, blau) bestimmt werden (siehe auch "RGB-Codetabelle" auf Seite 268).

   Alternativ können Sie hier per Mausklick oder mittels Pipette praktisch jede beliebige Farbe wählen.

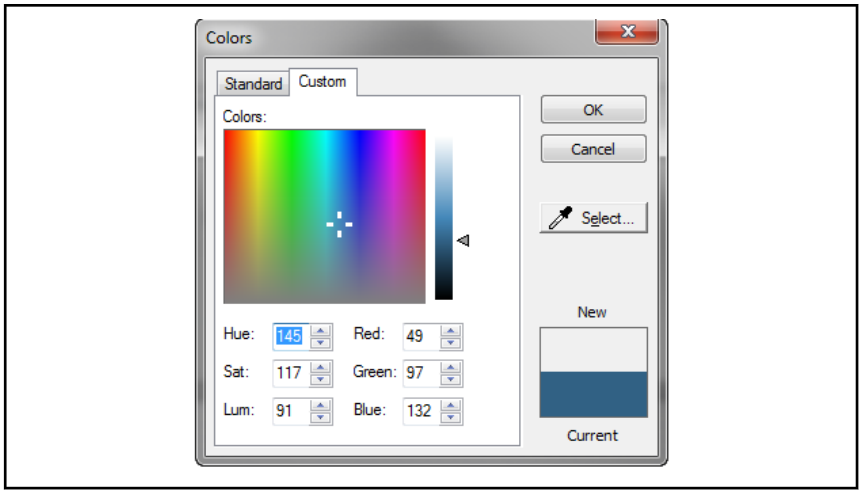


Abb.8-107: Popup-Box zum Erstellen einer benutzerdefinierten Farbe

3. Klicken Sie auf **OK**, um die ausgewählte Farbe auf die bearbeitete Komponente anzuwenden.

Importieren Diese Option ermöglicht den Import einer Textdatei "Benutzerdefinierte Farbe".

Fülleffekte

1. Je nach bearbeiteter Komponente ist die Option **Fülleffekte** auf der Popup-Schnittstelle verfügbar. Durch Anklicken dieser Option können graduelle Farben in verschiedenen Stilen und Varianten angewendet werden.



Abb.8-108: Dialog Fülleffekte

2. Wählen Sie in den Feldern **Start** und **Ende** zwei Farben aus, wählen Sie die "Schattierungsart" und klicken Sie auf die gewählte "Variante". Klicken Sie auf **OK**, um den Fülleffekt auf die bearbeitete Komponente anzuwenden.

Bei Projekten auf Windows CE-Zielsystemen ist die Schnittstelle **Fülleffekte** nur für das Objekt **Rechteck** verfügbar.

Auch wenn **Fülleffekte** ein nützliches Hilfsmittel zum Verbessern von Aussehen und Wirkung Ihrer WinStudio-Bilder ist, sollten Sie beachten, dass das Betriebssystem für das Füllen eines Objekts mit Fülleffekten länger braucht als für einfache Farben. Insbesondere bei der Projektierung von Win CE-Bedienoberflächen sollte dieser zusätzliche Ressourcenbedarf einkalkuliert werden.

Bilder und Grafiken

RGB-Codetabelle

Mit der Farbanimation auf Seite 251 können die Farben eines Objekts während der Laufzeit geändert werden. Beim Konfigurieren dieser Animation mit Typ = "Nach Farbe" kann mit dem Farbcode die Farbe festgelegt werden, die während der Laufzeit im Objekt angezeigt wird. Die folgende Tabelle zeigt die Codewerte und die RGB-Werte für die am häufigsten verwendeten Farben:

Name	RGB-Code			Code-Wert
	R (rot)	G (grün)	B (blau)	
Schwarz	0	0	0	0
Dunkelrot	128	0	0	128
Rot	255	0	0	255
Pink	255	0	255	16711935
Rosa	255	153	204	13408767
Braun	153	51	0	13209
Orange	255	102	0	26367
Hellorange	255	153	0	39423
Gold	255	204	0	52479
Hellbraun	255	204	153	10079487
Olivgrün	51	51	0	13107
Dunkelgelb	128	128	0	32896
Limette	153	204	0	52377
Gelb	255	255	0	65535
Hellgelb	255	255	153	10092543
Dunkelgrün	0	51	0	13056
Grün	0	128	0	32768
Seegrün	51	153	102	6723891
Leuchtendgrün	0	255	0	65280
Hellgrün	204	255	204	13434828
Dunkles Blaugrün	0	51	102	7877376
Blaugrün	0	128	128	8421376
Aqua	51	204	204	13421619
Türkis	0	255	255	16776960
Helltürkis	204	255	255	16777164
Dunkelblau	0	0	128	8388608
Blau	0	0	255	16711680
Hellblau	51	102	255	16737843
Himmelblau	0	204	255	16737843
Blassblau	153	204	255	16764057
Indigo	51	51	153	10040115

Name	RGB-Code			Code-Wert
	R (rot)	G (grün)	B (blau)	
Blaugrau	102	102	153	10053222
Violett	128	0	128	8388736
Pflaume	153	51	102	6697881
Lavendel	204	153	255	16751052
Grau-80%	51	51	51	3355443
Grau-50%	128	128	128	8421504
Grau-40%	150	150	150	9868950
Grau-25%	192	192	192	12632256
Weiß	255	255	255	16777215

Die Funktionen RGB-Farbe auf Seite 679 und RGB-Komponente auf Seite 679 können zum Verändern von Farbcodes während der Laufzeit verwendet werden.

Die Anzahl der bei der Entwicklung des Projekts verfügbaren Farben hängt von den im Betriebssystem der Entwicklungsstation konfigurierten Farbeinstellungen ab.

Die Anzahl der bei der Ausführung des Projekts verfügbaren Farben hängt von den im Betriebssystem der Laufzeitstation konfigurierten Farbeinstellungen ab.

8.4.12 Schriftarten

Durch Klicken auf das Werkzeug  , **Schriftarten**, Symbolleisten "WinStudio Objekteigenschaften", werden Schriftart und Farbe für die markierten Textobjekte auf Seite 190 oder eine Standard-Schriftart und -Farbe für neue Textobjekte festgelegt.



Um Entwicklungszeit zu sparen, können mehrere Textobjekte markiert und **Schriftart** angewendet werden, um gleichzeitig für alle Objekte Schriftart und Farbe festzulegen. (Diese Funktion kann allerdings nicht für gruppierte auf Seite 259 Textobjekte verwendet werden.)

"WinStudio" unterstützt jede auf dem Betriebssystem, verfügbare UNICODE-Schriftart. Daher ist es möglich, Bedienoberflächen für Sprachen zu erstellen, die keine westliche Standardschrift verwenden, z.B. Japanisch, Chinesisch, Arabisch, Kyrillisch usw.

Die Schriftart kann für mehrere Objekte gleichzeitig geändert werden, indem diese markiert werden (Umschalttaste beim Klicken gedrückt halten) und dann das Werkzeug **Schriftart** in der WinStudio-Symbolleiste angewendet wird.

Die Schriftart einzelner Bildobjekte kann auch über die Schaltfläche Schriftart im Fenster "Objekteigenschaften" angepasst werden:

Bilder und Grafiken

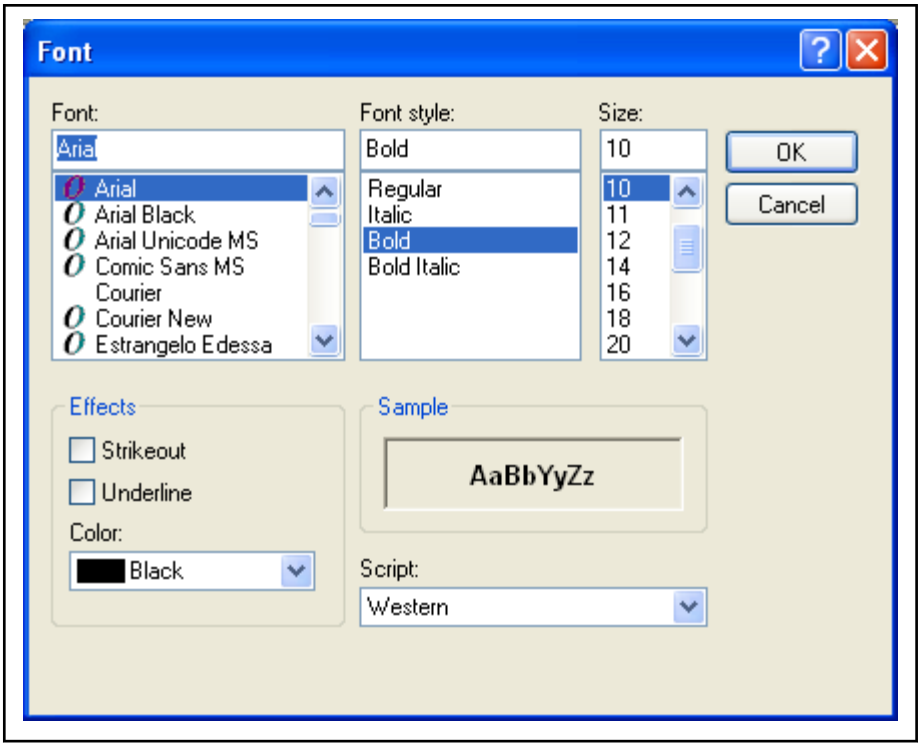


Abb.8-109: Standarddialog Schriftarten

Es können Schrifttyp, Stil, Größe, Effekte und Skript festgelegt werden.

Das links neben dem Namen der Schriftart angezeigte Symbol kennzeichnet das Schriftformat (Technologie)

Symbol	Technologie	Bemerkungen	
	TrueType	Outline Schrift-format	TrueType- und OpenType-Schriftarten sind sogenannte Outline Fonts, die aus Linien- und Kurvenbefehlen bestehen. OpenType ist eine Erweiterung von TrueType. Beide können skaliert und gedreht werden. Beide sind in allen Größen und auf allen von Windows unterstützten Ausgabegeräten darstellbar. Windows stellt eine Auswahl an OpenType-Schriftarten zur Verfügung, unter anderem Arial, Courier New, Lucida Console, Times New Roman, Symbol und Wingdings. Die Schriftart Type1 von Adobe Systems, Inc. ist eine Outline-Schriftart, die speziell für PostScript-Druckern entwickelt wurde. Die "Outlines" können skaliert und gedreht werden. Mit OpenType-Technologie bietet Windows volle Unterstützung für Type1-Schriftarten.
	OpenType		
entf.	Vektor		Vektor-Schriftarten werden unterstützt, weil eine Reihe von Programmen dieses Schriftformat noch verwenden. Vektor-Schriftarten werden aus einem mathematischen Modell erzeugt. Sie werden vorwiegend mit Plottern verwendet. Windows unterstützt drei Vektor-Schriftarten: Modern, Roman und Skript.
entf.	Raster		Raster-Schriftarten werden unterstützt, weil eine Reihe von Programmen dieses Schriftformat noch verwenden. Raster-Schriftarten werden als Bitmap-Bilder in Dateien gespeichert und bestehen aus einer Reihe von Punkten, sowohl auf dem Bildschirm als auch auf Papier.

⚠ VORSICHT **Es wird dringend empfohlen, nur TrueType- oder OpenType-Schriftarten zu verwenden.**

Mit anderen Technologien entwickelte Schriftarten (z.B. Courier) können nicht korrekt skaliert werden und könnten während der Laufzeit Probleme verursachen.

☞ Beim Entwerfen von Bildern werden die im Betriebssystem Ihrer Entwicklungsstation verfügbaren Schriftarten angeboten.

Auf der Laufzeitstation sehen die Schriftarten evtl. anders aus (z.B. andere Pixelgröße), auch wenn alle Einstellungen auf beiden Systemen gleich sind.

Diese Situation tritt häufiger auf, wenn Projekte auf Windows CE Betriebssystemen ausgeführt werden, wo die Schriftgrößen nicht immer identisch dargestellt werden. Daher ist es wichtig, die erstellten Bilder bereits während der Projektentwicklung auf der Zielplattform zu testen.

9 Alarme, Ereignisse und Trends

9.1 Alarme, Ereignisse und Trends, Allgemeines

Die Tasks Alarm und Trend werden zum Protokollieren historischer Daten, und die Objekte Alarm/Ereignis und Trend zur Anzeige historischer Daten in einem Projektbild verwendet.

Diese beiden Funktionen werden in der Regel zusammen verwendet, dies ist aber nicht zwingend notwendig.

Projektdaten können während der Laufzeit ohne Anzeige protokolliert werden und die während der Laufzeit angezeigten Daten können von außerhalb des Projekts stammen.

9.2 Alarmarbeitsblatt

9.2.1 Alarmarbeitsblatt, Allgemeines

Der Ordner Alarme ermöglicht die Konfiguration von Alarmgruppen und Tags für jede Gruppe. Das Arbeitsblatt Alarm definiert die von der Anwendung erzeugten Alarmmeldungen. Der Hauptzweck eines Alarms besteht darin, den Bediener über Probleme oder ungewöhnliche Zustände während des Prozesses zu informieren, damit er korrigierende Maßnahmen einleiten kann.

Das Arbeitsblatt Alarm wird durch das Modul Hintergrundtask ausgeführt (siehe <Ausführung Tasks>). Es verwaltet den Zustand aller Alarme und speichert die Alarmmeldungen in der Historie, falls dies konfiguriert wurde. Das Alarm-Arbeitsblatt selbst zeigt jedoch nicht die Alarmmeldungen in der Laufzeit an. Zum Anzeigen von Alarmen muss das Bildobjekt Alarm-/Ereignissteuerung auf Seite 290 aus dem Werkzeugfenster "WinStudio Screen Tools" in ein WinStudio-Bild eingefügt und konfiguriert werden.

Zum Erstellen eines neuen Alarmarbeitsblattsöffnen Sie das Kontextmenü des Ordners "Tasks" und wählen Sie **Hinzufügen ▸ Alarm**.

Zum Bearbeiten eines bestehenden Alarmarbeitsblatts führen sie im Projektexplorer einen Doppelklick auf das Element aus.

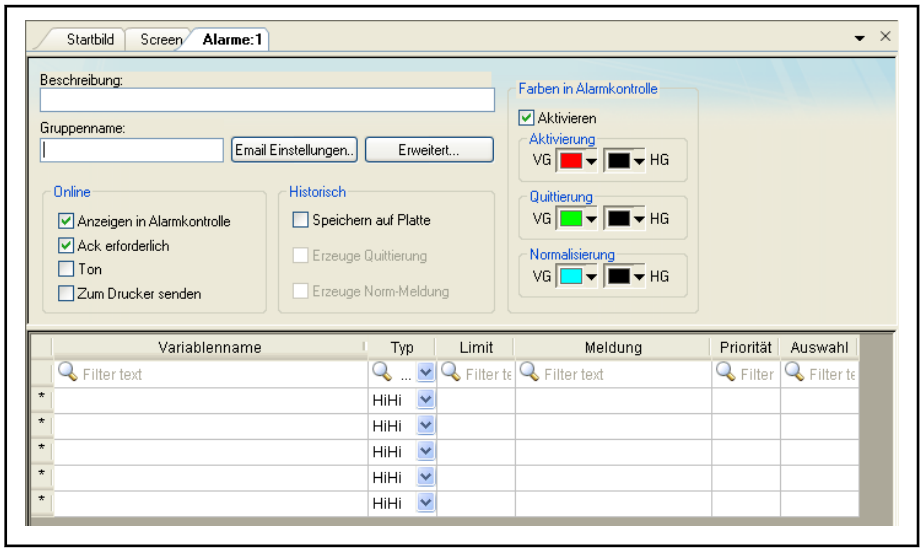


Abb.9-1: Alarmarbeitsblatt

Es können mehrfache Alarmgruppen (Arbeitsblätter) erstellt werden und jede Gruppe kann mit unabhängigen Einstellungen wie Meldungsfarbe, Aktivierung/Deaktivierung historisches Protokoll usw. konfiguriert werden.

Jedes Alarmarbeitsblatt besteht aus zwei Bereichen:

- **Titelbereich:** Auf alle in der gleichen Alarmgruppe konfigurierten Tags und Alarme angewendete Einstellungen. Diese Einstellungen erlauben die Konfiguration der Meldungsformatierung und der Aktionen, die aufgrund von Alarmereignissen ausgelöst werden müssen (z.B. Alarm drucken, Alarm per E-Mail verschicken usw.). Weitere Informationen finden Sie in Einstellungen Titelbereich auf Seite 275.
- **Hauptteil:** Tags auswählen und Bedingungen definieren, bei denen Alarmmeldungen erzeugt werden. Weitere Informationen finden Sie in Einstellungen Hauptbereich auf Seite 280.



- Die Alarmgruppe kann zum automatischen Versenden von E-Mail-Benachrichtigungen aufgrund von Alarmereignissen konfiguriert werden. Weitere Informationen finden Sie in E-Mail-Einstellungen auf Seite 276.
- Die mit jedem Tag verbundenen Alarmeigenschaften (im Hauptteil der Alarmgruppe konfiguriert) können auch mit dem Dialog Tag-Eigenschaften auf Seite 153 bearbeitet werden (**Tag-Eigenschaften** im Werkzeugfenster "Querverweise"). Allerdings ist es vor dem Verbinden eines Tags mit einer Alarmgruppe erforderlich, die Alarmgruppe zu erstellen und die Einstellungen in deren Kopfteil zu konfigurieren. Diese werden auf alle mit der Gruppe verbundenen Tags angewendet.
- Ab "WinStudio" v6.1+SP2 wurde der Alarm-Task verändert, um das automatische Bestätigen eines Alarms durch einen anderen Alarm zu verhindern. Zum Beispiel sollte der Hi- bzw. Lo-Alarm nicht automatisch bestätigt werden, wenn der HiHi- bzw. LoLo-Alarm und aktiviert wird. Zum Aktivieren des vorgenannten Verhaltens setzen Sie Ihrer Projektdatei (**.APP**) den folgenden Schlüssel:

  [Alarm]

  UseLegacyPriorityAck=1

HINWEIS

Diese im Hauptteil jedes Alarmarbeitsblatts konfigurierten Einstellungen werden in den Tag-Datenbank-Archiven gespeichert. Daher können Änderungen in der Tag-Datenbank den Inhalt der Alarm-Arbeitsblätter (Hauptteil) beeinflussen.

Beachten Sie, dass jeder Tag/Typ nicht in mehr als einer Alarmgruppe gleichzeitig verfügbar sein kann, da die Alarmgruppe eine mit jedem Tag-/Alarmtyp verbundene Eigenschaft ist

(z.B. Tag: Stufe; Alarm-Typ: Hi; Alarmgruppe: 2).

9.2.2 Titelbereich Alarmarbeitsblatt

Titelbereich Alarmarbeitsblatt, Allgemeines

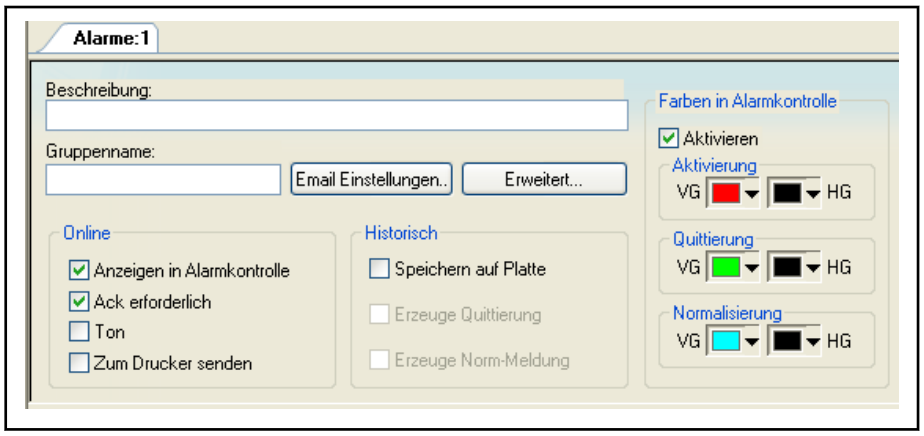


Abb.9-2: Einstellungen im Kopfteil eines Alarmarbeitsblatts

Die folgende Tabelle beschreibt die Einstellungen im Titelbereich eines Alarmarbeitsblatts (siehe Alarmarbeitsblatt auf Seite 273):

Feld	Bemerkungen	Syntax
Beschreibung	Beschreibung der Alarmgruppe. Wird im Arbeitsbereich angezeigt. Dieses Feld wird nur für Dokumentationszwecke verwendet.	Text (bis zu 80 Zeichen)
Gruppenname	Name der Alarmgruppe. Während der Laufzeit kann der Bediener Alarme mit dem eingebauten Dialog Filter des Alarm-/Ereignissteuerungsobjekts aufgrund des Gruppennamens filtern.	Text (bis zu 32 Zeichen)
E-Mail-Einstellungen	Öffnet den Dialog E-Mail-Einstellungen auf Seite 276, in dem die Einstellungen für automatisch aufgrund von Alarmbedingungen verschickte E-Mails konfiguriert werden können.	Schaltfläche
Erweitert	Öffnet den Dialog Weitere Einstellungen auf Seite 279, in dem die Einstellungen für automatisch aufgrund von Alarmbedingungen verschickte E-Mails konfiguriert werden können.	Schaltfläche
Online > Anzeige im Alarmobjekt	Wenn aktiviert, stehen die Alarme zur Anzeige auf dem Alarm-/Ereignissteuerungsobjekt auf Seite 290 zur Verfügung.	Kontrollkästchen
Online > Best. erforderlich	Wenn aktiviert, ist für Alarme eine Bestätigung erforderlich. In diesem Fall werden die Alarme auf dem Alarm-/Ereignissteuerungsobjekt auf Seite 290 (Online-Modus) angezeigt, bis sie bestätigt UND normalisiert sind.	Kontrollkästchen
Online > Piepton	Wenn aktiviert, gibt der Rechner einen Piepton aus, während zu bestätigende Alarme aktuell aktiv sind.	Kontrollkästchen

Alarme, Ereignisse und Trends

Feld	Bemerkungen	Syntax
Online > Senden an Drucker	Wenn aktiviert, werden die Alarmmeldungen sofort bei Auftreten des Alarmereignisses an den Drucker gesendet. Bei Verwendung dieser Option muss zum Drucken der Meldungen ein Matrixdrucker (anstelle eines Tintenstrahl- oder Laserdruckers) verwendet und um nur eine Zeile vorgeschobenen werden – andernfalls wird jeder Alarm auf ein neues Blatt Papier gedruckt. Die Alarme werden auf dem Standarddrucker gedruckt. Wenn Alarme an einen anderen Drucker als den Standarddrucker gesendet werden sollen, kann der Pfad/Name des Druckers durch Bearbeiten der folgenden Parameter in der Datei **projekt_name.APP** angegeben werden: **[AlarmProtokoll]** **Device=DruckerPfad/DruckerName**	Kontrollkästchen
Historie > Speichern auf Festplatte	Wenn aktiviert, werden die Alarmmeldungen bei Aktivierung im Protokoll Historie auf Seite 282 gespeichert.	Kontrollkästchen
Historie > Erzeugen v. Quittierungsmeldungen	Wenn aktiviert, werden die Alarmmeldungen bei Bestätigung im Protokoll Historie auf Seite 282 gespeichert.	Kontrollkästchen
Historie > Erzeugen norm. Meldungen	Wenn aktiviert, werden die Alarmmeldungen bei Normalisierung im Protokoll Historie auf Seite 282 gespeichert.	Kontrollkästchen
Farben im Alarmssteuerelementen > Aktivieren	Wenn aktiviert, werden die in dieser Gruppe konfigurierten Alarme in den für jeden Alarmzustand (**Aktivierung**, **Quittierung** oder **Normalisierung**) zugewiesenen Farben angezeigt, gemäß den in der Alarmgruppe konfigurierten Farben.	Farben auf Seite 265
Farben in Alarmssteuerelementen > VG und HG	Für die auf dem Objekt Alarme/Ereignissteuerung auf Seite 290 angezeigten Alarme können Vordergrundfarbe (VG) und Hintergrundfarbe (HG) konfiguriert werden. Jeder Alarmzustand kann mit einem anderen Farbschema angezeigt werden: • **Aktivierung**: Alarm aktiv und nicht bestätigt • **Quittierung**: Alarm aktiv und bestätigt • **Normalisierung**: Alarm nicht mehr aktiv und nicht bestätigt.	Farben auf Seite 265

E-Mail-Einstellungen für Alarmarbeitsblatt

"WinStudio" kann bei Auftreten von Alarmereignissen automatisch E-Mails versenden. Die E-Mails werden mit dem SMTP-Standard verschickt (Simple Message Transfer Protocol). Daher wird nur ein gültiges E-Mail-Konto mit SMTP-Server und POP3-Server benötigt – es muss keine zusätzliche Software wie z.B. Microsoft Outlook installiert werden.



Bevor E-Mails versendet werden können, muss die Funktion CnfEmail auf Seite 737 mindestens einmal erfolgreich ausgeführt werden (aus der internen WinStudio-Sprache). Diese Funktion legt die beim Versenden von E-Mails aus dem Projekt verwendeten Parameter für das E-Mail-Konto fest (z.B. SMTP-Server, Benutzername, Passwort usw.).

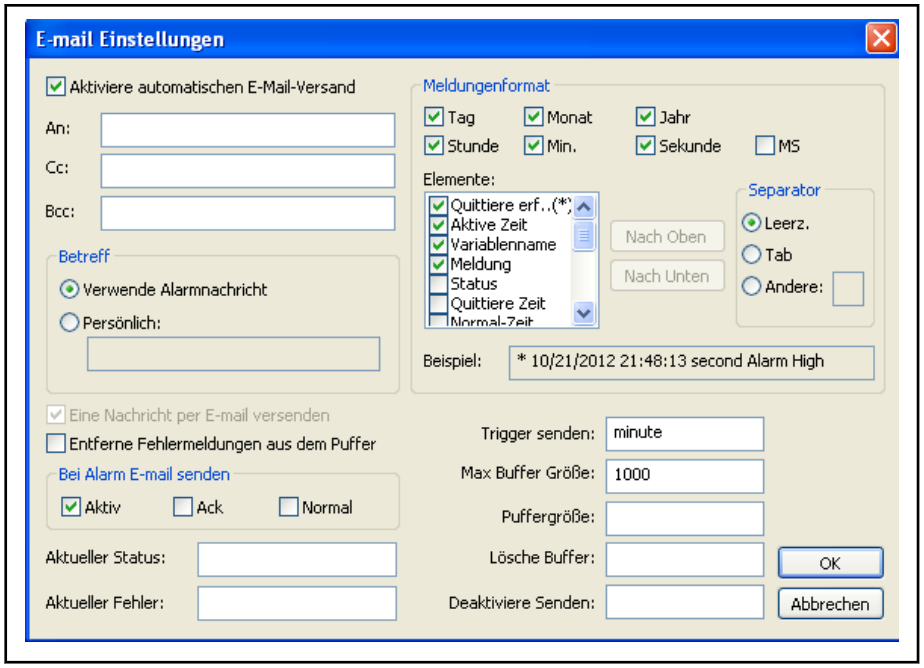


Abb.9-3: Alarmarbeitsblatt – E-Mail-Einstellungen

Feld	Bemerkungen	Syntax
Automatisches Versenden aktivieren	Name des mit dem Alarm verbundenen Tags.	Kontrollkästchen
An, Cc, Bcc	Empfängeradressen für die E-Mails. In jedem Feld (An, Cc und/oder Bcc) können mehrere E-Mail-Adressen eingegeben werden, zur Trennung der Adressen wird ein Semikolon (";") verwendet.	Text und/oder {Tag} (bis zu 1024 Zeichen)
Betreff	Wenn „Alarmmeldung verwenden“ ausgewählt ist, wird die Alarmmeldung selbst als Betreff für die zu versendende E-Mail verwendet. Bei Auswahl von „Benutzerdefiniert“ kann für den Betreff beim Versenden des Alarms ein benutzerdefinierter Text konfiguriert werden.	Optionsfeld / Text (bis zu 1024 Zeichen)
1 Meldung pro E-Mail senden	Wenn diese Option gewählt ist, wird jeder Alarm in einer eigenen E-Mail versendet und beim Auslösen des **Auslösers Senden** werden alle E-Mails versendet. Andernfalls werden alle Alarmmeldungen gepuffert und beim Auslösen des Auslösers Senden in einer einzigen E-Mail versendet. Wenn die Option Betreff mit „Alarmmeldung verwenden“ konfiguriert ist, kann diese Option nicht deaktiviert (abgewählt) werden.	Kontrollkästchen
Fehlgeschlagene Meldungen aus dem Puffer entfernen	Wenn diese Option angewählt ist, werden die E-Mails nach dem Sendeversuch aus dem Puffer entfernt, auch wenn ein Fehler aufgetreten ist und die E-Mail nicht verschickt wurde. Andernfalls verbleiben die Meldungen im Puffer, bis sie erfolgreich gesendet wurden oder bis der Puffer seine maximale Größe erreicht.	Kontrollkästchen

Alarme, Ereignisse und Trends

Feld	Bemerkungen	Syntax
E-Mail senden wenn Alarm	Ermöglicht das Konfigurieren, welche Alarmereignisse E-Mails erzeugen sollen: Bitte beachten Sie, dass jedes Ereignis individuell aktiviert/deaktiviert werden kann. • **Aktiv**: Wenn der Alarm aktiv wird. • **Best**: Wenn der Alarm bestätigt wird. • **Norm**: Wenn der Alarm normalisiert wird.	Kontrollkästchen
Aktueller Zustand	Der ggf. in diesem Feld konfigurierte Tag wird mit dem aktuellen Zustand aus der aktuellen oder letzten E-Mail, die das Projekt zu senden versucht hat, aktualisiert: • "-2": Falsche Version der Bibliothek INDMail.DLL. • "-1": Die Bibliothek INDMail.DLL ist korrupt. • " 0": Funktion SendEmailExt auf Seite 741 wird nicht ausgeführt. • " 1": E-Mail(s) versenden • " 2": Letzte E-Mail wurde erfolgreich versendet. • " 3": Fehler beim Senden der letzten E-Mail.	Tag
Aktueller Fehler	Der ggf. in diesem Feld konfigurierte Tag wird mit der Fehlermeldung aktualisiert, die das Ergebnis der letzten E-Mail beschreibt, die das Projekt zu senden versucht hat. Daher muss beim Konfigurieren eines Tags in diesem Feld dieser Tag vom Typ String sein.	Tag
Meldungsformat	Hier kann die tatsächliche Formatierung der als E-Mail gesendeten Meldung konfiguriert werden, auf Grundlage des/der Alarmereignisse/s: Bim Konfigurieren dieser Einstellungen zeigt das Feld **Beispiel** ein Beispiel für die Formatierung der Meldung gemäß der konfigurierten Einstellungen an. • **Tag**, **Monat**, **Jahr**, **Stunde**, **Minute**, **Sekunde**, **MS**: Die ausgewählten Optionen bilden den Zeitstempel für die Alarmmeldungen. MS steht für Millisekunden. • **Elemente**: Die ausgewählten Optionen bilden die E-Mail-Meldung für jeden Alarm. Die Reihenfolge der Elemente kann durch die Schaltflächen Nach oben und Nach unten konfiguriert werden. • **Trennzeichen**: Ermöglicht die Auswahl des Trennzeichens zwischen den hier ausgewählten Elementen.	Kontrollkästchen und Optionsfeld
Auslöser senden	Wenn die Alarmereignisse erzeugt werden, werden sie in einem internen Puffer (Speicher) vorgehalten. Wenn der in diesem Feld konfigurierte Tag seinen Wert ändert, werden die E-Mails im internen Puffer an die in den Feldern **An**, **Cc** und **Bcc** konfigurierten Adressen gesendet. Nach erfolgreichem Senden werden die E-Mails aus dem internen Puffer entfernt.	Tag

Feld	Bemerkungen	Syntax
Max Puffergröße	Höchstzahl der Alarmmeldungen (Ereignisse), die in diesem internen Puffer gleichzeitig gespeichert werden können. Wenn diese Obergrenze erreicht ist, folgt der Puffer dem FIFO-Prinzip (First in, First out) und löscht alte Meldungen, sobald neuere Meldungen erzeugt werden. So wird sichergestellt, dass der Puffer die in diesem Feld konfigurierte Höchstzahl nicht überschreitet.	Tag oder Zahl
Puffergröße	Der ggf. in diesem Feld konfigurierte Tag wird mit der aktuell im internen Puffer gespeicherten Anzahl der Meldungen (Ereignisse) aktualisiert.	Tag
Puffer löschen	Wenn der in diesem Feld konfigurierte Tag seinen Wert ändert, werden alle aktuell in diesem Puffer gespeicherten Meldungen (Ereignisse) gelöscht. Diese Meldungen werden niemals gesendet.	Tag
Senden deaktivieren	Wenn der in diesem Feld konfigurierte Wert WAHR ist, wird die E-Mail-Funktion vorübergehend deaktiviert. Bei deaktivierter E-Mail-Funktion erzeugte Alarme werden nicht im internen Puffer gespeichert. Außerdem werden in diesem Zustand KEINE E-Mails versendet, auch wenn der im Feld **Auslöser senden** konfigurierte Tag seinen Wert ändert.	Tag

Weitere Einstellungen für Alarmarbeitsblatt

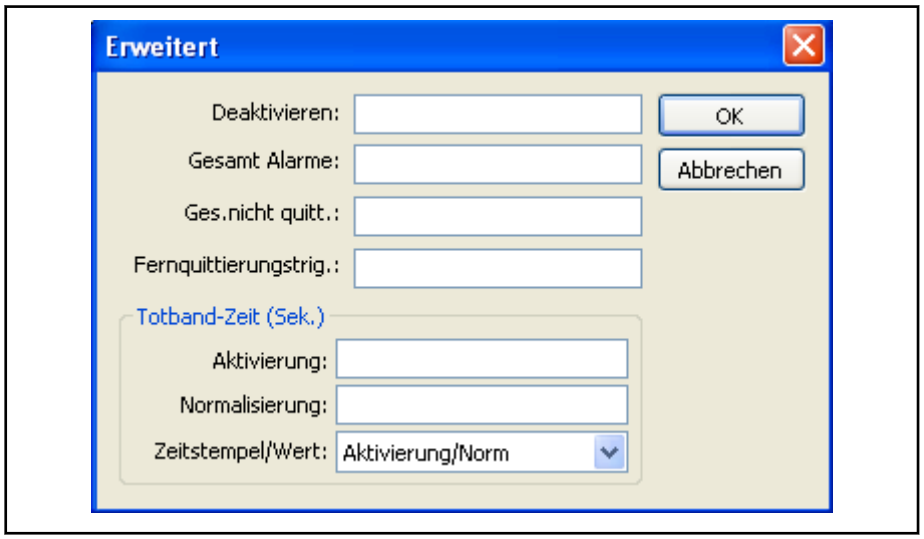


Abb.9-4: Alarmarbeitsblatt, Dialog Weitere Einstellungen

Die folgende Tabelle beschreibt die weiteren Einstellungen eines Alarmarbeitsblatts auf Seite 273:

Alarme, Ereignisse und Trends

Feld	Bemerkungen	Syntax
Deaktivieren	Wenn der hier konfigurierte Tag WAHR ist, werden alle in dieser Gruppe konfigurierten Alarme vorübergehend deaktiviert. Diese Option ist hilfreich zum Deaktivieren von Alarmen unter bestimmten Bedingungen (z.B. bei Wartungsarbeiten).	Tag
Alarme gesamt	Der ggf. in diesem Feld konfigurierte Tag wird mit der Anzahl der aktuell aktiven Alarme aus dieser Gruppe aktualisiert.	Tag
Gesamt unbest	Der ggf. in diesem Feld konfigurierte Tag wird mit der Anzahl der aktuell aktiven UND noch nicht bestätigten Alarme aus dieser Gruppe aktualisiert.	Tag
Remote Auslöser best	Wenn der in diesem Feld konfigurierte Tag seinen Wert ändert, werden alle aktiven Alarme aus dieser Gruppe bestätigt. Diese Option kann zum Bestätigen von Alarmen unabhängig von Aktionen des Bedieners verwendet werden.	Tag
Totband-Zeit > Aktivierung	Während der in diesem Feld festgelegten Zeit müssen alle Alarme kontinuierlich in ihrer Alarmbedingung bleiben, bevor sie aktiv werden. Diese Option ist hilfreich zum Erzeugen von Alarmen bei intermittierenden Bedingungen (z.B. Geräuschen). Wenn dieses Feld leer bleibt, wird der Alarm aktiviert, sobald seine Bedingung wahr wird.	Tag oder Zahl
Totband-Zeit > Normalisierung	Während der in diesem Feld festgelegten Zeit müssen alle Alarme kontinuierlich aus ihrer Alarmbedingung bleiben, bevor sie normalisiert werden. Diese Option ist hilfreich zum Normalisieren von Alarmen bei intermittierenden Bedingungen (z.B. Geräuschen). Wenn dieses Feld leer bleibt, wird der Alarm normalisiert, sobald seine Bedingung nicht mehr wahr ist.	Tag oder Zahl
Totband-Zeit > Zeitstempel/ Wert	Jeder Alarm erhält einen Zeitstempel der letzten wichtigen Aktivität, zusammen mit dem Tag-Wert zu diesem Zeitpunkt. Die Art der Aktivität zur Aktualisierung des Zeitstempels kann ausgewählt werden: • **Aktivierung/Norm** (Standard): Der Zeitpunkt, zu dem die Unempfindlichkeit endet, d.h. wenn der Alarm aktiviert oder normalisiert wird. • **Letzte Tag-Änderung**: Der Zeitpunkt der letzten Wertänderung des Tags während der Unempfindlichkeit. • **Startbedingung**: Zeitpunkt des Beginns der Unempfindlichkeit.	Combobox

9.2.3 Hauptteil Alarmarbeitsblatt

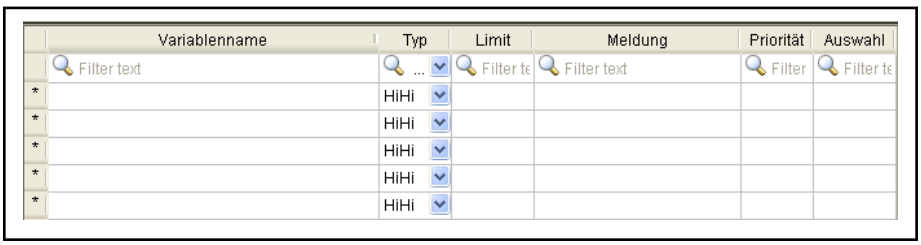


Abb.9-5: Hauptteil Alarmarbeitsblatt

Die folgende Tabelle beschreibt die Einstellungen im Hauptteil eines Alarmarbeitsblatts auf Seite 273:

Feld	Bemerkungen	Syntax
Tag-Name	Name des mit dem Alarm verbundenen Tags.	Tag
Typ	Alarm-Typ: Bei Verwendung der Typen Rate, AbwP und AbwM müssen im Dialog Tag-Eigenschaften zusätzliche Einstellungen konfiguriert werden (**Tag-Eigenschaften** im Werkezeugfenster "Querverweise"). • **HiHi**: Aktiviert den Alarm, wenn der Tag-Wert größer oder gleich dem Grenzwert ist. • **Hi**: Aktiviert den Alarm, wenn der Tag-Wert größer oder gleich dem Grenzwert ist. (Für Boolesche Tags wenn der Wert gleich 1 ist.) • **Lo**: Aktiviert den Alarm, wenn der Tag-Wert kleiner oder gleich dem Grenzwert ist. (Für Boolesche Tags wenn der Wert gleich 0 ist.) • **LoLo**: Aktiviert den Alarm, wenn der Tag-Wert kleiner oder gleich dem Grenzwert ist. • **Rate**: Aktiviert den Alarm, wenn der Tag-Wert sich schneller ändert als in der für den Alarm festgelegten Rate. (Für Boolesche Tags wenn sich der Wert ändert.) • **Dev+**: Aktiviert den Alarm, wenn der Tag-Wert größer oder gleich dem Sollwert plus der Höchstgrenze ist. • **Dev-**: Aktiviert den Alarm, wenn der Tag-Wert kleiner oder gleich dem Sollwert plus der Höchstgrenze ist.	Combobox
Limit	Mit jedem Alarm verbundener Grenzwert. Die Grenzwerte können während der Laufzeit dynamisch verändert werden, durch die Tag-Felder HiHiLimit, HiLimit, LoLimit, LoLoLimit, Rate, AbwP und AbwM (z.B. TagLevel->HiLimit).	Anzahl
Meldung	Mit dem Alarm verbundene Meldung. Die Meldung kann auf dem Alarm-/Ereignissteuerungsobjekt auf Seite 290 angezeigt und/oder in der Alarmhistorie auf Seite 282 gespeichert und/oder per E-Mail auf Seite 276 versendet werden, je nach den im Kopfbereich auf Seite 275 der Alarmgruppe konfigurierten Einstellungen.	Text und/oder {Tag} (bis zu 256 Zeichen)
Priorität	Mit dem Alarm verbundene Prioritätszahl. Beim Anzeigen von Alarmen auf dem Alarm-/Ereignissteuerungsobjekt auf Seite 290 kann der Bediener die Alarme nach Priorität filtern und/oder sortieren.	Zahl (von 0 bis 255)
Auswahl	Mit dem Alarm verbundenes Alias (z.B. BereichA, BereichB usw.). Beim Anzeigen von Alarmen auf dem Alarm-/Ereignissteuerungsobjekt auf Seite 290 kann der Bediener die Alarme nach deren Auswahlwert filtern und/oder sortieren.	Text (bis zu 7 Zeichen)

9.2.4 Speichern der Alarmhistorie in einer externen Datenbank

Standardmäßig werden die Alarm-Historie und das Ereignisprotokoll in Textdateien eines proprietären Formats im Ordner Alarme Ihres Projekts gespeichert. Sie können jedoch die Projekteinstellungen so abändern, dass diese statt dessen in einer externen SQL-Datenbank gespeichert werden.

1. Öffnen Sie die Registerkarte **Optionen** des Dialogs "Projekteinstellungen".

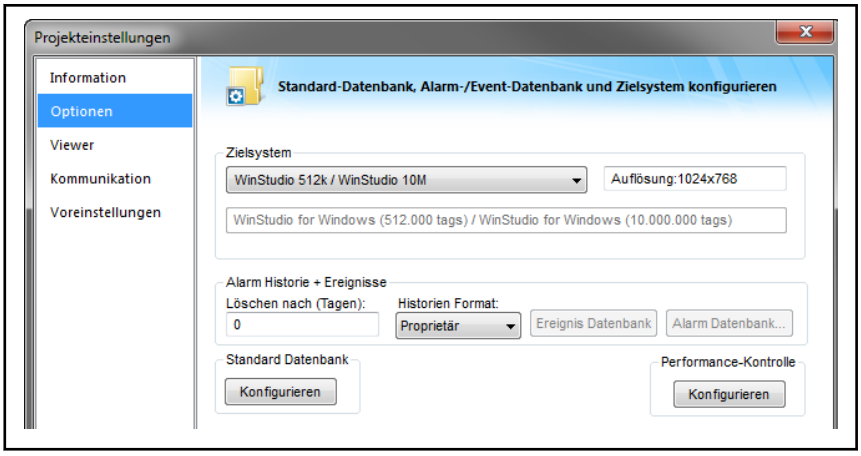


Abb.9-6: Projekteinstellungen: Optionen

2. Im Bereich **Alarm-Historie und Ereignisse** im Eingabefeld **Löschen nach (Tagen)** tragen Sie die Anzahl Tage der Historie ein, die Sie speichern wollen. In dem Maß, wie die Historie die festgelegte Anzahl an Tagen überschreitet, wird sie automatisch gelöscht, wobei die ältesten Einträge zuerst gelöscht werden. Wenn keine Zahl angegeben ist – wenn also das Feld leer gelassen oder auf 0 gesetzt wird – wird die Historie nie gelöscht. Wie viel Historie Sie speichern können, ist nicht beschränkt, aber je mehr Sie speichern, desto mehr Festplattenspeicher wird belegt.
3. Wählen Sie aus der Liste **Historien-Format** den Punkt **Datenbank** aus.
4. Um eine einzelne Standarddatenbank zu konfigurieren, die sowohl für die Alarm-Historie als auch das Ereignisprotokoll (und auch alle anderen Laufzeit-Tasks) verwendet werden soll, klicken Sie im Bereich **Standarddatenbank** auf den Eintrag **Konfigurieren**. Der Dialog "Konfiguration der Standarddatenbank" wird angezeigt. Konfigurieren Sie mithilfe dieses Dialogs die Datenbankverbindung. Nähere Informationen finden Sie unter Konfigurieren einer Standarddatenbank für alle Task-Historien auf Seite 508.
5. Um eine separate Datenbank für Ihr Ereignisprotokoll oder auch Ihre Alarm-Historie zu konfigurieren, klicken Sie jeweils auf **Ereignis-Datenbank** bzw. **Alarm-Datenbank**. In beiden Fällen wird der Dialog "Konfiguration der Datenbank" angezeigt. Konfigurieren Sie mithilfe dieses Dialogs die Datenbankverbindung. Weitere Informationen erhalten Sie in Konfiguration der Datenbank auf Seite 502.
6. Klicken Sie auf **OK**.

9.2.5 Format der Alarmhistorie

Speicherort und Format der Alarmhistorie hängen davon ab, ob das Historienformat auf Proprietär oder Datenbank eingestellt ist. Dieser Abschnitt beschreibt beide Möglichkeiten.

Wenn das Historienformat auf Proprietär eingestellt ist (Standardeinstellung), wird die Alarmhistorie im Alarmordner Ihres Projekts als eine Serie von Textdateien gespeichert, mit einer Datei pro Kalendertag. Der Name jeder Datei ist "AL"jjmmtt".ALH", wobei gilt:

- **jj** sind die letzten beiden Stellen des Jahres, in dem die Alarmhistoriendatei erzeugt wurde;
- **mm** ist der Monat, in dem die Alarmhistoriendatei erzeugt wurde; und
- **tt** ist der Tag, an dem die Alarmhistoriendatei erzeugt wurde.

Also befindet sich die Alarmhistoriendatei für den 7. Mai 2003 unter

[…]\Alarm\AL030507.ALH.



Zum Ändern des Speicherorts für diese Dateien verwenden Sie die Funktion SetAppAlarmPath auf Seite 706.

Innerhalb einer ***.ALH**-Datei für einen bestimmten Tag wird jeder Alarm in einer neuen Zeile gespeichert unter Verwendung des Verkettungszeichens ("|") zum Trennen der Felder, wie unten dargestellt:

**.ALH-Datei*

```
P1|P2|P3|P4|P5|P6|P7|P8|P9|P10|P11|P12|P13|P14|P15|P16|P17|P18|P19|P20|P21|P22|P23
P1|P2|P3|P4|P5|P6|P7|P8|P9|P10|P11|P12|P13|P14|P15|P16|P17|P18|P19|P20|P21|P22|P23
.
.
P1|P2|P3|P4|P5|P6|P7|P8|P9|P10|P11|P12|P13|P14|P15|P16|P17|P18|P19|P20|P21|P22|P23
```

Wenn das Historienformat auf Datenbank eingestellt ist, wird die Alarmhistorie in der als Alarmdatenbank konfigurierten Datenbank als Tabelle gespeichert. Weitere Informationen finden Sie unter Speichern der Alarmhistorie in einer externen Datenbank auf Seite 281.

Die Felder/Spalten der Alarmhistorie werden nachstehend beschrieben:

Proprietär	**Datenbank**		**Beschreibung**	**Dateivers.**
Feldnummer	**Spaltenname**	**Datentyp**		
P1	—	—	Dateiversion (aktuell = 003)	001
P2	Al_Start_Time	Zeitstempel	Startdatum (MM/TT/JJJJ)	001
P3			Startzeit (HH:MM:SS)	001
P4	Al_Tag	String	Tag-Name	001
P5	Al_Message	String	Alarmmeldung	001
P6	Al_Ack	Boolean	Bestätigung, wobei: • **0**: Alarm wurde bestätigt oder muss nicht bestätigt werden • **1**: Alarm wurde nicht bestätigt	001
P7	Al_Active	Boolean	Aktiv, wobei: • **0**: Alarm ist nicht aktiv • **1**: Alarm ist aktiv	001
P8	Al_Tag_Value	Real	Tag-Wert bei Auftreten des Alarms	001
P9	Al_Group	Integer	Alarmgruppennummer	001
P10	Al_Priority	Integer	Prioritätszahl	001
P11	Al_Selection	String	Auswahl	001

Alarme, Ereignisse und Trends

Proprietär	Datenbank		Beschreibung	Dateivers.
Feldnummer	Spaltenname	Datentyp		
P12	Al_Type	Integer	Typ, wobei: • **1** ist HiHi • **2** ist Hi(Ein) • **4** ist Lo(Aus) • **8** ist LoLo • **16** ist Rate(Veränderung) • **32** ist Abweichung+ • **64** ist Abweichung-	001
P13	Al_Ack_Req	Boolean	Bestätigung erforderlich, wobei: • **0**: Alarm erfordert Bestätigung • **1**: Alarm erfordert keine Bestätigung	001
P14	Al_Norm_Time	Zeitstempel	Normalisierungsdatum (MM/TT/JJJJ)	001
P15			Normalisierungszeit (HH:MM:SS)	001
P16	Al_Ack_Time	Zeitstempel	Bestätigungsdatum (MM/TT/JJJJ)	001
P17			Bestätigungszeit (HH:MM:SS)	001
P18	Al_User	String	Benutzername	002
P19	Al_User_Comment	String	Kommentar	002
P20	Al_User_Full	String	Vollständiger Benutzername	003
P21	Al_Station	String	Station	003
P22	Al_Prev_Tag_Value	Real	Vorheriger Wert	003
P23	Bias	Integer	Fehlerverzerrung Zeitzone	003
—	Al_Start_Time_ms	Integer	Anzahl der Millisekunden für den Zeitstempel Startzeit. Dieses Feld wird verwendet, wenn die Datenbank im Feld Zeitstempel keine ms unterstützt.	003
—	Al_Norm_Time_ms	Integer	Anzahl der Millisekunden für den Zeitstempel Normalisierungszeit. Dieses Feld wird verwendet, wenn die Datenbank im Feld Zeitstempel keine ms unterstützt.	003
—	Al_Ack_Time_ms	Integer	Anzahl der Millisekunden für den Zeitstempel Bestätigungszeit. Dieses Feld wird verwendet, wenn die Datenbank im Feld Zeitstempel keine ms unterstützt.	003
—	Al_Deleted	Boolean	Gelöscht, wobei: • **0**: Alarmmeldung wurde vom Benutzer nicht gelöscht (nicht sichtbar). • **1**: Alarmmeldung wurde vom Benutzer gelöscht (sichtbar).	003

Proprietär	Datenbank		Beschreibung	Dateivers.
Feldnummer	Spaltenname	Datentyp		
—	Last_Update	Zeitstempel	Zeitstempel der letzten Aktualisierung dieses Alarms.	003
—	Last_Update_ms	Integer	Anzahl der Millisekunden für den Zeitstempel "Letzte_Aktualisierung". Dieses Feld wird verwendet, wenn die Datenbank im Feld Zeitstempel keine ms unterstützt.	003



Die Spaltenbezeichnungen in der Datenbanktabelle können durch manuelles Bearbeiten der Projektdatei

(z.B. **[...]\projektname.APP**) wie folgt individuell angepasst werden:

[Alarm]

StandardName=NeuerName

Zum Beispiel:

[Alarm]

Message=Alarm_Meldung

Ack=Bestätigung

9.3 Ereignisse und Ereignisprotokolle

9.3.1 Ereignisse und Ereignisprotokolle, Allgemeines

Dieser Abschnitt beschreibt die Möglichkeiten von "WinStudio" für Protokollierung und Auslesen von Ereignissen. Ereignisse können Tag-Änderungen, das Erzeugen von Berichten oder Rezepturen, das Öffnen und Schließen von Bildern, das An- und Abmelden am bzw. vom Sicherheitssystem usw. sein. "WinStudio" speichert alle diese Ereignisse in einer Protokolldatei, die dann vom Alarm-/Ereignissteuerungsobjekt abgerufen werden kann.

9.3.2 Aktivierung der Ereignisprotokollierung

Standardmäßig ist die Ereignisprotokollierung deaktiviert, um die Laufzeitressourcen zu schonen. Mit der Ereignisprotokollierung im Projektexplorer wird das Speichern von Ereignissen in der Historiendatei aktiviert.

1. Öffnen Sie das Kontextmenü des Ordners "WinStudio" und wählen Sie den Eintrag **Werkzeuge ▸ Ereignis-Logger**. Der Dialog "Ereignis-Logger" wird angezeigt.

Alarme, Ereignisse und Trends

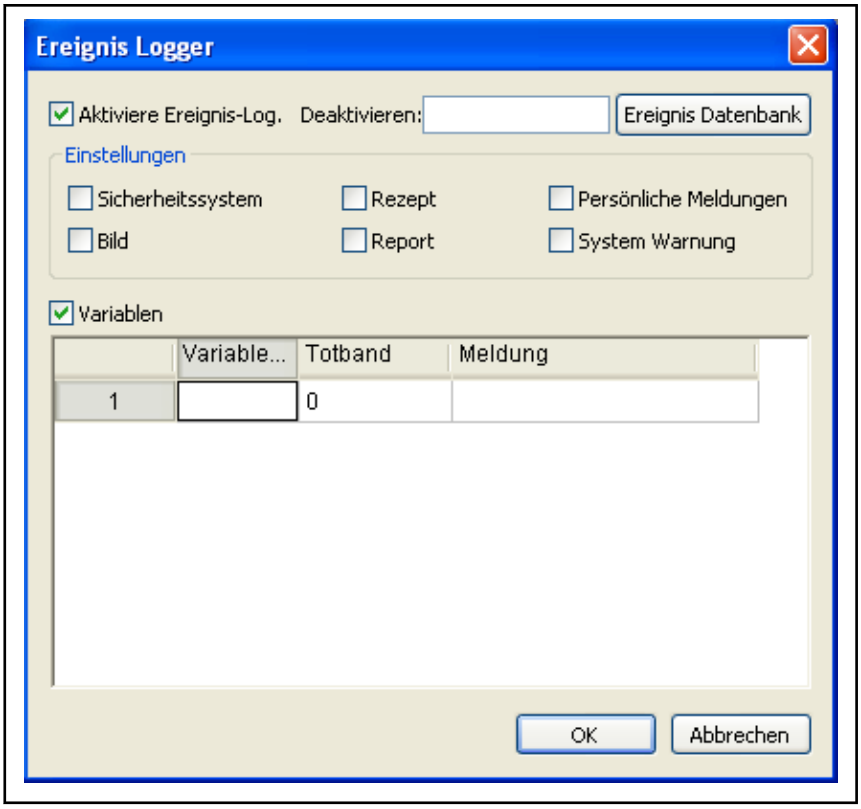


Abb.9-7: Dialog Ereigniseinstellungen

2. Wählen Sie **Ereignisprotokollierung aktivieren**.
3. Geben Sie im Feld **Deaktivieren** den Namen eines Anwendungs-Tags ein. Wenn während der Laufzeit der Wert des Tags WAHR ist (d.h. nicht null), wird die Ereignisprotokollierung ausgesetzt.
4. Wählen Sie im Bereich **Einstellungen**, welche Arten von Ereignissen Sie in der Historiendatei protokollieren möchten.

Option	Beschreibung
Sicherheitssystem	Vom Sicherheitssystem Ihres Projekts erzeugte Ereignisse, einschließlich: Benutzer anmelden/abmelden Benutzer erstellt/entfernt durch Aufrufen der Funktionen CreateUser auf Seite 611 oder RemoveUser auf Seite 619. Benutzer gesperrt/ entsperrt durch Aufrufen der Funktionen BlockUser auf Seite 610 oder UnblockUser auf Seite 620. Benutzer nach wiederholter Eingabe eines ungültigen Passworts vom Sicherheitssystem gesperrt Passwort abgelaufen Passwort geändert Ungültiger Anmeldeversuch
Anzeige	Ereignisse Bild öffnen und Bild schließen.
Rezeptur	Rezepturen geladen, gespeichert, initialisiert oder gelöscht.
Bericht	Berichte auf Festplatte gespeichert oder an Drucker gesendet.

Option	Beschreibung
Benutzerdef. Meldungen	Ereignisse erzeugt durch Aufruf der Funktion SendEvent auf Seite 766.
Systemwarnung	Verschiedene Laufzeitwarnungen und -Fehler, einschließlich: Fehler, die beim Versenden von Alarmen per E-Mail auftreten Tag wurde gesperrt/entsperrt Division durch Null Verbindung/Trennung zu/von Remote-Sicherheitssystem

5. Zum Protokollieren von Änderungen in bestimmten Anwendungsvariablen, aktivieren Sie den Bereich **Variablen** und legen dann in der Tabelle die Variablen fest.

   Die Option **Variablen** ist hilfreich zum Protokollieren von Ereignissen, die nicht wichtig genug sind, um Alarmbedingungen zu sein (zum Beispiel "Motor ein", "Motor aus" usw.).

Spalte	Beschreibung
Variablenname	Der Name der Anwendungsvariablen, die in der Historiendatei protokolliert werden soll.
Totband	Ein Wert zum Filtern von Änderungen, so dass nur über diesem Wert liegende Änderungen protokolliert werden (Unempfindlichkeitsbereich). Wenn zum Beispiel für einen Tag-Wert von 50 ein Totbandwert von 5 festgelegt wird und der Tag-Wert sich auf 52 ändert, dann registriert das System diese Veränderung nicht im Ereignisprotokoll, weil die Änderung kleiner als 5 ist. Wenn die Änderung des Tags aber größer oder gleich 5 ist, dann protokolliert das System den neuen Wert in der Historiendatei.
Meldung	Eine Zeichenkette (Meldung) mit Bezug auf diese Tag-Änderung. Tags können in Meldungen unter Verwendung der Syntax **{tagname}** festgelegt werden.

Die Option **Variablen** ist hilfreich für das Protokollieren von Ereignissen, die nicht wichtig genug sind, um Alarmbedingungen zu sein (zum Beispiel Motor ein, Motor aus usw.).

6. Klicken Sie auf **OK**.

Standardmäßig wird das Ereignisprotokoll im Alarmordner Ihres Projekts als Serie von Textdateien gespeichert. Weitere Informationen unter Format des Ereignisprotokolls auf Seite 288.

Alternativ kann das Ereignisprotokoll in einer externen SQL-Datenbank gespeichert werden. Weitere Informationen unter Speichern des Ereignisprotokolls in einer externen Datenbank auf Seite 281.

9.3.3 Speichern des Ereignisprotokolls in einer externen Datenbank

Standardmäßig werden die Alarm-Historie und das Ereignisprotokoll in Textdateien eines proprietären Formats im Ordner Alarme Ihres Projekts gespeichert. Sie können jedoch die Projekteinstellungen so abändern, dass diese statt dessen in einer externen SQL-Datenbank gespeichert werden.

1. Öffnen Sie die Registerkarte **Optionen** des Dialogs "Projekteinstellungen".

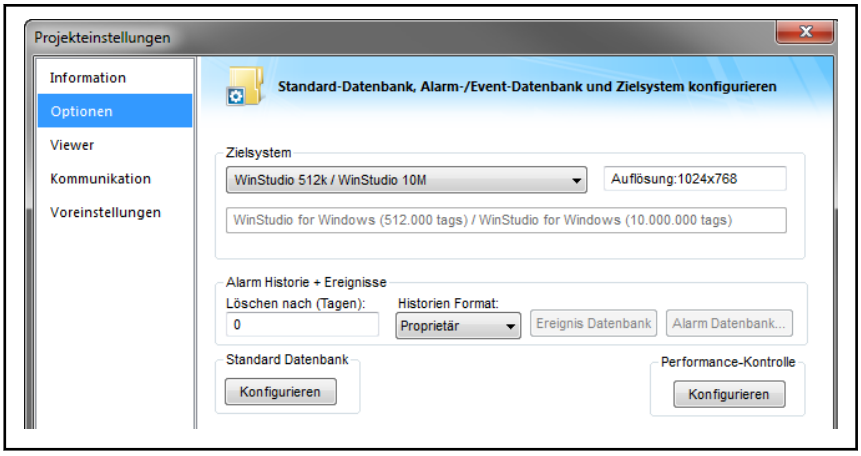


Abb.9-8: Projekteinstellungen: Optionen

2. Im Bereich **Alarm-Historie und Ereignisse** im Eingabefeld **Löschen nach (Tagen)** tragen Sie die Anzahl Tage der Historie ein, die Sie speichern wollen. In dem Maß, wie die Historie die festgelegte Anzahl an Tagen überschreitet, wird sie automatisch gelöscht, wobei die ältesten Einträge zuerst gelöscht werden. Wenn keine Zahl angegeben ist – wenn also das Feld leer gelassen oder auf 0 gesetzt wird – wird die Historie nie gelöscht. Wie viel Historie Sie speichern können, ist nicht beschränkt, aber je mehr Sie speichern, desto mehr Festplattenspeicher wird belegt.
3. Wählen Sie aus der Liste **Historien-Format** den Punkt **Datenbank** aus.
4. Um eine einzelne Standarddatenbank zu konfigurieren, die sowohl für die Alarm-Historie als auch das Ereignisprotokoll (und auch alle anderen Laufzeit-Tasks) verwendet werden soll, klicken Sie im Bereich **Standarddatenbank** auf den Eintrag **Konfigurieren**. Der Dialog "Konfiguration der Standarddatenbank" wird angezeigt. Konfigurieren Sie mithilfe dieses Dialogs die Datenbankverbindung. Nähere Informationen finden Sie unter Konfigurieren einer Standarddatenbank für alle Task-Historien auf Seite 508.
5. Um eine separate Datenbank für Ihr Ereignisprotokoll oder auch Ihre Alarm-Historie zu konfigurieren, klicken Sie jeweils auf **Ereignis-Datenbank** bzw. **Alarm-Datenbank**. In beiden Fällen wird der Dialog "Konfiguration der Datenbank" angezeigt. Konfigurieren Sie mithilfe dieses Dialogs die Datenbankverbindung. Weitere Informationen erhalten Sie in Konfiguration der Datenbank auf Seite 502.
6. Klicken Sie auf **OK**.

9.3.4 Format des Ereignisprotokolls

Speicherort und Formatierung des Ereignisprotokolls hängen davon ab, ob das Historienformat auf Proprietär oder Datenbank eingestellt ist. Dieser Abschnitt beschreibt beide Möglichkeiten.

Wenn das Historienformat auf Proprietär eingestellt ist (Standardeinstellung), wird das Ereignisprotokoll im Alarmordner Ihres Projekts als eine Serie von Textdateien gespeichert, mit einer Datei pro Kalendertag. Der Name jeder Datei ist "EV"jjmmtt".ALH", wobei gilt:

- **jj** sind die letzten beiden Stellen des Jahres, in dem die Ereignisprotokolldatei erzeugt wurde;
- **mm** ist der Monat, in dem die Ereignisprotokolldatei erzeugt wurde; und
- **tt** ist der Tag, an dem die Ereignisprotokolldatei erzeugt wurde.

Also befindet sich die Ereignisprotokolldatei für den 7. Mai 2003 unter **[...]\Alarm\EV030507.EVT**.

Zum Ändern des Speicherorts für diese Dateien verwenden Sie die Funktion SetAppAlarmPath auf Seite 706.

Innerhalb einer ***.EVT**-Datei für einen bestimmten Tag wird jedes Ereignis in einer neuen Zeile gespeichert unter Verwendung des Verkettungszeichens ("|") zum Trennen der Felder, wie unten dargestellt:

**.EVT*

```
P1|P2|P3|P4|P5|P6|P7|P8|P9|P10|P11|P12|P13
P1|P2|P3|P4|P5|P6|P7|P8|P9|P10|P11|P12|P13
.
.
P1|P2|P3|P4|P5|P6|P7|P8|P9|P10|P11|P12|P13
```

Wenn das Historienformat auf Datenbank eingestellt ist, wird das Ereignisprotokoll in der als Ereignisdatenbank konfigurierten Datenbank als Tabelle gespeichert. Weitere Informationen unter Speichern des Ereignisprotokolls in einer externen Datenbank auf Seite 281.

Die Felder/Spalten des Ereignisprotokoll werden nachstehend beschrieben:

Proprietär	**Datenbank**		**Beschreibung**	
Feldnummer	**Spaltenname**	**Datentyp**		
P1	Version	Integer	Dieses Feld wird nur erstellt, wenn das Historienformat auf Proprietär eingestellt ist. Aktuelle Version: 002	
P2	Event_Type	Integer	1	Sicherheitssystem
			2	Anzeige
			3	Rezeptur
			4	Bericht
			5	Benutzerdefinierte Meldung
			6	Systemwarnung
			7	Protokoll-Tags
P3	Event_Time	Zeitstempel	Zeitstempel zur Angabe, wann das Ereignis auftrat. Wenn das Historienformat auf Proprietär eingestellt ist, speichert das Projekt die Ereigniszeit im folgenden Format: **MM/TT/JJJJ HH:MM:SS.MSS**. Wenn das Historienformat auf Datenbank eingestellt ist, speichert das Projekt die Ereigniszeit im Standard-Zeitstempelformat der Datenbank.	
P4	Event_Info	String	Tag-Name.	
P5	Wert	Real	Tag-Wert bei Auftreten des Ereignisses.	
P6	Source	String	Name des Tasks, der das Ereignis erzeugt hat.	
P7	Benutzer	String	Benutzer angemeldet bei Auftreten des Ereignisses.	
P8	User_Full	String	Vollständiger Name des Angemeldeten Benutzers bei Auftreten des Ereignisses.	
P9	Meldung	String	Ereignismeldung.	
P10	Station	String	Name der Station (Rechner), auf der das Ereignis aufgetreten ist.	
P11	Kommentar	String	Vom Benutzer bei Auftreten des Ereignisses eingegebener Kommentar (optional). Dieses Feld existiert nur für Version >= 2.	

Alarme, Ereignisse und Trends

Proprietär	Datenbank		Beschreibung
Feldnummer	Spaltenname	Datentyp	
P12	Previous_Value	Real	Vor dem Ereignis aufgetretener Tag-Wert. Dieses Feld existiert nur für Version >= 2.
—	Gelöscht	Boolean	Dieses Feld wird nur erstellt, wenn das Historienformat auf Datenbank eingestellt ist. • 0 (FALSCH): Ereignismeldung wurde nicht gelöscht. • 1 (WAHR): Ereignismeldung wurde gelöscht.
P13	Bias	Integer	Unterschied (in Minuten) zwischen Ereignis_Zeit und WEZ_Zeit. Dieses Feld existiert nur für Version >= 2.
—	Last_Update	Zeitstempel	Zeitstempel zur Angabe, wann das Register erstellt/geändert wurde. Dieses Feld dient zur Synchronisation der Datenbanken bei Verwendung der sekundären Datenbank zusätzlich zur primären Datenbank. Dieses Feld wird nur erstellt, wenn das Historienformat auf Datenbank eingestellt ist.

Die Spaltenbezeichnungen in der Datenbanktabelle können durch manuelles Bearbeiten der Projektdatei

(z.B. **[...]\Eigene Dateien\WinStudio v8.0 Projekte\projektname\projektname.APP**)

wie folgt individuell angepasst werden:

[EventLogger]

StandardName=NeuerName

Zum Beispiel:

[EventLogger]

Event_Info=Information

Message=Ereignis_Meldung

9.4 Alarm-/Ereignissteuerungsobjekt

Verwenden Sie das Werkzeug **Alarm/Ereignissteuerung** zum Hinzufügen eines Alarm- oder Ereignissteuerungsobjekts zu einem Projektbild.

Erstellen und Konfigurieren eines Alarm-/Ereignissteuerungsobjekts:

1. Klicken Sie im Werkzeugfenster "WinStudio Screen Tools" auf das Element **Alarmobjekt**.
2. Klicken Sie in ein geöffnetes WinStudio-Bild und ziehen Sie mit der Maus, um den Umriss des Objekts zu erstellen und anzupassen.
3. Mit einem Doppelklick auf das Objekt öffnet sich der folgende Dialog "Objekteigenschaften".

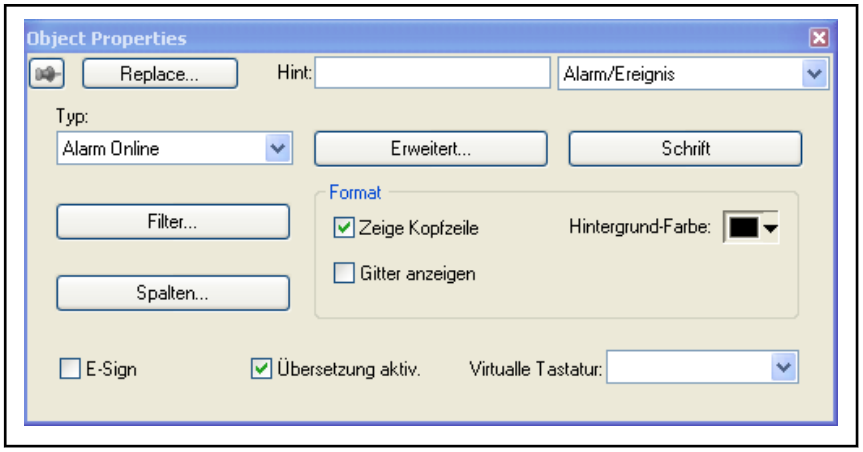


Abb.9-9: Objekteigenschaften: Alarm-/Ereignissteuerung

Der Dialog wird zur Festlegung der folgenden Parameter verwendet:

- Wählen Sie im Bereich "Typ" einen Alarmobjektmodus aus:
  - **Alarm Online**: Zeigt nur aktuelle Alarmmeldungen an.
  - **Alarmhistorie**: Zeigt nur Alarmmeldungen aus der Alarmhistoriendatenbank an.
  - **Alarmhistorie + Ereignis**: Zeigt sowohl Alarmmeldungen aus der Alarmhistoriendatenbank und protokollierte Ereignisse aus der Ereignishistoriendatenbank an.
  - **Ereignis**: Zeigt nur protokollierte Ereignisse aus der Ereignishistoriendatenbank an.
- Durch Klicken (Aktivieren) des Kontrollkästchens **Gitter anzeigen** werden im Objekt Gitterlinien angezeigt.

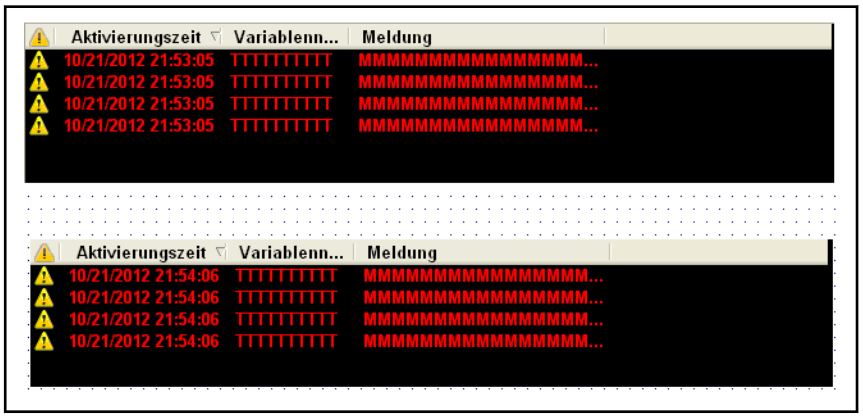


Abb.9-10: Anzeige von Gitterlinien

- Durch Klicken (Aktivieren) des Kontrollkästchens **Kopfzeile anzeigen** wird am Objekt ein Titelbereich angezeigt.

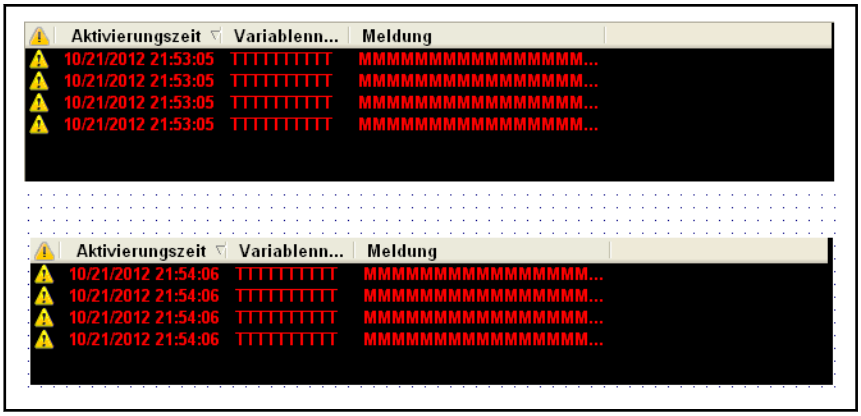


Abb.9-11: Anzeige eines Titelbereichs

- Verwenden Sie das Farbfeld **Hintergrundfarbe** zum Auswählen einer Hintergrundfarbe für das Objekt. Klicken Sie zum Öffnen der Farbpalette auf das Farbfeld und wählen Sie dann eine Farbe durch Anklicken aus.
- Durch Klicken (Aktivieren) des Kontrollkästchens **Übersetzung aktivieren** wird die externe Übersetzung von Meldungen mithilfe des Übersetzungswerkzeugs aktiviert. (Siehe Übersetzungswerkzeug auf Seite 437 für weitere Informationen.)
- Kontrollkästchen **E-Sign**: Wenn diese Option angewählt ist, wird der Benutzer vor dem Ausführen der Animation zur Eingabe der elektronischen Signatur aufgefordert.
- **Virtuelle Tastatur**: Für dieses Objekt wird der Typ Virtuelle Tastatur verwendet. Vor dem Konfigurieren der Virtuellen Tastatur für diese Schnittstelle muss die Option" Virtuelle Tastatur" in den "Viewer"-Einstellungen (Register **Viewer** im Eigenschaftendialog von WinStudio) ausgewählt werden.

Schriftarten Klicken Sie auf die Schaltfläche **Schriftarten** zum Öffnen einer Standardschnittstelle Schriftarten auf Seite 269, in der die Anzeigeeigenschaften für den Meldungstext festgelegt werden können.

Spalten Klicken Sie auf die Schaltfläche **Spalten** zum Öffnen des Dialogs "Spalten", in dem die Anzeigeeigenschaften für die Spalten im Objekt festgelegt werden können.



Abb.9-12: Dialog Spalten

- Die Liste "Verfügbar" enthält alle für dieses Objekt verfügbaren Spaltentypen. Die Liste "Sichtbar" enthält alle für dieses Objekt aktuell verwendeten Spaltentypen. Durch Klicken auf die Schaltflächen **»** und **«** können Elemente zwischen den beiden Listen bewegt werden.

Das Alarmobjekt kann so konfiguriert werden, dass kürzlich ersetzte Werte zusammen mit den entsprechenden neuen Werten angezeigt werden. Verschieben Sie dazu sowohl den Eintrag **Wert** als auch den Eintrag **Vorherige** in die Liste "Sichtbar". Verwenden Sie die Schaltflächen **Nach oben** oder **Nach unten** zum Anordnen der Spalten in der Liste "Sichtbar".

- Verwenden Sie die Felder **Marke** und **Breite** im Bereich "Eigenschaften" zum Ändern der Standard-Spaltenbeschriftungen und -breiten während der Laufzeit.
- Verwenden Sie die Combobox **Ausrichten** zum Festlegen der Ausrichtung (**Links**, **Mitte** oder **Rechts**) für den Alarmmeldungstext innerhalb einer bestimmten Spalte.
- Durch Klicken (Aktivieren) des Kontrollkästchens **Verfügbar während Laufzeit** kann der Benutzer während der Laufzeit auf Seite 290 der sichtbaren Liste ausgewählte Spalten hinzufügen.
- Verwenden Sie das Feld **Taste**, um jeder Spalte ein Tastaturkürzel hinzuzufügen. So können die Informationen im Alarmsteuerungsobjekt anstelle mit dem Mauszeiger mit der Tastatur nach jeder Spalte sortiert werden.

Zum Beenden klicken Sie auf **OK** zum Schließen des Dialogs "Spalte".

Beim Bestätigen eines Alarms sendet das Alarmsteuerungsobjekt eine Meldung mit den folgenden Informationen an den Alarm-Task auf Seite 273: **Tag-Name**, **Typ**, **Benutzer** und **Station**. Mit dieser Lösung können bestätigte Alarme von einem Thin Client gesteuert werden.

Filter

Zum Filtern von Alarmmeldungen während der Laufzeit klicken Sie auf die Schaltfläche **Filter**. Im angezeigten Dialog "Filter" können Filterparameter für das Alarmsteuerungsobjekt festgelegt werden.

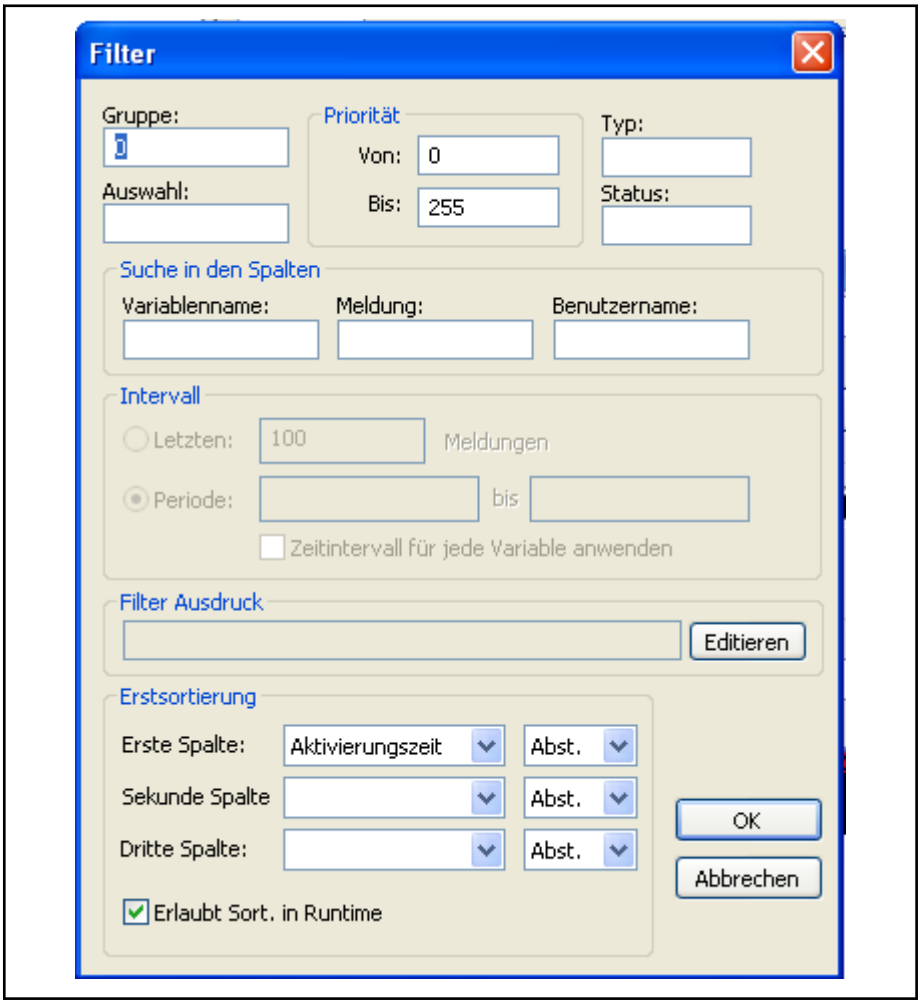


Abb.9-13: Dialog Filter

- Verwenden Sie das Feld **Gruppe** zum Filtern von Meldungen nach der Nummer des Alarmarbeitsblatts auf Seite 273. Die Arbeitsblätter sind im Ordner Alarme unterhalb des Ordners WinStudio\Tasks organisiert, beginnend mit 1. Wenn für die Gruppe 0 angegeben wird, werden alle Arbeitsblätter angezeigt. Zum Festlegen eines Arrays von Gruppen können Kommas oder Bindestriche verwendet werden, z.B. **1,3,5-6**.
- Verwenden Sie das Feld **Auswahl** zum Filtern von Meldungen nach dem im Alarmarbeitsblatt konfigurierten Auswahltext.
- Verwenden Sie die Felder **Von** und **Bis** im Bereich "Priorität" zum Filtern von Meldungen nach der im Alarmarbeitsblatt konfigurierten Priorität. Geben Sie in diesen Feldern numerische Werte zum Begrenzen des Prioritätsbereichs ein.

- Verwenden Sie das Feld **Typ** zum Filtern von Meldungen nach Alarmtyp (z.B. HiHi, Hi, Lo, LoLo, Rate, Abw+, Abw-). Zum Festlegen mehrerer Typen werden Kommas verwendet, z.B. "HiHi, LoLo".
- Verwenden Sie das Feld **Status** zum Filtern von Meldungen nach Alarmzustand:

Wert	Beschreibung
0	Alle Alarme (Standard)
1	Alle aktiven und unbestätigten Alarme
2	Alle aktiven und bestätigten Alarme
3	Alle inaktiven und bestätigten Alarme
4	Alle inaktiven und unbestätigten Alarme

Wird dieses Feld leer gelassen, kommt dies der Eingabe eines Nullwerts gleich.

- Verwenden Sie die Textfelder **Variablenname**, **Meldung** und/oder **Benutzername** im Bereich "Suchen in Spalten" zum Festlegen von Filterkriterien für Meldungen. Geben Sie in das Textfeld einen Tag-Namen, eine Meldung und/oder einen Benutzernamen ein, nach dem "WinStudio" suchen soll.
- Verwenden Sie die Parameter im Bereich "Intervall" zum Filtern von Meldungen nach den letzten **x** Meldungen (**Neueste**) oder auf Grundlage einer Zeitspanne (**Zeitraum**). Wenn kein Intervall angegeben wird, werden nur die Alarme für den aktuellen Tag angezeigt.
- Im Bereich "Filterausdruck" kann ein Ausdruck konfiguriert werden, der unerwünschte Meldungen aus der Anzeige herausfiltert. Es werden nur Meldungen angezeigt, die dem Ausdruck entsprechen. Zum Eingeben eines Ausdrucks klicken Sie auf die Schaltfläche **Bearbeiten**. Der Dialog "Filterausdruck Alarm" wird angezeigt.

- In den Feldern **Gruppe**, **Auswahl**, **Variablenname**, **Meldung** und **Benutzername** können zum Ändern dieser Werte während der Laufzeit String-Tags in geschweiften Klammern festgelegt werden (z.B. {tagname}).
- Im Feld **Typ** und den Feldern **Zeitraum** im Bereich Intervall müssen String-Tags ohne geschweifte Klammern angegeben werden (z.B. {tagname}). In diese Felder können keine Werte direkt eingegeben werden.
- In den Feldern **Von** und **Bis** im Bereich "Priorität", den Feldern **Status** und dem Feld **Neueste** im Bereich "Intervall" können Integer-Tags festgelegt werden.
- Im Filterausdruck können Platzhalterzeichen (* und ? für das Festlegen von Werten für die Felder **Auswahl**, **Variablenname**, **Meldung** und **Benutzername** verwendet werden.

- Im Bereich Filterausdruck kann ein Ausdruck konfiguriert werden, der unerwünschte Meldungen aus der Anzeige herausfiltert. Es werden nur Meldungen angezeigt, die dem Ausdruck entsprechen.

  Zum Eingeben eines Ausdrucks klicken Sie auf die Schaltfläche **Bearbeiten**. Der Dialog "Filterausdruck Alarm" wird angezeigt.

Der Filter Ausdruck muss der folgenden grundlegenden Syntax entsprechen:

[Spalten Name]Vergleichsoperator'Wert'

wobei **Spalten Name** der Name der Spalte im Alarm-/Ereignissteuerungsobjekt ist.

Zum Beispiel:

[Aktivierungszeit]>'08/17/2007 15:00'

Dieser Filter zeigt nur Alarmmeldungen mit Aktivierungszeiten größer (später) als 15:00 am 17.08.2007 an.



- Die Höchstzahl der Zeichen ist 1024 für den Engineering-Modus und 2048 für den Laufzeit-Modus.
- Die Spalten **Anzeigewert** und **Status.** werden von dem Filterausdruck nicht unterstützt.

- Durch Verwendung der logischen Operatoren UND, ODER und NICHT können mehrere Bedingungen gleichzeitig verwendet werden.

  Zum Beispiel: **[Typ]='HiHi' ODER [Typ]='LoLo' UND [Aktivierungszeit]>'08/17/2007 15:00'**
- Im Filterausdruck können Platzhalterzeichen (* und ?) verwendet werden.
- Wenn der **Spaltenname** aus nur einem Wort besteht (z.B. **Wert**), ist die Verwendung der eckigen Klammern nicht notwendig.
- Während der Laufzeit kann der Filterausdruck durch Festlegung von String-Tags in geschweiften Klammern verändert werden.

  Zum Beispiel: **[Wert]='{AlarmFilterWert}'**
- Zum Verwenden von mehr als 1024 Zeichen im Filterausdruck während der Laufzeit muss mehr als ein Tag zwischen geschweiften Klammern mit der Syntax **{TagName1} UND {TagName2}** verwendet werden.

- Zum Festlegen der Standard-Sortierreihenfolge verwenden Sie die Parameter im Bereich "Anfangssortierung". Wählen Sie aus der Combobox **Spalte** einen Sortiertyp aus und verwenden Sie dann **Aufst** oder **Abst** zum Sortieren in aufsteigender oder absteigender Reihenfolge. Es können bis zu drei Sortierebenen konfiguriert werden.

☞ Wenn alle drei Ebenen nicht den Sortiertyp Aktivierungszeit verwenden, sortiert das Projekt automatisch auf einer vierten Ebene nach der Aktivierungszeit in absteigender Reihenfolge.

Diese Art der Sortierung auf vierter Ebene kann nicht geändert werden, es kann aber durch manuelles Bearbeiten der Projektdatei (**projekt_name.app**) die Standardreihenfolge zwischen aufsteigend und absteigend umgeschaltet werden, durch Ändern der folgenden Einstellung:

[Objects]

DescendingAlarmListTime=TRUE | FALSE

TRUE sortiert in absteigender Reihenfolge, **FALSE** sortiert in aufsteigender Reihenfolge. Bitte beachten Sie, dass diese Einstellung nur für Projekte funktioniert, die mit "Rexroth WinStudio v7.2" oder höher erstellt oder aktualisiert wurden.

Aktivieren Sie das Kontrollkästchen **Sortieren während der Laufzeit erlauben**, wenn es dem Benutzer möglich sein soll, die Sortierreihenfolge während der Laufzeit zu ändern.

Weitere Einstellungen

Klicken Sie auf die Schaltfläche **Erweitert** zum Öffnen des Dialogs "Weitere Einstellungen", in dem weiterführende Einstellungen für das Alarmobjekt festgelegt werden können.

Alarme, Ereignisse und Trends

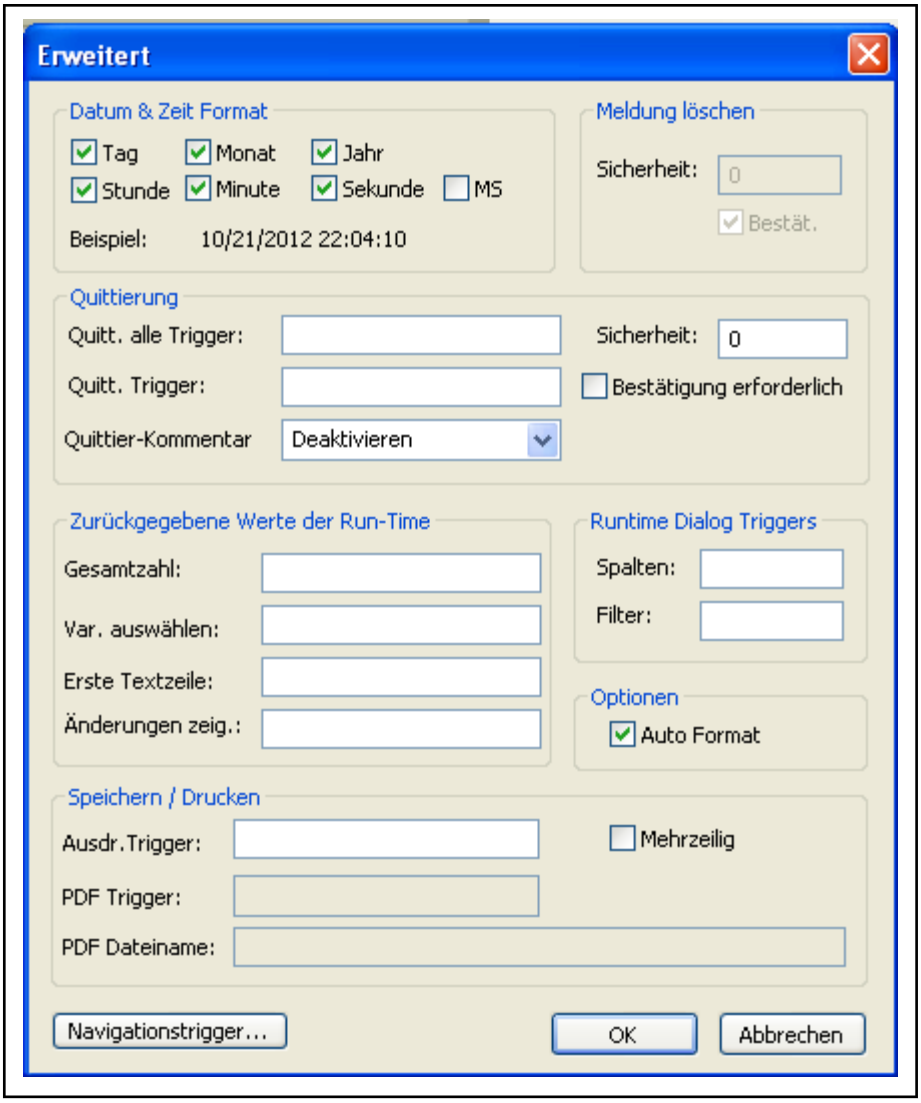


Abb.9-14: Dialog Weitere Einstellungen

- Verwenden Sie die Parameter im Bereich "Datums- & Zeitformat" zur Steuerung, welche Datums- und Zeitinformationen in der Alarmmeldung angezeigt werden. Durch Klicken (Aktivieren) eines Kontrollkästchen wird dieses Element in die Anzeige aufgenommen. Hinweis: **MS** steht für Millisekunden.

Beachten Sie den **Beispieltext** als Vorschau für das Aussehen der Informationen in der Alarmmeldung.

- Verwenden Sie die Parameter im Bereich "Quittierung" zur Steuerung, wie Alarme bestätigt werden.
  - Feld **Sicherheit**: Geben Sie einen numerischen Wert ein, um festzulegen, welche Sicherheitsstufe zum Bestätigen einer Alarmmeldung erforderlich ist. Nur Benutzer mit der angegebenen Sicherheitsstufe können auf die Meldung reagieren.

  - Feld **Trigger alle best**: Geben Sie einen Tag ein, um einen Wert zu erhalten. Wenn der Tag seinen Wert ändert, zeigt er an, dass alle Meldungen im Alarmobjekt bestätigt wurden.
  - Feld **Trigger best.**: Geben Sie einen Tag ein, um einen Wert zu erhalten. Wenn der Tag seinen Wert ändert, zeigt er an, dass die oberste Meldung im Alarmobjekt bestätigt wurde.
  - Kontrollkästchen **Bestätigung erforderlich**: Durch Klicken (Aktivieren) dieses Kontrollkästchens wird ein Bestätigungsdialog angezeigt, wenn der Benutzer versucht, einen einzelnen Alarm zu bestätigen.
  - Kontrollkästchen **Kommentar freigegeben (nur für Einzelbest)**: Durch Klicken (Aktivieren) dieses Kästchen kann der Benutzer direkt nach der Bestätigung Kommentare zu dem Alarm eingeben.
- Verwenden Sie die Parameter im Bereich "Auslöser Laufzeitdialog" zur Steuerung von
  - Feld **Spalte**: Geben Sie einen Tag ein, um einen Wert zu erhalten. Wenn der Tag seinen Wert ändert, wird ein Dialog geöffnet, in dem der Benutzer die im Objekt sichtbaren Spalten benutzerdefiniert anpassen kann.
  - Feld **Filter**: Geben Sie einen Tag ein, um einen Wert zu erhalten. Wenn der Tag seinen Wert ändert, wird ein Dialog geöffnet, in dem der Benutzer die im Objekt sichtbaren Spalten filtern kann.
- Verwenden Sie die Parameter im Bereich "Meldung löschen" zur Steuerung, wer Alarmmeldungen aus der Alarmhistorie löschen darf:
  - **Sicherheit**: Verwenden Sie dieses Feld zur Festlegung, welche Sicherheitsstufe zum Löschen von Alarmmeldungen benötigt wird. Nur Benutzer mit der entsprechenden Sicherheitsstufe können Alarmmeldungen löschen.
  - **Bestätigen**: Durch Klicken (Aktivieren) dieses Kästchens muss der Benutzer das Löschen einer Meldung bestätigen, bevor "WinStudio" die ausgewählte Alarmmeldung tatsächlich löscht.
- **Auslöser Drucken**: Wenn der in diesem Feld konfigurierte Tag umgeschaltet wird, wird der aktuelle Status des Alarm-/Ereignissteuerungsobjekts an den Standarddrucker geschickt.
- Feld **Auslöser PDF**: Wenn der in diesem Feld konfigurierte Tag umgeschaltet wird, wird der aktuelle Status des Alarm-/Ereignissteuerungsobjekts als PDF-Datei an dem unter **PDF Dateiname** festgelegten Ort gespeichert.
- Feld **PDF Dateiname**: Geben Sie einen vollständigen Dateipfad für das Speichern der PDF-Datei ein. Es kann auch ein Tag-Name mit der Syntax **{tag}** eingegeben werden.

☞ **Auslöser PDF** und **PDF Dateiname** werden in Projekten, die auf Windows Embedded oder Thin Client laufen, nicht unterstützt.

- Kontrollkästchen **Mehrzeilig**: Wenn diese Option angewählt ist, wird der Ausdruck oder das PDF gemäß der verfügbaren Spaltenbreite formatiert und der Text in jeder Zelle wird umgebrochen, so dass der gesamte Text zu sehen ist.

- Feld **Anzahl gesamt**: Geben Sie einen Integer-Tag ein, um zu sehen, wie viele Alarme verbleiben, nachdem "WinStudio" die Alarmobjekte mit den im Dialog Filter auf Seite 290 festgelegten Parametern gefiltert hat.
- Kontrollkästchen **Auto-Format**: Wenn dieses Kästchen aktiviert ist, werden Dezimalwerte in den Spalten Anzeigewert, Vorherige und Werte gemäß der von der Funktion SetDecimalPoints auf Seite 681 erzeugten virtuellen Tabelle formatiert.
- Feld **Ausgewählter Tag**: Geben Sie einen String-Tag ein, um es dem Benutzer zu ermöglichen, durch Klicken auf eine Alarmmeldung den Namen des mit dem Alarmereignis verbundenen Tags anzuzeigen.
- Feld **Erste Textzeile**: Geben Sie einen String-Tag ein. Dieser Tag erhält den Text aus allen Feldern aus der ersten Zeile der Alarm-/Ereignissteuerung. Diese Felder sind <tab>-getrennt. Bei jeder Änderung der ersten Zeile – entweder durch neue Alarme/Ereignisse oder einfach durch neue Anordnung der Zeilen – wird der Wert des konfigurierten Tags aktualisiert.
- Feld **Gesamtanzahl**: Geben Sie einen Integer-Tag ein. Dieser Tag erhält die laufende Gesamtanzahl der Änderungen in der Alarm-/Ereignissteuerung. Wenn zum Beispiel ein neuer Alarm auftritt oder wenn ein Alarm bestätigt wird, wird der Wert des konfigurierten Tags erhöht. Umsortieren der Zeilen wird nicht als Änderung gezählt.
- Klicken Sie auf die Schaltfläche **Auslöser Navigation**, um den folgenden Dialog zu öffnen:

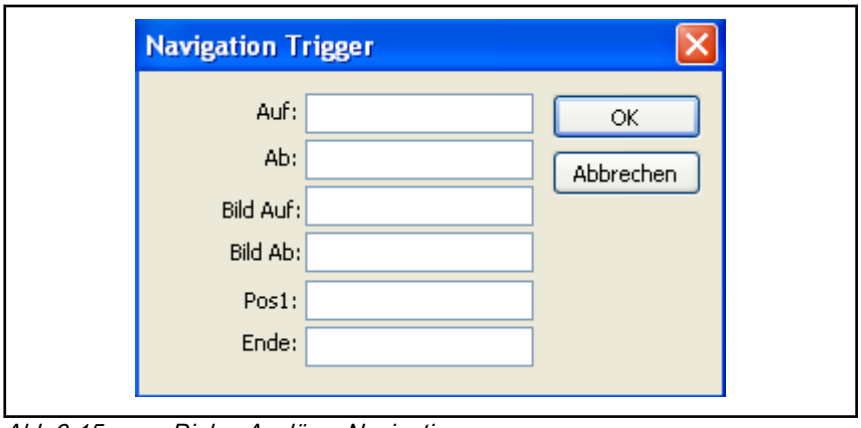


Abb.9-15: Dialog Auslöser Navigation

Durch Konfigurieren von Tags in den entsprechenden Feldern können für angezeigte Alarmobjekte die Funktionen nach oben rollen, nach unten rollen auf Seite nach oben auf Seite nach unten, zum Seitenanfang oder zum Seitenende gehen ausgeführt werden. Wenn sich die Werte der konfigurierten Tags ändern, navigiert das Alarmsteuerungsobjekt in diese Richtung. Dies ist hilfreich für das Hinzufügen von Navigationssteuerelementen zu einem Bild. Wenn zum Beispiel der gleiche Tag zu dem Feld **Nach oben** in diesem Dialog und zu einem Drucktastenobjekt auf Seite 193 konfiguriert wird, dann blättert das Alarmsteuerungsobjekt bei jedem Drücken des Drucktastenobjekts nach oben.

Zum Beenden klicken Sie auf **OK** zum Schließen des Dialogs "Weitere Einstellungen".

9.5 Trendarbeitsblatt

9.5.1 Trendarbeitsblatt, Allgemeines

Der Ordner "Trend" ermöglicht das Konfigurieren von Historiengruppen, die Trendkurven speichern. Sie können das Trendarbeitsblatt zum Deklarieren verwenden, welche Tags ihre Werte auf die Festplatte speichern, und zum Anlagen von Historiedateien für Trendgraphen. Ihr Projekt speichert die Stichproben in einer binären Historiedatei (*.hst) und zeigt die historischen und die Online-Stichprobenwerte in einem Trendgraph-Bild an.

Das Trendarbeitsblatt wird durch das Modul Hintergrundtask ausgeführt. Es verwaltet das Speichern von Trenddaten in der Historie, zeigt diese Daten aber nicht für den Bediener an. Zum Anzeigen von Trenddaten muss das Trendobjekt auf Seite 313 aus dem Werkzeugfenster "WinStudio Screen Tools" in ein WinStudio-Bild eingefügt und konfiguriert werden.

Zum Erstellen eines neuen Trendarbeitsblatts öffnen Sie das Kontextmenü des Ordners "Tasks" und wählen aus dem Menü "Hinzufügen" den Eintrag "Trend."

Zum Bearbeiten eines bestehenden Trendarbeitsblatts führen sie im Projekt-explorer auf ein Trendarbeitsblatt einen Doppelklick aus.

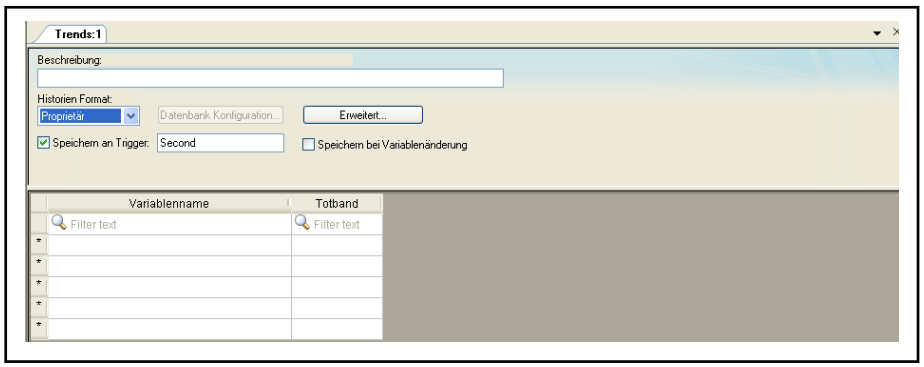


Abb.9-16: Trendarbeitsblatt

Das Trendarbeitsblatt gliedert sich in zwei Bereiche:

- Titelbereich (oberer Teil), der Informationen für die gesamte Gruppe enthält
- Hauptbereich (unterer Teil), in dem jeder Tag in der Gruppe definiert wird. Dieser Bereich enthält mehrere Spalten (in der obigen Abbildung sind nur zwei dargestellt).

Die Titel-Parameter auf diesem Arbeitsblatt werden wie folgt verwendet:

- Feld **Beschreibung**: Geben Sie zu Dokumentationszwecken eine Beschreibung für das Arbeitsblatt ein.
- **Historienformat**: Klicken Sie zum Auswählen eines Trendhistorienformats. Die folgenden Optionen sind verfügbar:
  - **Proprietär**
    - **Dateiformat:** Binär
    - **Standardpfad: …\projekt_name\Hst\GGJJTTMM.HST**, wobei:
      - **JJ** = die letzten beiden Stellen der Jahreszahl
      - **MM** = Monat
      - **TT** = Tag

☞ "WinStudio" stellt die Programme **HST2TXT.EXE** und **TXT2HST.EXE** zur Verfügung, mit denen Trendhistoriendateien von binär (**.hst**) in Text (**.txt**) konvertiert werden können und umgekehrt. Weitere Informationen über diese Programme unter Konvertieren von Trendhistoriendateien von binär in Text auf Seite 303 und Konvertieren von Trendhistoriendateien von Text in binär auf Seite 304.

- **Datenbank**
  - **Datenbanktyp:** Vom Benutzer gewählt
  - **Standard-Tabellenname:TRENDGGG** (**GGG** = Trendarbeitsblattnummer; z.B., **TREND001** für das Trendarbeitsblatt 001)

  Weitere Informationen zur Struktur der Datenbanktabelle, die "WinStudio" zum Speichern von Historiendateien verwendet, unter Datenbankschnittstelle auf Seite 493.

HINWEIS

In vielen Feldern des Trendarbeitsblatts können String-Tags zum Ändern dieser Werte während der Laufzeit festgelegt werden. Allerdings kann dies beeinflussen, wie diese Werte in der Trendhistorie gespeichert werden:

- Wenn das Historienformat auf **Proprietär** eingestellt ist, wird der Wert des String-Tags in einen numerischen Wert konvertiert (falls möglich) und dann in der Historiendatei gespeichert. Wenn die numerische Konvertierung nicht möglich ist, wird ein Nullwert gespeichert.
- Wenn das Historienformat auf **Datenbank** eingestellt ist, wird der tatsächliche Wert des String-Tags in der Datenbank gespeichert.

- **Datenbankkonfiguration**: Öffnet den Dialog Datenbankkonfiguration auf Seite 502, in dem die notwendigen Einstellungen zum Verknüpfen des Projekts mit einer externen relationalen SQL-Datenbank für das Speichern der Trendhistorie eingegeben werden können.
- Kontrollkästchen und Feld **Trigger für Speichern**: Dieses Kontrollkästchen anklicken (aktivieren) und einen Tag-Namen eingeben, um Trendbeispielwerte zu speichern, wenn der angegebene Tag geändert wird. (Die Tag-Änderung kann ein Ereignis aus dem Scheduler sein.)
- Kontrollkästchen **Speichern bei Variablenänderung**: Durch Anklicken (Aktivieren) wird der Trendbeispielwert immer gespeichert, wenn in einem der Tags aus dieser Gruppe eine Wertänderung stattfindet.
- **Erweitert**: Durch Klicken wird der Dialog "Erweiterte Einstellungen" angezeigt. Informationen zum Ausfüllen der Felder in diesem Fenster unter Stapelkonfiguration Historie auf Seite 305.

Die Haupt-Parameter auf diesem Arbeitsblatt werden wie folgt verwendet:

- Feld **Variablenname**: Geben Sie einen Variablennamen zum Speichern in der Historiendatei ein.
- Kontrollkästchen **Totband**: Geben Sie einen Wert zum Filtern annehmbarer Änderungen bei Verwendung von **Speichern bei Variablenänderung** ein.

  Zum Beispiel:

Das Totband hat einen Wert = 5. Wenn der Tag-Wert 50 ist und sich auf 52 ändert, registriert das System diese Änderung nicht in der Datenbank, weil sie kleiner als 5 ist. Wenn die Änderung größer oder gleich 5 ist, wird der neue Wert in der Historiendatei gespeichert.

- **Feld**: Name des Felds in der Datenbank, in der der Tag gespeichert wird. Wenn das Feld leer bleibt, wird der Name des Tags als Tag-Name verwendet. Bei Array-Tags und Klassen werden die Zeichen "[", "] " und " . " durch " _ " ersetzt.

  Beispiele:

Tag-Name	Standardfeld
MeinArray[1]	MeinArray_1
MeineKlasse.Member1	MeineKlasse_Member1
MeineKlasse[3].Member2	MeineKlasse_3_Member2

☞ Der Trend-Task kann nur maximal 240 Tags in einem einzigen Arbeitsblatt aufnehmen. Wenn manuell mehr als 240 Tags im gleichen Arbeitsblatt konfiguriert werden, erzeugt der Trend-Task einen Fehler beim Ausführen des fertigen Projekts.

Siehe auch:

- Konvertieren von Trendhistoriendateien von binär in Text auf Seite 303
- Konvertieren von Trendhistoriendateien von Text in binär auf Seite 304
- Stapelerstellung Historie auf Seite 305
- Konfigurieren einer Standarddatenbank für die gesamte Task-Historie auf Seite 508

9.5.2 Konvertieren von Trendhistoriendateien von binär in Text

Standardmäßig speichert "WinStudio" Historiendateien im Binärformat (**.hst**). Wenn diese Dateien im Format **.txt** benötigt werden, stellt "WinStudio" das Programm **HST2TXT.EXE** zum Konvertieren von Trendhistoriendateien von binär in Text zur Verfügung.

Zum Konvertieren einer Datei gehen Sie wie folgt vor:

1. Wechseln Sie aus einem DOS-Fenster heraus das Verzeichnis (**cd**) zum Verzeichnis **\Bin** von "WinStudio":

   cd Bin

2. In der Eingabeaufforderung kopieren Sie die Datei **Hst2txt.exe** in das gleiche Verzeichnis, in dem sich die **.hst**-Datei befindet.
3. Geben Sie in der Eingabeaufforderung **Hst2txt.exe** ein und legen Sie die folgenden Parameter fest:
   - **filename:** Name der zu konvertierenden Trendhistoriendatei
   - **[trennzeichen]:** Datentrennzeichen (Standard ist <TAB>)
   - **[/e]:** Erweiterte Funktionalität (Daten mit mehr als zehn Zeichen konvertieren)
   - **[/i:HH:MM:SS]:** Startzeit in Stunden (**HH**), Minuten (**MM**) und Sekunden (**SS**)

- **[/i:HH:MM:SS]:** Endzeit in Stunden (**HH**), Minuten (**MM**) und Sekunden (**SS**)
- **[/m]:** Millisekunden in die Spalte **Zeit** aufnehmen (Typ **1** zum Drucken des Millisekunden-Werts in die aus der **.hst**-Datei erstellte Textdatei)

  Zum Beispiel:

  Hst2txt.exe 01952010.hst

  Das Programm erstellt eine **.hdr**-Datei (Titelbereich) und die **.txt**-Datei. Beide sind einfache Textdateien, die mit einem Texteditor angezeigt werden können (zum Beispiel Notepad).
- Die **.hdr**-Datei enthält den Namen des im Trendarbeitsblatt konfigurierten Tags.
- Die **.txt**-Datei enthält die in der Historiendatei gespeicherten Tag-Werte.

4. Wenn das Programm die Datei konvertiert hat, geben Sie zum Schließen des DOS-Fensters "Exit" ein.

☞ Alternativ kann das Berechnungs-Skript HST2TXT auf Seite 667 in einem Mathematik-Arbeitsblatt auf Seite 367 zum automatischen Konvertieren von binären Dateien in Textformat verwendet werden, ohne Verwendung eines DOS-Fensters.

Siehe auch:

- Konvertieren von Trendhistoriendateien von Text in binär auf Seite 304
- Stapelerstellung Historie auf Seite 305
- Konfigurieren einer Standarddatenbank für die gesamte Task-Historie auf Seite 508

9.5.3 Konvertieren von Trendhistoriendateien von Text in binär

Zum Konvertieren von Textdateien zurück in das binäre Format stellt "WinStudio" das Programm **TXT2HST.EXE** zur Verfügung.

Zum Konvertieren einer Datei gehen Sie wie folgt vor:

1. Wechseln Sie aus einem DOS-Fenster heraus das Verzeichnis (**cd**) zum Verzeichnis **\Bin** von "WinStudio":

   cd Bin
2. In der Eingabeaufforderung kopieren Sie die Datei **Txt2hst.exe** in das gleiche Verzeichnis, in dem sich die **.txt**-Datei befindet.
3. Geben Sie in der Eingabeaufforderung **Txt2hst.exe** ein und legen Sie die folgenden Parameter fest:
   - **filename:** Name der ASCII-Datei mit den zu konvertierenden Historiendaten
   - **[trennzeichen]:** Datentrennzeichen (Standard ist <TAB>)
   - **[/e]:** Erweiterte Funktionalität (Datenwerte mit mehr als 10 Zeichen konvertieren)
   - **[/i:HH:MM:SS]:** Startzeit des Datenwerts in Stunden (**HH**), Minuten (**MM**) und Sekunden (**SS**)
   - **[/i:HH:MM:SS]:** Endzeit des Datenwerts in Stunden (**HH**), Minuten (**MM**) und Sekunden (**SS**)

     Zum Beispiel:

Txtt2hst.exe 02950201.txt

Das Programm erstellt eine **.hdr**-Datei ("Kopfbereich") und konvertiert die **.txt**-Datei in eine binäre **.hst**-Datei.

4. Wenn das Programm die Datei konvertiert hat, geben Sie zum Schließen des DOS-Fensters "Exit" ein.

☞ Es kann kein Berechnungsskript für das Programm **TXT2HST.EXE** erstellt und in einem Mathematik-Arbeitsblatt auf Seite 367 zum Konvertieren von Textdateien in das binäre Format verwendet werden, wie es für **HST2TXT.EXE** möglich ist. Die Abkürzung über ein Berechnungsskript steht nur für binäre Dateien zur Verfügung.

Siehe auch:

- Konvertieren von Trendhistoriendateien von binär in Text auf Seite 303
- Stapelerstellung Historie auf Seite 305
- Konfigurieren einer Standarddatenbank für die gesamte Task-Historie auf Seite 508

9.5.4 Stapelerstellung Historie

"WinStudio" stellt leistungsfähige Werkzeuge zur Verfügung, mit denen der Benutzer historische Stapelinformationen erstellen und verwalten kann. Der Benutzer kann durch Verwendung der folgenden Formate Stapel erstellen:

- **Proprietär**: Bei Verwendung des proprietären Formats wird jeder Stapel in einer anderen Historiendatei gespeichert. Der Benutzer kann historische Daten gleichzeitig in der normalen Historiendatei und in Stapeldateien speichern (weitere Informationen über diese Dateien unter Ordner Trend auf Seite 301).
- **Datenbank**: Die für den Stapel verwendeten historischen Daten werden in der gleichen Tabelle gespeichert wie die normalen historischen Daten; eine zusätzliche Tabelle mit der Bezeichnung StapelHistorie enthält Register mit den Informationen über die Stapel. Die folgende Liste beschreibt die Felder in der Tabelle Stapel-Historie:

Feldname	Datentyp	Beschreibung
Group_Number	Integer	Nummer der Trendgruppe. Dies ist die Nummer des Arbeitsblatts, das zum Festlegen der Tags erstellt wird, die in Ihrer Stapelhistorie gespeichert werden.
Batch_Name	String	Name des Stapels
Start_Time	Zeitstempel	Datum und Uhrzeit, wann der Stapel begonnen wurde.
End_Time	Zeitstempel	Datum und Uhrzeit, wann der Stapel beendet wurde.
Pri_Table	String	Reserviert
Sec_Table	String	Reserviert
Description	String	Stapelbeschreibung
Deleted	Boolean	0: Stapel wurde nicht gelöscht 1: Stapel wurde gelöscht

Die Namen der in der Datenbank erstellten Tabelle und der Spalten können durch Bearbeiten der Datei **projekt_name.APP** wie folgt angepasst werden:

[Trend]
StandardName=NeuerName
[TrendGroupPRI|SEC]
BatchHistory=Tabellenname

Zum Beispiel:
[TREND001PRI]
BatchHistory=MeineTabelleFürPrimärDB

[TREND001SEC]
BatchHistory=MeineTabelleFürSekundärDB

[Trend]
Group_Number=Trend_Arbeitsblatt
Batch_Name=Laden_Number

Konfiguration der Stapelhistorie

Wenn ein Trendarbeitsblatt (siehe Ordner Trend auf Seite 301) hinzugefügt und auf die Schaltfläche Weitere Einstellungen geklickt wird, erscheint das folgende Fenster:



Abb.9-17: Erweiterte Einstellungen Trend

Im Bereich **Stapel** kann das Speichern der Stapelhistorie konfiguriert werden:

- **Start/Stop (Eingabe):** Geben Sie den Tag zum Starten/Beenden Ihrer Stapel ein. Wenn der Tag in diesem Feld auf WAHR gesetzt ist (nicht 0), beginnt "WinStudio" entweder mit dem Speichern von Daten in der

Stapeldatei (bei Verwendung des proprietären Formats) oder fügt der Tabelle StapelHistorie in Ihrer Datenbank ein neues Register hinzu zur Anzeige, dass ein neuer Stapel begonnen wurde. Bitte beachten Sie, dass historische Daten gemäß der Konfiguration in den Feldern **Auslöser Speichern.** und **Speichern bei Variablenänderung** auf dem Trendarbeitsblatt gespeichert werden.

- **Name (Eingabe):** Dieses Feld stellt den Stapelnamen dar; die Bedeutung hängt von dem auf dem Trendarbeitsblatt ausgewählten Format ab:
  - Wenn im Feld **Typ proprietär** ausgewählt wurde, sollte **Name** dem Format **[Pfad]<Dateiname>**, wobei gilt:
    - **Pfad**: Optionales Feld. Wenn kein Pfad angegeben wird, wird die Stapelhistoriendatei unter dem gleichen Pfad wie die Datei **projekt_name.app** gespeichert.
    - **DateiName**: Name der Stapelhistoriendatei.
  - Wenn im Feld **Typ Datenbank** ausgewählt wurde, wird der Wert in diesem Feld im Feld **Stapel_Name** der Tabelle "StapelHistorie" gespeichert.



In diesem Feld können Tag-Namen in geschweiften Klammern eingegeben werden

(z.B. **C:\MeineStapel\{MeinTagMitName}{MeinTagMitNummer}.hst**).

- **Löschen (Eingabe):** Wenn der in diesem Feld angegebene Tag seinen Wert ändert, wird der Stapel gelöscht. Bei Format **Proprietär** wird die Stapelhistoriendatei entfernt. Bei Format **Datenbank** wird das Feld **Löschen** in der Tabelle "StapelHistorie" auf wahr gesetzt, die gespeicherten historischen Daten bleiben aber erhalten. Das Trendobjekt erkennt nur Stapel, bei denen das Löschen-Feld auf "0" (Null) gesetzt ist.
- **Vorhanden (Ausgabe)**: Der in diesem Feld eingegebene Tag erhält den Wert "1", wenn der im Feld **Name** festgelegte Stapel bereits besteht; andernfalls erhält der Tag den Wert "0".
- **Beschreibung (Ausgabe)**: Dieses Feld ist nur verfügbar, wenn das Format **Datenbank** verwendet wird. Wenn sich der Tag im Feld **Start/Stopp** auf WAHR ändert, zeigt das der Tabelle "StapelHistorie" hinzugefügte Register den String in diesem Feld an.



In diesem Feld können Tag-Namen in geschweiften Klammern eingegeben werden (z.B. **{MeinTag}**).

- **Daten speichern, auch wenn Stapel nicht läuft**: Wenn dieses Feld nicht markiert ist, werden die historischen Daten nur gespeichert, wenn der Tag im Feld **Start/Stopp** WAHR ist.



Die historischen Stapeldaten können dem Benutzer im grafischen oder im Tabellenformat angezeigt werden.

Siehe Ordner Trend auf Seite 301 oder Rasterobjekt auf Seite 345 zur Anzeige von Informationen in diesen Formaten.

Im Bereich **Steuerung Festplattenspeicher** kann die Festplattenbelegung gesteuert werden:

- Feld **Löschen nach Zeit (Tagen)**: Geben Sie ein, wie viele Tage die Historiendatei auf der Festplatte verbleiben soll. Nach dem angegebenen Zeitraum löscht "WinStudio" die Datei automatisch. Verwenden Sie diese Option nur für datumsbasierte Dateien.
- Feld **Komprimieren nach (Tagen)**: Geben Sie ein, wie viele Tage die Trendhistoriendatei (***.hst**) auf der Festplatte verbleiben soll, bevor sie komprimiert wird. Nach dem angegebenen Zeitraum komprimiert "WinStudio" die Datei automatisch. Verwenden Sie diese Option nur für datumsbasierte Dateien. Diese Option steht für Windows Embedded Zielsysteme nicht zur Verfügung.

Im Bereich **Qualität schlecht** kann festgelegt werden, welcher Wert tatsächlich in der Stapelhistorie gespeichert wird, wenn die Tag-Qualität und schlecht (BAD) ist.

Typ	Beschreibung
Tag-Wert	Der tatsächliche Wert des Projekt-Tags, wenn die Tag-Qualität schlecht war, zuzüglich des ggf. angegebenen **Offset**.
Min. Wert	Der minimale historische Wert des Projekt-Tags, abzüglich des ggf. angegebenen **Offset**.
Max. Wert	Der maximale historische Wert des Projekt-Tags, zuzüglich des ggf. angegebenen **Offset**.
Wert	Nur der angegebene **Wert**.
NaN	Not a Number (keine Zahl). Bitte beachten Sie: Wenn das **Historienformat Datenbank** ist und **Schlechte Qualität NaN** ist, werden alle Datenbankfelder als Float-Typ gespeichert. Wenn außerdem ein Trendobjekt zur Verwendung der durch dieses Trendarbeitsblatt erzeugten Historie konfiguriert wurde, zählen NaN-Eingaben bei der Berechnung von statistischem Durchschnitt und Abweichung eines Trends als 0.



Die Option **Qualität schlecht** kann bei Projekten, die auf Windows Embedded Zielsystemen ausgeführt werden, nicht verwendet werden.

Geben Sie im Feld **Gesamte Datenspeicherung deaktivieren** den Namen eines Tags ein. Wenn der Wert des Tags während der Laufzeit WAHR ist (nicht null), wird die Datenspeicherung für dieses Arbeitsblatt deaktiviert. Andere Trendarbeitsblätter sind nicht betroffen.

9.5.5 Trenddatenbank einstellen

1. Öffnen Sie ein Trendarbeitsblatt indem Sie auf ein Element im Ordner "Trends" einen Doppelklick ausführen.

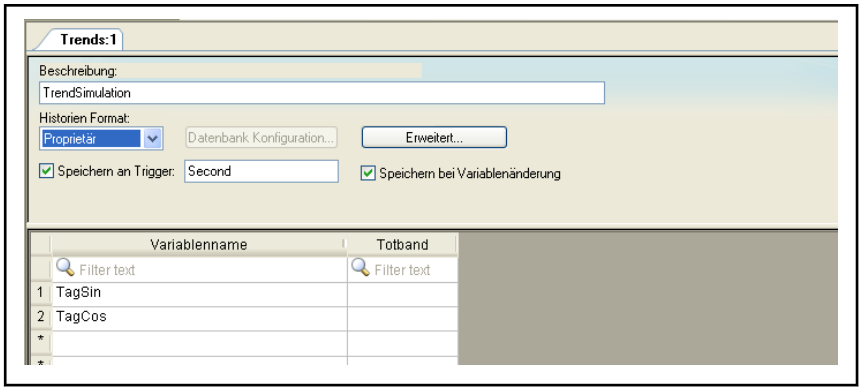


Abb.9-18: Beispiel für Trendarbeitsblatt

2. Klicken Sie auf den Pfeil rechts neben dem Feld **Historienformat**. Wählen Sie **Datenbank**.
3. Klicken Sie auf die Schaltfläche **Datenbankkonfiguration**. Der Dialog Datenbankkonfiguration auf Seite 502 wird geöffnet. Geben Sie die entsprechenden Daten in diesem Fenster ein. Klicken Sie auf **OK**, wenn Sie fertig sind.
4. Klicken Sie auf das Auswahlmenü **Datei** und dann auf **Speichern** zum Speichern der Änderungen am "Trend".

Trendtabelle auf einer relationalen Datenbank

Die in der Trendhistorie gespeicherten Felder werden in der folgenden Tabelle beschrieben:

Feldname	Datentyp	Bemerkungen
Time_Stamp	Zeitstempel	Zeitstempel (Datum und Uhrzeit), wann die Daten gespeichert wurden.
<TagName>	Integer oder Real (je nach Tag-Typ)	"WinStudio" erstellt für jeden im Trendarbeitsblatt konfigurierten Tag ein Feld (eine Spalte) in der Datenbank.

9.6 Trendanzeige

9.6.1 Trendanzeige, Allgemeines

Die wichtigsten Funktionen des Trendobjekts sind:

- Gleichzeitige Anzeige verschiedener Trendkurven
- Unterstützung für verschiedene Datenquellen, zum Beispiel Tag, Stapel, Datenbank und Textdatei
- Möglichkeit zum Erzeugen von X/Y-Graphen aus den konfigurierten Datenquellen (siehe Anhang A auf Seite 335 für ein Beispiel eines X/Y-Graphen).
- Gleichzeitige Anzeige einer unbegrenzten Anzahl von Datenpunkten. Diese Funktion kann aufgrund unterschiedlicher Speicherkapazität und Leistungsfähigkeit durch die Hardware begrenzt sein.
- Eingebaute Symbolleiste, mit deren Schnittstellen der Benutzer während der Laufzeit mit dem Trend Steuerungsobjekt arbeiten kann.
- Eingebaute Legende, die die wichtigsten Informationen für jeden mit dem Objekt verknüpften Stift anzeigt.
- Werkzeuge für Vergrößern/Verkleinern und automatisches Skalieren
- Horizontale und vertikale Lage

Alarme, Ereignisse und Trends

9.6.2 Bedienelemente der Trendanzeige

Während der Laufzeit verfügt eine Trendanzeige über eine eigene eingebaute Bedienelemente mit deren Hilfe der Benutzer die Anzeige von Trends verändern kann. Nachstehend werden die wichtigsten Bedienelemente und deren Verwendung beschrieben.

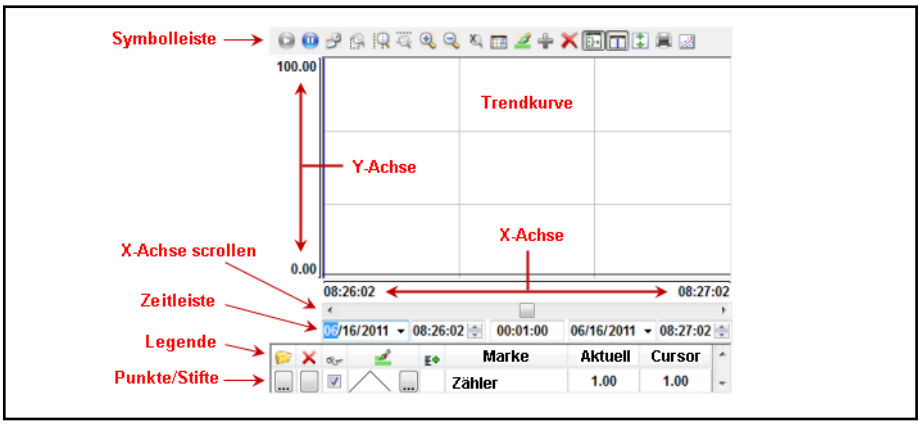


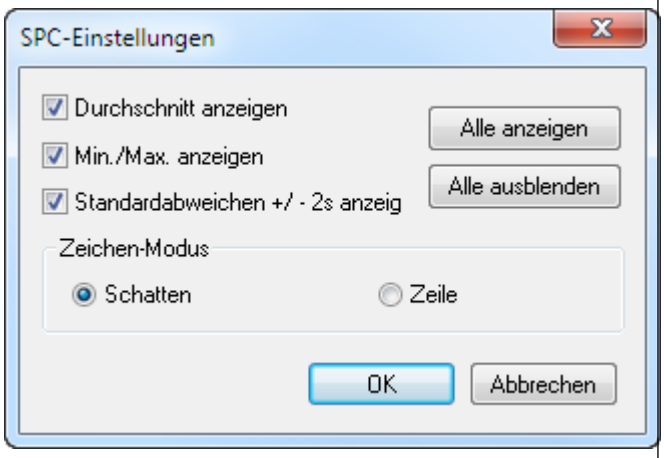
Abb.9-19: Bedienelemente der Trendanzeige

Symbolleiste

Befehl	Symbol	Beschreibung	Aktivierungs-Tag
Ausführen		Setzt die Trendobjekt in den Ausführungsmodus (auch Online-Modus genannt). In diesem Modus läuft die x-Achse mit dem Zeitverlauf mit und die Trends werden mit aktuellen Tag-Werten aktualisiert.	0 = Ausführungsmodus ein 1 = Ausführungsmodus aus
Stopp		Setzt die Trendanzeige in den Stoppmodus (auch Historischer Modus genannt). In diesem Modus wird die x-Achse angehalten und die Trends zeigen nur historische Daten an. Wenn für einen oder mehrere Trends Dezimierung aktiviert ist, werden Berechnung und Neuzeichnen nur in diesem Modus ausgeführt.	0 = Stoppmodus ein 1 = Stoppmodus aus
Periode		Öffnet einen Dialog, mit dem die wichtigsten Einstellungen für die Skala der X-Achse verändert werden können.	Dieser Befehl wird ausgeführt, wenn der Aktivierungs-Tag seinen Wert ändert (z.B. umschaltet).
Fenster Zoom		Ermöglicht es dem Benutzer, auf den Trendgraphen zu klicken und mit dem Cursor den Bereich auszuwählen, der bei Loslassen das Cursors sichtbar sein muss. Diese Option ist deaktiviert, wenn die Option **Mehrfachabschnitt** (für die Y-Skale) aktiv ist.	
Zoom horizontal		Ermöglicht es dem Benutzer, die verfügbare horizontale Skala durch Klicken auf zwei Punkte im Trendgraphen zu definieren.	
Zoom vertikal		Ermöglicht es dem Benutzer, die verfügbare vertikale Skala durch Klicken auf zwei Punkte im Trendgraphen zu definieren. Diese Option ist deaktiviert, wenn die Option **Mehrfachabschnitt** (für die Y-Skale) aktiv ist.	
Vergrößern		Bei jedem Klicken auf den Trendgraphen wird der Ausschnitt vergrößert (Anzeige der Hälfte der aktuellen X- und Y-Skalen).	0 = Vergrößern ein 1 = Vergrößern aus

Befehl	Symbol	Beschreibung	Aktivierungs-Tag
Verkleinern		Bei jedem Klicken auf den Trendgraphen wird der Ausschnitt verkleinert.	0 = Verkleinern ein 1 = Verkleinern aus
Vergrößern/ Verkleinern abbrechen		Bricht die aktuelle **horizontale**, **vertikale** oder **Fenster**-Vergrößerung ab und setzt den Trendgraphen auf seine ursprüngliche Skale zurück.	Dieser Befehl wird ausgeführt, wenn der Aktivierungs-Tag seinen Wert ändert (z.B. umschaltet).
Eigenschaften Legende		Öffnet einen Dialog, mit dem die wichtigsten Einstellungen für die Legende verändert werden können.	
Stiftstil		Öffnet einen Dialog, mit dem der Stiftstil des gewählten Trends verändert werden kann.	
Stift hinzufügen		Öffnet einen Dialog, mit dem der Trendanzeige ein neuer Trend hinzugefügt werden kann.	
Stift entfernen		Entfernt den ausgewählten Trend aus der Trendanzeige.	
Mehrfachabschnitte		Schaltet die Y-Skale zwischen Mehrfachabschnitten (ein Abschnitt für jeden Trend) und Einzelabschnitt (alle Trends teilen sich den gleichen Y-Skalenabschnitt) um.	0 = Mehrfachabschnitte ein 1 = Mehrfachabschnitte aus
Cursor		Schaltet den Cursor (Lineal) zwischen sichtbar und unsichtbar um.	0 = Cursor ein 1 = Cursor aus
Autom. skalieren		Ändert die Skale der Y-Achse so, dass alle Werte aus aktuell überwachten Trends hinein passen.	Dieser Befehl wird ausgeführt, wenn der Aktivierungs-Tag seinen Wert ändert (z.B. umschaltet).

Alarme, Ereignisse und Trends

Befehl	Symbol	Beschreibung	Aktivierungs-Tag
Drucken		Druckt den aktuellen Zustand der Trennsteuerung. (Historische Daten werden nicht gedruckt.)	Dieser Befehl wird ausgeführt, wenn der Aktivierungs-Tag seinen Wert ändert (z.B. umschaltet).
SPC		Öffnet einen Dialog zur Anzeige der Informationen aus der statistischen Prozesssteuerung (statistical process control, SPC) für den ausgewählten Trend:  *Modus Zeichnen...* • **Schattierung**: Zeichnet den Mittelwert als gestrichelte Linie und zeichnet die Min/Max-Werte und die Standardabweichung als schattierte Bereiche. **Linie**: Zeichnet den Mittelwert und die Standardabweichung als gestrichelte Linien und zeichnet die Min/Max-Werte als durchgehende Linien. • **Durchschnitt anzeigen**: Berechneten Durchschnitt aller historischen Trendwerte anzeigen. **Hinweis:** Wenn ein Wert keine Zahl ist (NaN) – zum Beispiel wenn ein Tag als schlechte Qualität markiert ist – zählt er für die Berechnung des Durchschnitts als 0. • **Standardabweichung anzeigen**: Zeigt die Standardabweichung des Trends an. Eine geringe Standardabweichung zeigt an, dass der tatsächliche Wert sich nah am Durchschnitt bewegt; eine hohe Standardabweichung zeigt an, dass der tatsächliche Wert stark vom Durchschnitt abweichend schwankt. • **Min/Max anzeigen**: Zeigt die historischen Mindest- und Maximalwerte des Trends an.	Dieser Befehl wird ausgeführt, wenn der Aktivierungs-Tag seinen Wert ändert (z.B. umschaltet). Außerdem können die Bit-Eigenschaften des Tags (B0-B4) zur Vorauswahl von Optionen im Dialog verwendet werden: • **tagname->B0** 0 = Dialog schließen 1 = Dialog öffnen • **tagname->B1** 0 = **Modus Zeichnen: Linie** ausgewählt 1 = **Modus Zeichnen: Schattierung** ausgewählt • **tagname->B2** 0 = **Durchschnitt anzeigen** gelöscht 1 = **Durchschnitt anzeigen** ausgewählt • **tagname->B3** 0 = **Standardabweichung anzeigen** gelöscht 1 = **Standardabweichung anzeigen** ausgewählt • **tagname->B4** 0 = **Min/Max anzeigen** gelöscht 1 = **Min/Max anzeigen** ausgewählt

☞ Aktivierung-Tags werden in den Objekteigenschaften des Trendobjekts konfiguriert. Weitere Informationen unter Symbolleiste auf Seite 325.

Legende

Befehl	Symbol	Beschreibung
Auswahl		Öffnet einen Dialog, in dem der Benutzer den mit dem auf der Legende ausgewählten Trend verbundenen Datenpunkt ersetzen kann.
Entfernen		Entfernt den ausgewählten Trend aus der Trendanzeige.
Verbergen		Wenn diese Option aktiviert ist, ist der ausgewählte Trend sichtbar, andernfalls ist er verborgen.
Stiftstil		Öffnet einen eingebetteten Dialog, in dem der Benutzer den Stiftstil des gewählten Trends ändern kann.
Skala		Wenn diese Option aktiviert ist, ist die Skala der y-Achse sichtbar, andernfalls ist sie verborgen. Die Skala kann nur verborgen werden, wenn die Option Mehrfachabschnitte ausgeschaltet ist.

9.6.3 Eigenschaften eines Trendobjekts

Eigenschaften eines Trendobjekt, Allgemeines

Der Dialog "Objekteigenschaften: Trendobjekt" wird zum Konfigurieren der grundlegenden Eigenschaften eines Trendobjekts verwendet.

Zugriff auf den Dialog

Für den Zugriff auf den Dialog "Objekteigenschaften" für ein bestimmtes Bildobjekt gehen Sie wie folgt vor:

- Öffnen Sie das WinStudio-Bild mit dem zu konfigurierenden Trendobjekt im Editor
- führen Sie einen Rechtsklick auf das Bildobjekt aus und wählen dann **Eigenschaften** aus dem Kontextmenü oder
- führen Sie einen Doppelklick auf das Bildobjekt aus.

Der Dialog im Detail

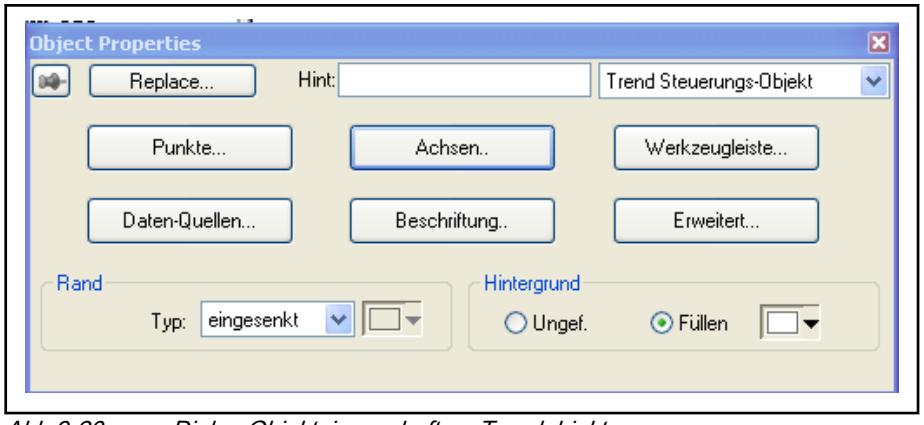


Abb.9-20: Dialog Objekteigenschaften: Trendobjekt

Zusätzlich zu den in allen Objekteigenschaftsdialogen gleichen Elementen, enthält der Dialog "Objekteigenschaften: Trendobjekt" die folgenden Elemente:

Bereich / Elementname		Beschreibung
Rand	**Typ**	Bestimmt die Rahmenart um den Bereich des Graphen in der Trendanzeige. (Keine Rahmen um die Legende oder Symbolleiste der Trendanzeige)
	Farbe	Bestimmt die Rahmenfarbe, wenn als Rahmentyp **Durchgehend** gewählt ist. Weitere Informationen unter Auswählen von Farben und Fülleffekten.

Alarme, Ereignisse und Trends

Bereich / Elementname		Beschreibung
Hintergrund	**Keine Füllung / Füllung**	Aktiviert die Hintergrundfüllung für den Bereich des Graphs in der Trendanzeige. (Keine Hintergründe für Legende oder Symbolleiste der Trendanzeige.) Wenn Sie nicht aktiviert ist, ist der Graph transparent für alle Bildobjekte hinter der Trendanzeige.
	Farbe	Bestimmt Farbe und Fülleffekt für die Hintergrundfüllung, falls aktiviert. Weitere Informationen unter Auswählen von Farben und Fülleffekten.
Punkte		Öffnet den Dialog Punkte, mit dem die Datenpunkte (oder Stifte) der Trendanzeige konfiguriert werden können. Weitere Informationen unter Dialog Trendobjekt: Dialog Punkte auf Seite 314
Achsen		Ermöglicht die Konfiguration der x- und y-Achsen der Trendanzeige sowie deren horizontale oder vertikale Lage. Weitere Informationen unter Dialog "Trendanzeige: Achsen"
Symbolleiste		Ermöglicht die Konfiguration der Benutzersymbolleiste, die über der Trendanzeige dargestellt wird. Weitere Informationen unter Dialog "Trendanzeige: Symbolleiste"
Datenquellen		Ermöglicht die Konfiguration von mehrfachen Datenquellen für die Trendanzeige. Weitere Informationen unter Dialog "Trendanzeige: Datenquellen"
Legende		Ermöglicht die Konfiguration der Legende, die unter der Trendanzeige dargestellt wird. Weitere Informationen unter Dialog "Trendanzeige: Legende"
Weitere Einstellungen		Ermöglicht die Konfiguration der erweiterten Einstellungen der Trendanzeige, z.B. Laufzeitoptionen und Tag-Auslöser. Weitere Informationen unter Dialog "Trendanzeige: Weitere Einstellungen"

Obwohl die Trendobjekte flexible Konfigurationen unterstützen, um den speziellen Bedürfnissen Ihres Projekts gerecht zu werden, werden die meisten Einstellungen auf Grundlage der häufigsten Funktionen standardmäßig vorgenommen. Daher werden in vielen Fällen nur (während der Laufzeit angezeigte) Datenpunkte konfiguriert, was ganz einfach durch Klicken auf die Schaltfläche **Punkte** aus dem Fenster "Objekteigenschaften" erfolgt.

Punkte

Dialog Punkte, Allgemeines

Der Dialog "Punkte" dient der Konfiguration der Datenpunkte für das Bildobjekt Trendanzeige. Der Wert jedes Datenpunkts wird in der Trendanzeige durch einen Stift dargestellt. Welche Datenpunkte sichtbar sind, kann während der Laufzeit dynamisch verändert werden, ungeachtet der Anzahl der mit dem Bildobjekt verbundenen Datenpunkte.

Zugriff auf den Dialog

Um auf den Dialog "Punkte" für ein bestimmtes Bildobjekt Trendanzeige zuzugreifen, wählen Sie zunächst den Dialogobjekteigenschaften für dieses Bildobjekt auf Seite 313 aus und klicken dann auf **Punkte**.

Der Dialog im Detail

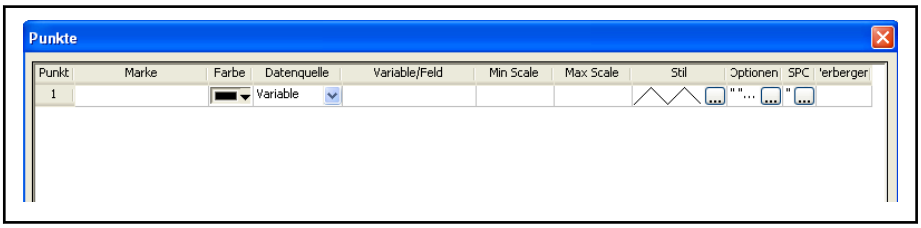


Abb.9-21: Dialog Objekteigenschaften: Trendanzeige

In der folgenden Tabelle sind die Eigenschaften für jeden Datenpunkt zusammengefasst:

Spaltenname	Beschreibung
Punkt	Eine einmalige Kennnummer für den Punkt, die automatisch beim Erstellen des Punkts in dieser Schnittstelle zugewiesen wird.
Beschriftung	Die Beschriftung für den Punkt kann während der Laufzeit in der Legende angezeigt werden und gibt dem Benutzer einen kurzen Bezug für jeden Punkt.
Farbe	Verwendete Stiftfarbe zum Zeichnen der Werte des Punkts in der Trendsanzeige.
Datenquelle	Die Datenquelle für diesen Punkt. **Tag** ist standardmäßig verfügbar. Alle anderen Quellen müssen im Dialog "Datenquellen" konfiguriert werden.
Tag/Feld	Die Bedeutung dieses Parameters ist abhängig von der Art der mit dem Datenpunkt verbundenen **Datenquelle**. • **Tag**: Geben Sie den Namen des Tags mit anzuzeigenden Werten ein. Wenn der Tag im Trend-Task konfiguriert ist, werden die historischen Daten automatisch abgerufen; andernfalls werden die Online-Werte angezeigt. • **Stapel**: Geben Sie den Namen des Tags ein, dessen Werte aus der von dem Trend-Task erzeugten Stapelhistoriendatei abgerufen und auf dem Objekt angezeigt werden sollen. • **Datenbank**: Geben Sie den Namen des Feldes (der Spalte) in der relationalen SQL-Datenbank ein, die die Datenpunktwerte enthält. • **Textdatei**: Geben Sie die Anzahl der Spalten ein, die die Punktwerte enthalten. Die Anzahl 0 bezieht sich auf die erste Spalte, 1 bezieht sich auf die zweite Spalte usw.
Min Skala / Max Skala	Die Skala der y-Achse für diesen Punkt. Diese hat Vorrang vor der Standardskala, die im Dialog Achsen auf Seite 320 festgelegt ist. **Hinweis:** Die Eigenschaften **Min Skala** und **Max Skala** können numerische Real-Werte mit bis zu sechs Dezimalstellen enthalten. Wenn eine größere Genauigkeit erforderlich ist, müssen Sie die Eigenschaften **Min Skala** und **Max Skala** mit Real-Tags konfigurieren und dann die Werte in diesen Tags speichern.
Stil	Die Linien- und Markierungsart für diesen Punkt. Klicken Sie auf die … Schaltfläche, um den Dialog Stiftstil auf Seite 315 zu öffnen.
Optionen	Zusätzliche Optionen für diesen Punkt; klicken Sie auf die … Schaltfläche zum Öffnen des Dialogs Optionen auf Seite 317.
SPC	Berechnete Statistiken zur Verwendung in der statistischen Prozesssteuerung (SPC). Klicken Sie auf die … Schaltfläche, um den Dialog SPC auf Seite 319 zu öffnen.
Verbergen	Tag-Auslöser – wenn der Wert WAHR ist, ist der Datenpunkt in der Trendanzeige verborgen.

Stiftstil

Mit diesem Dialog kann während der Laufzeit den Stiftstil zum Zeichnen der Datenpunktwerte auf dem Objekt konfiguriert werden.

Zugriff auf den Dialog

Um den Dialog "Stiftstil" für einen bestimmten Datenpunkt zu öffnen, öffnen Sie zunächst den Dialog "Punkte" und klicken dann auf die Spalte **Stile** für diesen Datenpunkt.

Er kann außerdem während der Laufzeit gestartet werden, damit der Benutzer diese Einstellungen fliegend bearbeiten kann.

Alarme, Ereignisse und Trends

Der Dialog im Detail

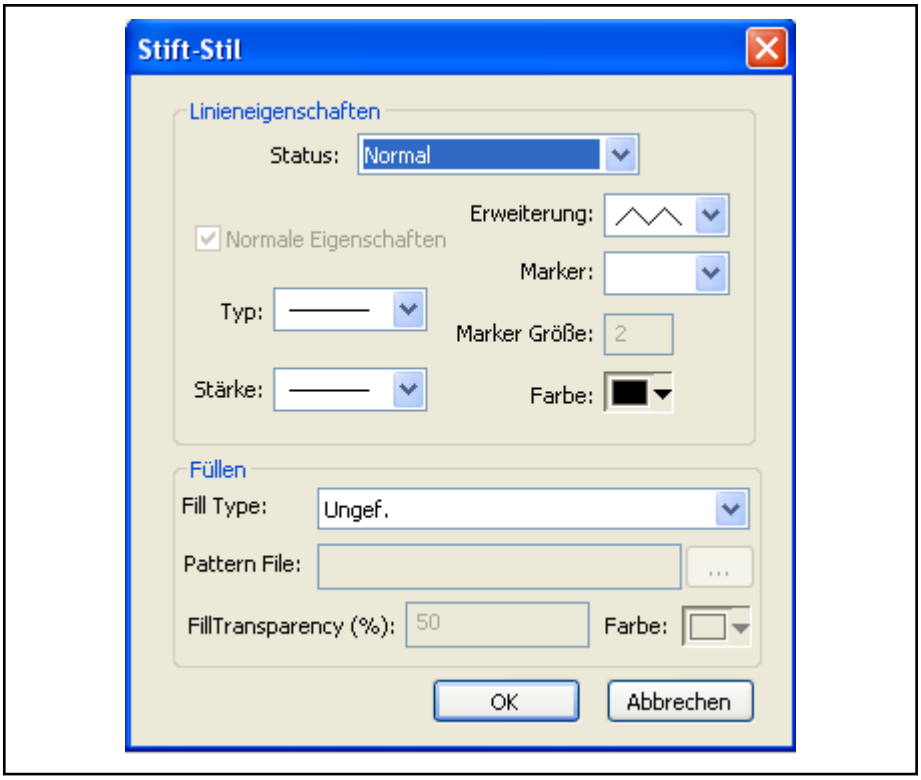


Abb.9-22: Dialog Stiftstil

Der Dialog "Stiftstil" enthält die folgenden Elemente:

Elemente im Dialog "Stiftstil"

Bereich / Elementname		Beschreibung
Linieneigenschaften	**Zustand**	Option zum Definieren eines Hi- und eines Lo-Limits für jeden Datenpunkt, mit dem Dialog Optionen. Der Dialog Stiftstil ermöglicht das Konfigurieren verschiedener Einstellungen für den Stift (z.B. Farbe), sowohl für Werte innerhalb der Grenzwerte (Normalzustand) als auch für Werte außerhalb der Grenzwerte (Grenzwert-Überschreitungszustand).
	Normaleinstellungen verwenden	Nur für den Zustand **Grenzwertüberschreitung** verfügbar. Wenn diese Option aktiviert ist, wird der Stift immer mit den Einstellungen für den Normalzustand angezeigt, auch wenn die Datenpunktwerte sich nicht innerhalb der konfigurierten Werte befinden.
	Typ	Die Art der Linie (z.B. durchgehend, gestrichelt, punktiert), die die Datenpunkte verbindet.
	Stärke	Die Stärke der Linie, die die Datenpunkte verbindet.
	Erweiterung	Der zum Verbinden der Punkte verwendete Algorithmus, wie folgt: •  : Aufeinanderfolgende Punkte werden durch eine analoge Linie direkt miteinander verbunden. Diese Option ist für numerische Werte geeignet. •  : Aufeinanderfolgende Punkte werden nur durch horizontale oder vertikale Schritte verbunden (abhängig von der Lage der Trendanzeige). Diese Option ist für Boolesche Werte geeignet.
	Marker	Die Form zum Markieren jedes Datenpunkts. Wenn keine Form ausgewählt wird, wird nur die Verbindungslinie zwischen den Punkten angezeigt.
	Markergröße	Die Größe der Datenpunktmarkierung.
	Farbe	Die Farbe der Trendlinien- und Datenpunktmarkierungen.

Bereich / Elementname		Beschreibung
Fülleffekt	**Füllart**	Die Art der Füllung zwischen Trendlinie und Zahlenlinie.
	Musterdatei	Die Grafikdatei, die zum Füllen des Trendbereichs verwendet wird. Nur verfügbar, wenn Füllart auf **Benutzerdefiniertes Muster** eingestellt ist. Klicken Sie auf die Suchen-Schaltfläche, um einen Windows-Browser zu öffnen und die gewünschte Grafikdatei zu öffnen. Die Datei sollte sich in Ihrem Projektordner befinden. Nachfolgend ein Beispiel für Trends mit benutzerdefinierten Füllmustern.
	Farbe	Die Farbe, die zum Füllen des Trendbereichs verwendet wird. Nur verfügbar, wenn Füllart auf **Einfarbig** eingestellt ist.
	Fülltransparenz (%)	Die Transparenzstufe der Füllung. (Wenn die Füllung transparent ist, sind die dahinterliegenden Trends sichtbar und der gesamte Graph wird leichter lesbar.) Verfügbar für **Benutzerdefiniertes Muster** und **Einfarbig**.

☞ Beim Anzeigen Ihres Projekts auf einem Windows-Embedded-Gerät oder einem Thin Client (alle Betriebssystems) kann der Benutzer im Dialog Stiftstil – während der Laufzeit verfügbar – nur die Stiftfarbe ändern.

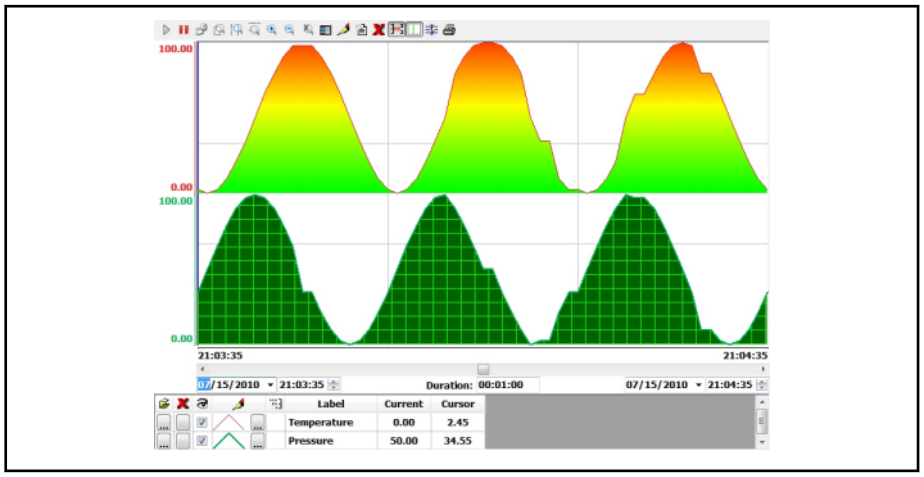


Abb.9-23: Trends mit benutzerdefinierten Füllmustern

Optionen

Zugriff auf den Dialog

Um den Dialog "Stiftstil" für einen bestimmten Datenpunkt zu öffnen, öffnen Sie zunächst den Dialog "Punkte" und klicken dann auf die Spalte **Optionen** für diesen Datenpunkt.

Alarme, Ereignisse und Trends

Der Dialog im Detail

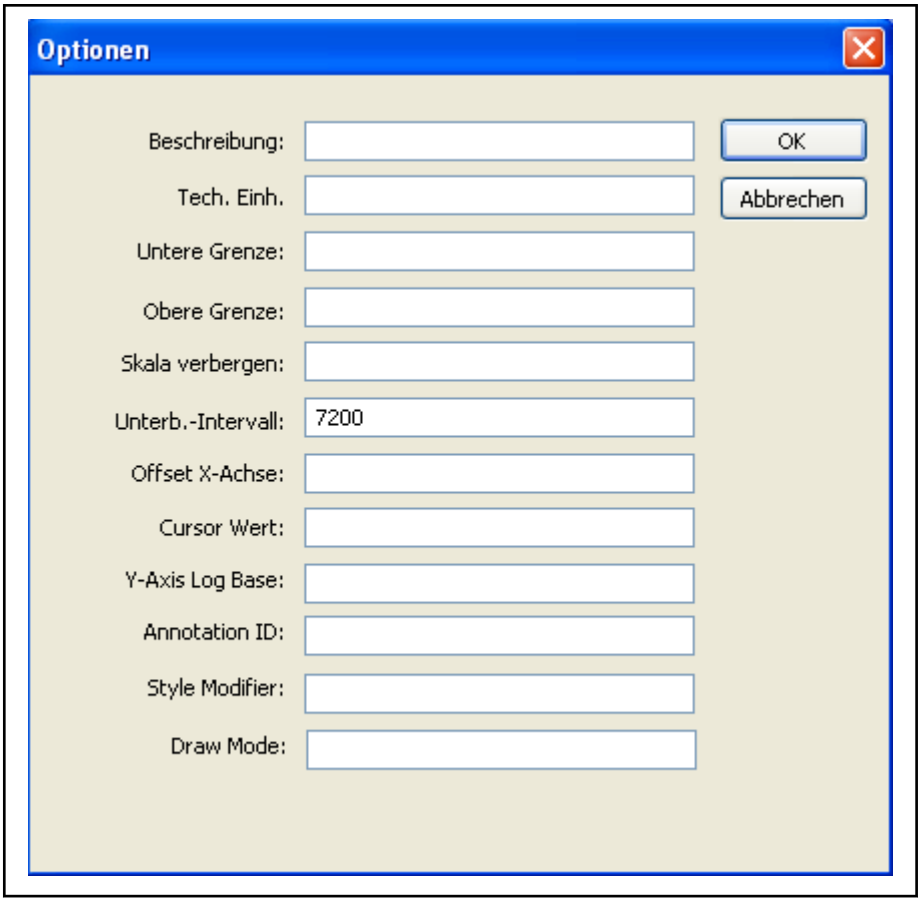


Abb.9-24: Dialog Optionen

Der Dialog "Optionen" enthält die folgenden Elemente:

Elementname	Beschreibung
Beschreibung	Dieser Text kann in der Legende angezeigt werden und liefert während der Laufzeit eine kurze Beschreibung des Datenpunkts. Wenn Tags verwendet werden, wird die für den Tag konfigurierte Beschreibung als Standard verwendet.
Maßeinheit	Dieser Text kann in der Legende angezeigt werden und gibt während der Laufzeit die mit dem Datenpunkt verbundene Einheit (d.h. die Maßeinheit) an. Wenn Tags verwendet werden, werden die für den Tag konfigurierten Einheiten als Standard verwendet.
Untere Grenze	Wenn der Datenpunktwert unterhalb dieses Grenzwerts liegt, kann dessen Stift während der Laufzeit in einem anderen Stil (z.B. Farbe) angezeigt werden. Weitere Informationen unter Dialog Stiftstil auf Seite 314. Bei der Verwendung von Tags ist der untere Standardgrenzwert der für diesen Tag konfigurierte niedrige Alarmwert.
Obere Grenze	Wenn der Datenpunktwert oberhalb dieses Grenzwerts liegt, kann dessen Stift während der Laufzeit in einem anderen Stil (z.B. Farbe) angezeigt werden. Weitere Informationen unter Dialog Stiftstil auf Seite 314. Bei der Verwendung von Tags ist der obere Standardgrenzwert der für diesen Tag konfigurierte hohe Alarmwert.
Skala verbergen	In diesem Feld kann ein Tag zur Steuerung der Sichtbarkeit der mit diesem Stift verbundenen Skala (y-Achse) während der Laufzeit konfiguriert werden, und zwar durch Wertänderung dieses Tags (0=Anzeigen, 1=Verbergen).

Elementname	Beschreibung
Unterbrechungs-intervall	Maximales Intervall zwischen zwei aufeinanderfolgenden Punkten. Wenn die Zeit zwischen zwei aufeinanderfolgenden Beispiel werden größer als diese Zahl ist (in Sekunden), nimmt das Trendobjekt an, dass in diesem Zeitraum keine Datenerfassung stattgefunden hat und zeichnet keine Verbindungslinie zwischen den beiden Beispielwerten. Wenn die x-Achse als numerisch konfiguriert ist, stellt der Wert in diesem Feld einen numerischen Skalenwert dar. Wenn die x-Achse als Datum/Uhrzeit konfiguriert ist, wird der Wert für dieses Feld in Sekunden ausgegeben. Dieses Feld hat einige spezielle Werte: • -1: Punkte nicht verbinden. • -2: Nur Punkte in aufsteigende Reihenfolge verbinden.
Offset X-Achse	Versatz für diesen Datenpunkt von der für das Objekt konfigurierten Skala der x-Achse. Diese Option ist hilfreich, wenn Daten von zwei oder mehr Datenpunkten angezeigt werden sollen, die jeweils verschiedene x-Skalen verwenden (Zeitraum/Wert). So können diese verglichen werden. Wenn die x-Achse als numerisch konfiguriert ist, stellt der Wert in diesem Feld einen numerischen Skalenwert dar. Wenn die x-Achse als Datum/Uhrzeit konfiguriert ist, wird der Wert für dieses Feld in Sekunden ausgegeben.
Cursor-Wert	Geben Sie den Namen eines Projekt-Tags ein. Während der Laufzeit aktualisiert das Trendobjekt den Wert dieses Tags mit dem Wert des Schnittpunkts zwischen dem Datenpunktstift und den vertikalen Cursor (falls vorhanden).
Log Basis y-Achse	Geben Sie einen Tag-Namen oder numerischen Wert ein. Bei einem Wert von 1 wird die y-Achse des Trends auf einer logarithmischen (z.B. 1, 10, 100, 1000) anstelle einer linearen Skala angezeigt.
Annotation ID	entf.
Style Modifier	entf.
Modus Zeichnen	Geben Sie einen Tag-Namen oder numerischen Wert ein. Bei einem Wert von 1 werden die historischen Daten für diesen Trend dezimiert, bevor der Trend gezeichnet wird. Das bedeutet, dass die x-Achse des Trends in ein bestimmte Anzahl von Intervallen unterteilt wird (bestimmt durch **Max Punkte** in den Erweiterten Einstellungen auf Seite 332) und es wird aus allen Datenpunkten innerhalb jedes Intervalls der Durchschnitt gebildet und als einzelner Punkt gezeichnet. Dies ähnelt der Dezimierungsfunktion in den Erweiterten Einstellungen, außer dass die Dezimierung nur für diesen Trend ausgeführt wird und nicht für alle Trends in der Trendanzeige.

SPC

Im Dialog "SPC" können Projekt-Tags festgelegt werden, die bestimmte, aus der Gesamthistorie der Datenpunkte errechnete statistische Werte erhalten. Diese statistischen Werte werden in der statistischen Prozesssteuerung (SPC) verwendet, einer Methode zur Überwachung von Prozessen und der Sicherstellung ihrer effizienten Ausführung.

Zugriff auf den Dialog

Zur Verwendung des Dialogs "SPC" für einen bestimmten Datenpunkt öffnen Sie zunächst auf den Dialog Punkte für das Bildobjekt Trendsteuerung auf Seite 314 und klicken dann auf die Spalte **SPC** für diesen Datenpunkt.

Alarme, Ereignisse und Trends

Der Dialog im Detail

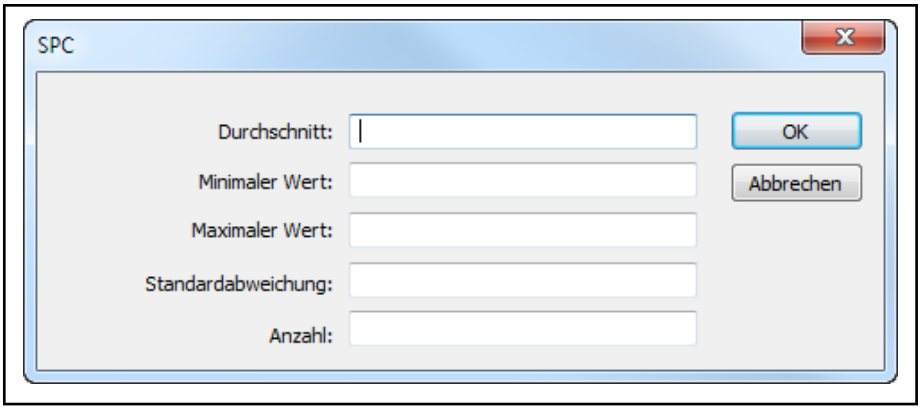


Abb.9-25: Dialog: Eigenschaften Trendobjekt

Der Dialog "SPC" enthält die folgenden Elemente:

Elementname	Beschreibung
Durchschnitt	Projekt-Tag (Typ Real), der den rechnerischen Durchschnitt aller historischen Werte des Datenpunkts erhält. Wenn ein Wert keine Zahl ist (NaN) – zum Beispiel wenn ein Tag als schlechte Qualität markiert ist – zählt er für die Berechnung des Durchschnitts als 0.
Minimaler Wert	Projekt-Tag (Typ Real), der den historischen Minimalwert des Datenpunkts erhält.
Maximaler Wert	Projekt-Tag (Typ Real), der den historischen Maximalwert des Datenpunkts erhält.
Standardabweichung	Projekt-Tag (Typ Real), der die Standardabweichung des Datenpunkts erhält. Eine geringe Standardabweichung zeigt an, dass der Wert des Datenpunkts sich nah am Durchschnitt bewegt; eine hohe Standardabweichung zeigt an, dass der Wert stark vom Durchschnitt abweichend schwankt.
Anzahl	Projekt-Tag (Typ Integer oder Real), der Gesamtzahl aller historischen Werte oder Beispielwerte des Datenpunkts erhält. Die Anzahl erhöht sich beim Ausführen des Projekts und die historische Datenbank wächst.

Achsen

Zugriff auf den Dialog

Um auf den Dialog "Achsen" für ein bestimmtes Trendobjekt zuzugreifen, wählen Sie zunächst den Dialogobjekteigenschaften für dieses Bildobjekt auf Seite 313 aus und klicken dann auf **Achsen**.

Der Dialog im Detail

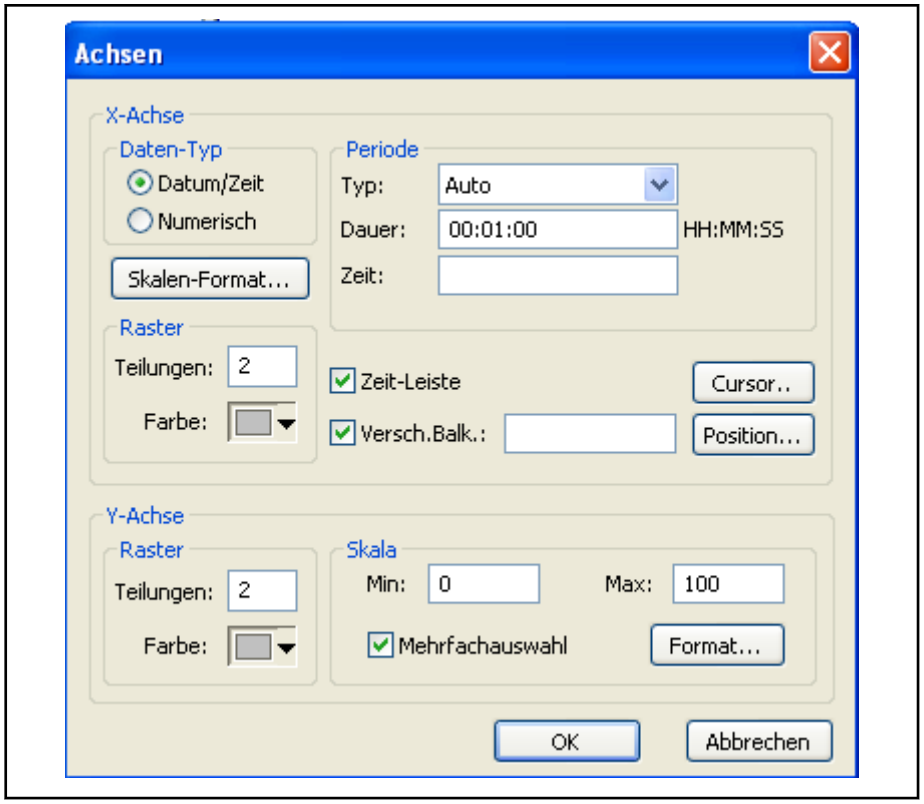


Abb.9-26: Dialog Achsen

Der Dialog "Achsen" enthält die folgenden Elemente:

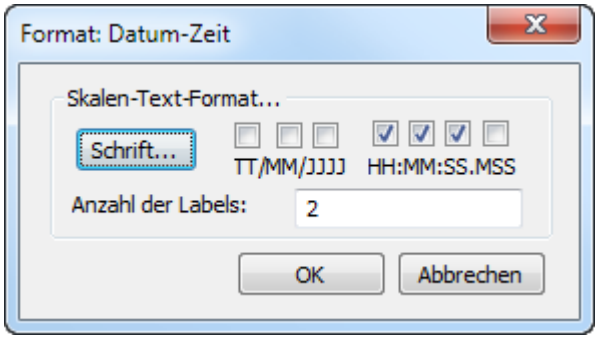
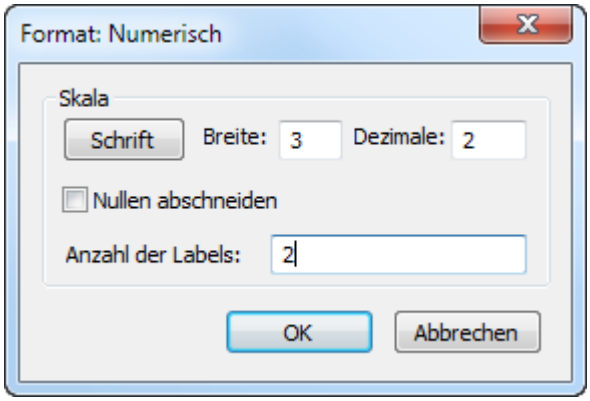
Alarme, Ereignisse und Trends

Bereich / Elementname			Beschreibung
x-Achse (1)	Datentyp	Datum/ Uhrzeit	
		Numerisch	
		Skalenformat	
	Intervall (wenn Datentyp auf Datum/ Uhrzeit gesetzt ist)	Typ	• **Auto**: Wenn diese Option gewählt ist, arbeitet das Trendobjekt mit **Start Datum/Uhrzeit**, wenn es in den Modus Pause ausgelöst wird und arbeitet mit **Zeit bisher**, wenn es in den Modus Ausführen ausgelöst wird. • **Start Datum/Uhrzeit**: Wenn diese Option ausgewählt ist, definiert der Wert des im Feld **Zeit** konfigurierten Tags Startdatum/ -uhrzeit für die auf dem Objekt angezeigten Daten. • **Zeit bisher**: Wenn diese Option ausgewählt ist, definiert der Wert des im Feld **Zeit** konfigurierten Tags die Zeitdauer vor dem aktuellen Datum/ Uhrzeit, die als Startdatum/ -uhrzeit für die auf dem Objekt angezeigten Daten verwendet wird.
		Dauer	Definiert die Dauer für die Anzeige von Daten auf dem Objekt. In diesem Feld kann ein String-Tag konfiguriert werden, damit die Dauer während der Laufzeit durch Ändern dieses Tags dynamisch angepasst werden kann. Das Format des von dieser Eigenschaft unterstützten Werts ist HH:MM:SS. Zum Beispiel "36:00:00" (sechsunddreißig Stunden).
		Zeit	Dieses Feld ist optional. Der Wert des in diesem Feld konfigurierten Tags stellt einen Zeitraum dar und kein spezielles Datum oder Uhrzeit. Die Bedeutung dieses Werts ist abhängig von der für die Eigenschaft **Typ** festgelegten Option. Wenn das Feld **Zeit** leer bleibt (oder wenn der in diesem Feld konfigurierte Tag den Wert "0" hat), zeigt das Objekt Daten bis zum aktuellen Datum/ Uhrzeit an. • Wenn der **Typ** auf Startdatum/ -uhrzeit eingestellt ist, muss der in diesem Feld konfigurierte Tag dem Format Datum/Uhrzeit entsprechen. Zum Beispiel "02/10/2005 18:30:00". • Wenn der **Typ** auf Zeit bisher eingestellt ist, muss der in diesem Feld konfigurierte Tag einem der folgenden Formate entsprechen. **Zeit (String-Wert)** z. B. "48:00:00" (achtundvierzig Stunden). **Anzahl der Stunden (Real-Wert)** z. B. "2,5" (zwei Stunden und dreißig Minuten). Wenn das Feld **Zeit** leer bleibt (oder wenn der in diesem Feld konfigurierte Tag den Wert 0 hat), zeigt das Objekt Daten bis zum aktuellen Datum/Uhrzeit an.
	Intervall (wenn Datentyp auf Numerisch gesetzt ist)	Min / Max	Auf der x-Achse angezeigte Minimal- und Maximalwerte. Die Eigenschaften **Min** und **Max** können numerische Real-Werte mit bis zu sechs Dezimalstellen enthalten. Wenn eine größere Genauigkeit erforderlich ist, müssen Sie die Eigenschaften **Min** und **Max** mit Real-Tags konfigurieren und dann die Werte in diesen Tags speichern.
		Einheiten	Technische Einheiten (z.B. kg, psi), die während der Laufzeit mit der x-Achse verbunden sind.

Bereich / Elementname			Beschreibung
x-Achse (2)	Raster	Teilungen	Die auf der x- bzw. y-Achse eingetragene Anzahl der Unterteilungen (vertikalen oder horizontalen Linien) sowie die Farbe dieser Linien können konfiguriert werden.
		Farbe	
		Zeitleist	Bei Aktivierung wird die Zeitleiste während der Laufzeit unterhalb der x-Achse angezeigt, andernfalls bleibt er verborgen. Der Zeitleiste ist eine Standardschnittstelle, die vom Benutzer während der Laufzeit zum Ändern der Skala der x-Achse verwendet werden kann.
		Bildlaufleiste	Bei Aktivierung wird die Bildlaufleiste während der Laufzeit unterhalb der x-Achse angezeigt, andernfalls bleibt sie verborgen. Die Bildlaufleiste ist eine Standardschnittstelle, die vom Benutzer während der Laufzeit zum Navigieren der Skale der x-Achse verwendet werden kann. Optional kann im Feld Bildlaufleiste ein Tag konfiguriert werden, der die Zeitdauer für die Bildlaufleiste festgelegt. Bleibt das Feld leer, ist der Zeitraum gleich dem aktuellen Wert der Dauer für die x-Achse.
		Cursor	Der Cursor ist ein optionales Lineal orthogonal zur x-Achse, das während der Laufzeit verwendet werden kann, um an einem bestimmten Punkt den Wert eines beliebigen Stifts zu erhalten (Schnittstelle zwischen Stift und Cursor). Durch Klicken auf diese Schaltfläche wird der Dialog Cursor geöffnet, in dem die Einstellungen für den optionalen vertikalen Cursor wie folgt konfiguriert werden können:
		Position	Definiert die Position auf der x-Achse sowie Richtung und Lage, wie folgt:
Y-Achse	Raster	Teilungen	Die auf der x- bzw. y-Achse eingetragene Anzahl der Unterteilungen (vertikalen oder horizontalen Linien) sowie die Farbe dieser Linien können konfiguriert werden.
		Farbe	
	Skala	Min / Max	Auf der y-Achse angezeigte Standardminimal- und Maximalwerte. Wird verwendet, wenn mehr als ein Stift die gleiche Skala verwendet (Mehrfachabschnitte deaktiviert), und/oder für die Punkte, deren Felder Min und Max nicht konfiguriert sind (leer bleiben).
		Mehrfachabschnitte	Bei Aktivierung wird die y-Skala automatisch in einen Abschnitt pro Stift aufgeteilt; andernfalls teilen sich alle Stifte die gleiche y-Skale.
		Format	Öffnet einen Dialog zu Konfigurieren des Formats der von der y-Achse angezeigten Beschriftungen.

☞ Die in den Feldern **Dauer/Bereich** konfigurierten Tags werden automatisch aktualisiert, wenn der Benutzer während der Laufzeit die x-Skala dynamisch verändert unter Verwendung der im Trendobjekt eingebetteten Zeitleiste.

- **Datentyp**: Die X-Achse kann gemäß dieser Einstellung entweder Datums-/Uhrzeitwerte oder numerische Werte anzeigen.

Datentyp	Format Skale
Datum/Uhrzeit	
Numerisch	

☞ Die Anzahl der Dezimalstellen für die x- oder y-Skala (Dezimalzahlen) kann mit einem Tag konfiguriert werden. Daher kann diese Einstellung während der Laufzeit dynamisch verändert werden.

- **Cursor**: Der Cursor ist ein optionales Lineal orthogonal zur x-Achse, das während der Laufzeit verwendet werden kann, um an einem bestimmten Punkt den Wert eines beliebigen Stifts zu erhalten (Schnittstelle zwischen Stift und Cursor). Durch Klicken auf diese Schaltfläche wird der Dialog Cursor geöffnet, in dem die Einstellungen für den optionalen vertikalen Cursor wie folgt konfiguriert werden können:

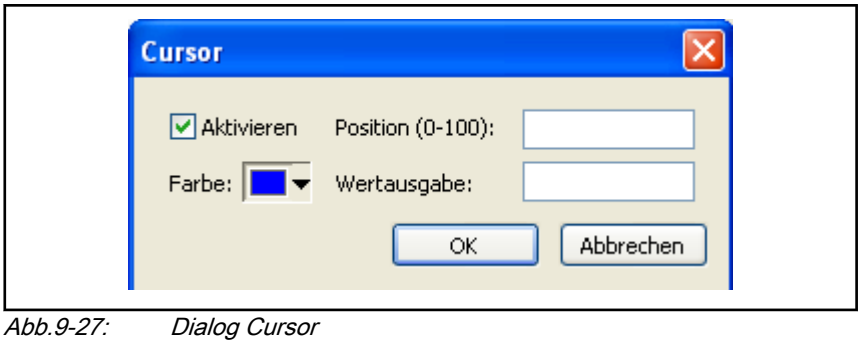


Abb.9-27: Dialog Cursor

Eigenschaft	Beschreibung
Aktivieren	Bei Aktivierung ist der vertikale Cursor während der Laufzeit sichtbar.
Farbe	Farbe der für den Cursor gezeichneten Linie.

Eigenschaft	Beschreibung
Position (0-100)	In diesem Feld kann ein numerischer Tag konfiguriert werden, der proportional zur Cursorposition auf der X-Achse ist, von 0-100 %. Wenn dieser Wert verändert wird, wird die Position des Cursors automatisch verändert.
Wertausgabe	In diesem Feld kann ein String-Tag konfiguriert werden, der den Wert der x-Achse zurück gibt, auf der der Cursor aktuell steht.

- **Position**: Definiert die Position auf der x-Achse sowie Richtung und Lage, wie folgt:

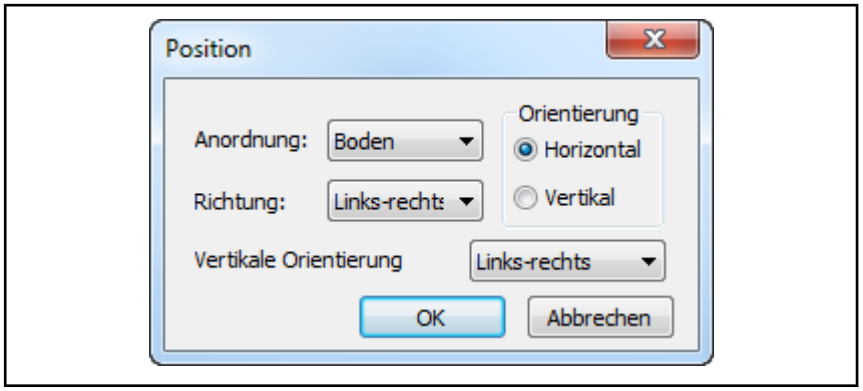


Abb.9-28: Dialog Position

Eigenschaft	Beschreibung
Anordnung	Seite der Trendanzeige, auf der die x-Achse platziert wird.
Richtung	Richtung der X-Achse.
Orientierung	Lage der X-Achse.
Vertikale Orientierung	Die Lage der Beschriftung auf der vertikalen Achse, egal ob die vertikale Achse x oder y ist.

Werkzeugleiste

Der Dialog "Werkzeugleiste" wird zum Anpassen der Werkzeugleiste in einem Trendobjekt verwendet.

Zugriff auf den Dialog

Um auf den Dialog "Werkzeugleiste" für ein bestimmtes Trendobjekt zuzugreifen, wählen Sie zunächst den Dialogobjekteigenschaften für dieses Bildobjekt auf Seite 313 aus und klicken dann auf **Werkzeugleiste**.

Alarme, Ereignisse und Trends

Der Dialog im Detail

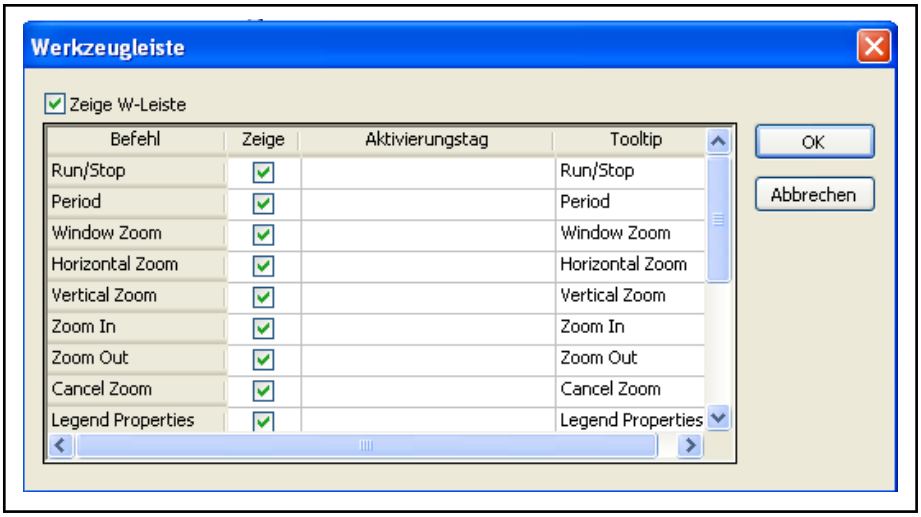


Abb.9-29: Dialog Werkzeugleiste

Die Option **Werkzeugleiste anzeigen** steuert, ob die gesamte Werkzeugleiste während der Laufzeit angezeigt wird. Die Werkzeugleiste kann verborgen werden, um Platz zu sparen oder zu verhindern, dass Benutzer die Trendanzeige verändern.

Außerdem haben alle Befehle/Werkzeuge auf der Werkzeugleiste die folgenden Eigenschaften:

Spaltenname	Beschreibung
Befehl	Name des Befehls/Werkzeugs. Weitere Informationen zu jedem Werkzeug unter
Anzeigen	Option zum Anzeigen des Werkzeugs auf der Werkzeugleiste
Aktivierungs-Tag	Ein optionaler Tag-Auslöser – wenn sich der Tag-Wert von FALSCH (0) auf WAHR (alle Nicht-Null-Werte) ändert, wird der Befehl aktiviert, als habe der Benutzer auf das Werkzeug geklickt. Dieser kann während der Laufzeit zum Schreiben von Änderungen in der Trendanzeige verwendet werden.
Tooltip	Der angezeigte Tooltip, wenn der Mauszeiger über dem Werkzeug stehen bleibt.

Datenquellen

Zugriff auf den Dialog

Um auf den Dialog "Datenquellen" für ein bestimmtes Trendobjekt zuzugreifen, wählen Sie zunächst den Dialog Objekteigenschaften für dieses Bildobjekt auf Seite 313 aus und klicken dann auf **Datenquellen**.

Der Dialog im Detail



Abb.9-30: Dialog Objekteigenschaften: Trendobjekt

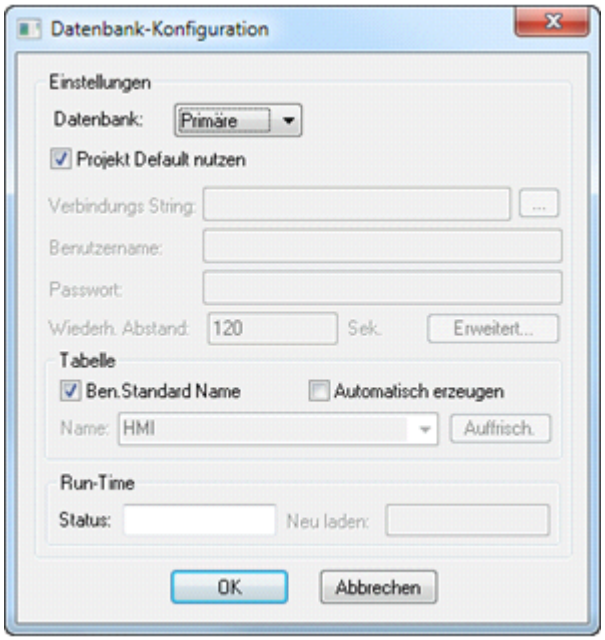
Die **Datenquelle** bestimmt den Ort der Werte der damit verbundenen Datenpunkte. Mehrere Punkte können die gleiche Datenquelle haben – es muss nicht für jeden Datenpunkt eine Datenquelle erstellt werden.

Der Datenquellen-Tag steht dem Trendobjekt standardmäßig zur Verfügung. Zusätzliche Datenquellen können mit der Schaltfläche **Neu** hinzugefügt werden. Der angegebene Name wird als Alternativname zum Verknüpfen der Datenpunkte mit dieser neuen Datenquelle verwendet.

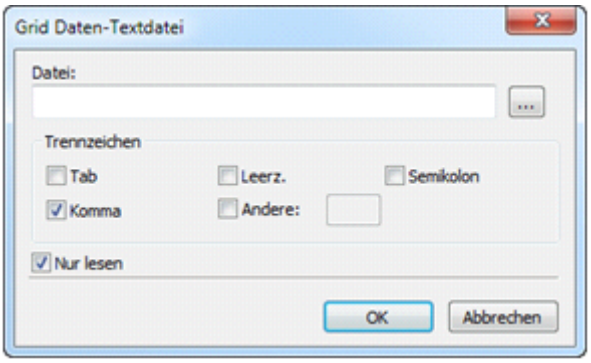
Mit den anderen Feldern in diesem Dialog können die Datenquelleneinstellungen bearbeitet werden:

- **Quellentyp**: Auswahl des Quelltyps für den Ort der Datenpunktwerte.
- **x-Achse**: Geben Sie den Namen des Felds (der Spalte) aus der Datenquelle ein, das die Daten für die x-Achse enthält.
- **Max. Puffer**: Die maximale Datenmenge (in Bytes), die im Laufzeitspeicher vorgehalten wird.
- **Ladefortschritt**: Der Tag in diesem Feld erhält einen Real-Wert (0-100), der den prozentualen Fortschritt des Ladens der Datenquelle darstellt.
- **Sortieren**: Diese Option ist hilfreich für die punktweise Aufzeichnung von Daten aus einer Textdatei. Wenn diese Funktion aktiviert (ausgewählt) ist, werden die Daten sortiert und der Cursorspaltenwert angezeigt, bis der **Max. Puffer** voll ist. Wenn diese Funktion deaktiviert (abgewählt) ist, werden die Daten nicht sortiert und der Cursorspaltenwert wird nicht angezeigt.
- **Geöffnet halten**: Diese Option erhält die Datenquelle so lange offen, wie das Bild mit der Trendanzeige geöffnet ist. Dies verbessert die Leistung des Laufzeitprojekts. Offenhalten der Datenquelle kann allerdings andere Probleme verursachen, zum Beispiel Datenbankverbindungsfehler (wenn für **Quellentyp Datenbank** gewählt ist) und Dateischreibkonflikte (wenn für **Quellentyp Stapel** oder **Textdatei** gewählt ist). Diese Option zum Schließen der Datenquelle nachdem Laden der Daten löschen (abwählen).
- **Einstellungen Datenquelle**: Anklicken zum Definieren der Einstellungen für den ausgewählten **Quellentyp**.

Die folgende Tabelle zeigt eine Zusammenfassung der Einstellungen für jeden **Quellentyp**:

Daten-quellentyp	Beschreibung	Feld x-Achse	Einstellungen Datenquelle
Stapel	Vom Task **Trend** von "WinStudio" erstellter Stapel	Deaktiviert. Die Daten der x-Achse werden automatisch an der richtigen Position von der durch "WinStudio" erzeugten, proprietären Stapeldatei abgerufen.	 Geben Sie die Datenpunktwerte in **Stapelname** für deren Abruf ein. In diesem Feld kann ein Tag in geschweiften Klammern konfiguriert werden, um diese Einstellung während der Laufzeit dynamisch zu ändern.
Datenbank	Relationale SQL-Datenbank	Feldname, der die Daten der x-Achse enthält	 Konfigurieren Sie die Einstellungen für die Verknüpfung dieser Datenquelle mit der relationalen SQL-Datenbank, die die Datenpunktwerte enthält. Weitere Informationen über diese Dialogschnittstelle unter Dialogfenster Datenbankkonfiguration auf Seite 502. Ein Beispiel für die Konfiguration von Datenbanken finden Sie in Anhang B auf Seite 339.

Alarme, Ereignisse und Trends

Daten-quellentyp	Beschreibung	Feld x-Achse	Einstellungen Datenquelle
Textdatei	Textdatei (z.B. CSV-Datei) mit Datenpunktwerten, getrennt durch einen bestimmten Feldbegrenzer	Nummer der Spalte, die die Daten der x-Achse enthält. Die Anzahl 0 bezieht sich auf die erste Spalte, 1 bezieht sich auf die zweite Spalte usw.	 Geben Sie den Namen der Textdatei ein, die die Datenpunkte enthält. Der Standardpfad ist der aktuelle Projektordner. In diesem Feld kann zum dynamischen Ändern dieser Einstellung während der Laufzeit ein Tag in geschweiften Klammern konfiguriert werden. Außerdem können ein oder mehrere Feldbegrenzer für die in der Textdatei gespeicherten Daten ausgewählt werden. Der Wert jeder Zeile wird zwischen zwei Feldbegrenzer in die Textdatei geschrieben. Bei Verwendung von Kommas als Feldbegrenzer kann das Rasterobjekt Daten aus CSV-Dateien auslesen. Durch Auswählen der Option **Andere** kann auch ein benutzerdefinierter Feldbegrenzer ausgewählt werden. Ein Beispiel für die Konfiguration von Textdateien finden Sie in Anhang A auf Seite 335.

Legende

Zugriff auf den Dialog Um auf den Dialog "Legende" für ein bestimmtes Trendobjekt zuzugreifen, wählen Sie zunächst den Dialog Objekteigenschaften für dieses Bildobjekt auf Seite 313 aus und klicken dann auf **Legende**.

Der Dialog im Detail

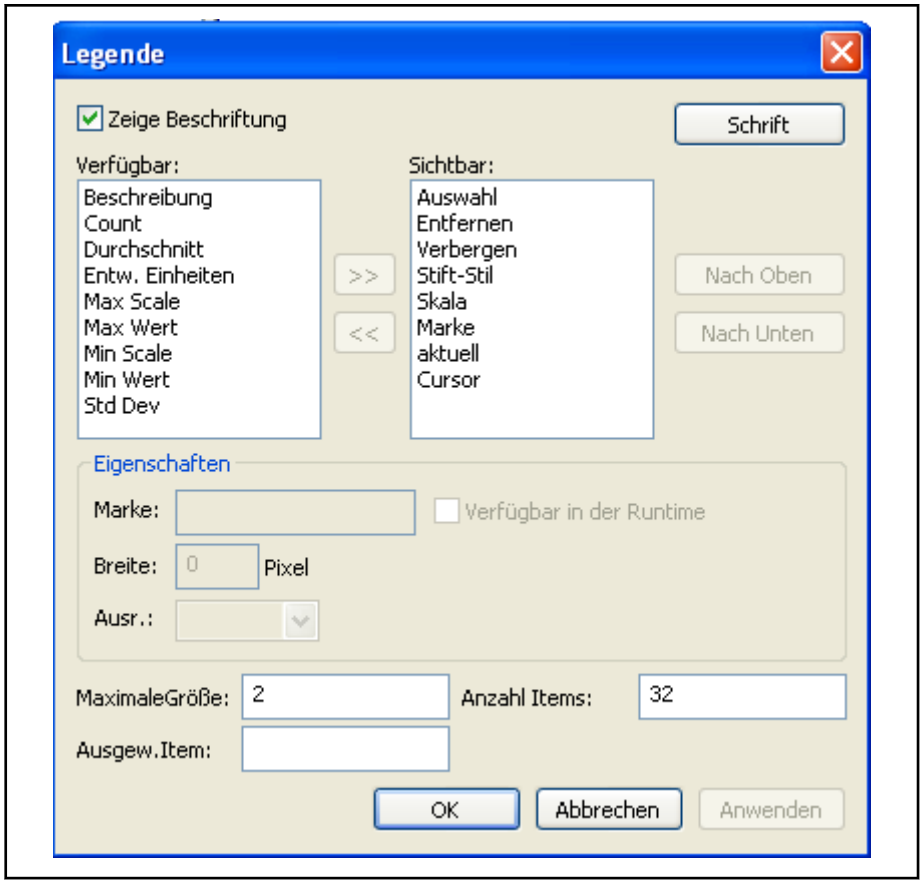


Abb.9-31: Dialog Legende

Der Dialog "Legende" enthält die folgenden Elemente:

- **Legende anzeigen**: Bei Aktivierung wird die eingebettete Legende während der Laufzeit angezeigt. Diese Schnittstelle liefert nützliche Informationen im Zusammenhang mit den aktuell verwendeten Stiften.
- **Verfügbar/sichtbar**: Die Elemente im Kasten **Sichtbar** werden während der Laufzeit in der Legende angezeigt. Durch Verwendung der Schaltflächen **»** und **«** können dem Liste **Sichtbar** Elemente hinzugefügt oder daraus entfernt werden. Außerdem können die Schaltflächen **Nach oben** und **Nach unten** zum Ändern der Anzeigereihenfolge der während der Laufzeit in der Legende angezeigten Elemente verwendet werden. Die folgende Tabelle führt die verfügbaren Elemente der Legende auf:

Element	Legenden-symbol	Beschreibung
Einheiten		Die Einheiten von Tag/Stift
Min. Wert		Der kleinstmögliche Wert von Tag/Stift
Max. Wert		Der größtmögliche Wert von Tag/Stift
Auswahl		Durch Klicken auf diese Schaltfläche wird ein anderer Tag für diesen Stift ausgewählt
Entfernen		Durch Klicken auf diese Schaltfläche wird dieser Stift vollständig aus der Legende und der Trendgrafik entfernt
Verbergen		Diese Option auswählen (aktivieren), um diesen Stift in der Trendgrafik zu verbergen

Alarme, Ereignisse und Trends

Element	Legenden-symbol	Beschreibung
Stiftstil		Durch Klicken auf diese Schaltfläche werden Linienart, Stärke, Farbe, Markierungen usw. des Stifts verändert
Skala		Diese Option auswählen (aktivieren), um die Skala dieses Stifts in der Trendgrafik anzuzeigen
Beschreibung		Beschreibung des Tags/Stifts
Aktuell		Der aktuelle Wert des für den Stift konfigurierten Tags
Cursor		Der Wert des Stifts an der Schnittstelle mit der Cursorlinie

- **Eigenschaften**: Ermöglicht das Konfigurieren der Eigenschaften für das in der Liste **Verfügbar** oder **Sichtbar** markierte Feld:

Eigenschaft	Beschreibung
Beschriftung	Beschriftung für das während der Laufzeit angezeigte Feld
Breite	Breite für das Feld (im Pixel) während der Laufzeit
Ausrichtung	Ausrichtung der in dem Feld angezeigten Daten
Verfügbar während Laufzeit	Wenn diese Option aktiviert ist, kann der Benutzer während der Laufzeit das Feld anzeigen oder verbergen

- **Maximale Größe**: Bestimmt die Größe der Legende als Anzahl von Zeilen. Wenn zum Beispiel der Benutzer acht Punkte im Trendobjekt anzeigen lässt und die Maximalgröße auf zwei eingestellt ist, erhält die Legende eine Leiste, mit der der Benutzer zu den anderen Punkten blättern kann.
- **Anzahl der Elemente**: Anzahl der auf der Legende angezeigten Punkte (Standard). Dem Benutzer kann gestattet werden, während der Laufzeit Punkte hinzuzufügen bzw. zu entfernen, ungeachtet des Werts in diesem Feld.
- **Ausgewähltes Element**: In diesem Feld kann ein numerischer Tag konfiguriert werden. Das Objekt schreibt die Nummer der ausgewählten Zeile in diesem Tag. Zusätzlich können verschiedene Zeilen durch Schreiben ihrer Werte in diesen Tag ausgewählt werden.
- **Schriftarten**: Legt die Schriftart für den in der Legende angezeigten Text fest.

Weitere Einstellungen

Zugriff auf den Dialog

Um auf den Dialog "Weitere Einstellungen" für ein bestimmtes Trendobjekt zuzugreifen, wählen Sie zunächst den Dialogobjekteigenschaften für dieses Bildobjekt auf Seite 313 aus und klicken dann auf **Weitere Einstellungen**.

Der Dialog im Detail

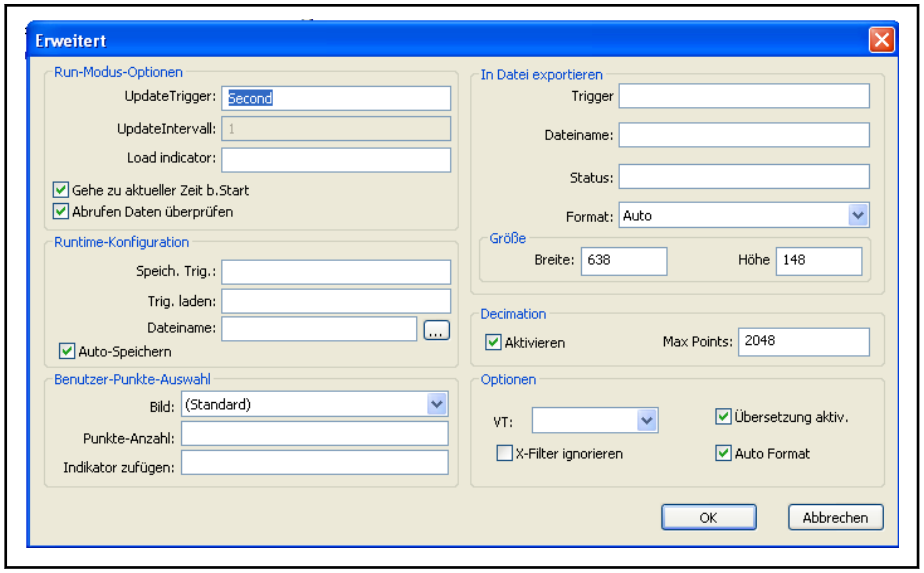


Abb.9-32: Dialog Achsen

Der Dialog "Weitere Einstellungen" enthält die folgenden Elemente:

Bereich / Elementname		Beschreibung
Optionen Modus Ausführen	**Aktualisierungstrigger**	Wenn der in diesem Feld konfigurierte Tag seinen Wert ändert, wird das Trendobjekt aktualisiert.
	Update Intervall	Wenn der Aktualisierungstrigger ausgelöst wird und die x-Achse vom Typ Numerisch ist, wird der Wert in diesem Feld zu den Minimal- und Maximalwerten der x-Achse hinzu addiert.
	Ladestatus	Geben Sie den Namen eines Projekt-Tags ein. Während das Trendobjekt externe Daten lädt, erhält der Tag den Wert 1. Wenn das Trendobjekt das Laden der Daten beendet hat, erhält der Tag den Wert 0.
	Bei Ausführung zu aktueller Zeit bewegen	Wenn dieses Kästchen aktiviert ist, verschiebt sich die x-Achse automatisch zu aktuellen Zeit, wenn das Objekt während der Laufzeit in den Modus Ausführen gewechselt wird.
	Ausgehende Werte abrufen	Wenn dieses Kästchen aktiviert ist, ruft das Objekt die ausgehenden Daten des Objekts ab (nur erste Punkte). Deaktivieren dieser Option kann die Leistung verbessern, da die ausgehenden Punkte des Objekts nicht aus der Historie abgerufen werden. Andererseits zieht das Objekt dann keine Linien zum Verbinden der ersten und letzten Beispielwerte mit den Höchstgrenzen des Objekts.

Alarme, Ereignisse und Trends

Bereich / Elementname		Beschreibung
Runtime-Konfiguration	**Trigger für Speichern**	Die während der Laufzeit veränderten Einstellungen des Trendobjekts können in temporären Dateien gespeichert werden. Diese Option kann wie folgt verwendet werden: Wenn der in diesem Feld konfigurierte Tag seinen Wert ändert (z.B. umschaltet), werden die aktuellen Einstellungen des Trendobjekts in der temporären Datei gespeichert. Dieser Befehl ist für den Thin Client nicht verfügbar. • Einstellungen konsistent halten, so dass die während der Laufzeit konfigurierten Einstellungen nicht verloren gehen, wenn der Benutzer das Bild schließt und wieder öffnet oder das Projekt neu startet. • Erstellen von Standardeinstellungen für verschiedene Szenarien und Laden der entsprechenden Konfiguration während der Laufzeit, aufgrund einer festgelegten Bedingungen oder aufgrund der Benutzerauswahl.
	Trigger für Laden	Wenn der in diesem Feld konfigurierte Tag seinen Wert ändert (z.B. umschaltet), werden die Einstellungen aus der temporären Datei geladen und während der Laufzeit auf das Trendobjekt angewendet. **Hinweis:** Nach dem Speichern des Bilds, in der das Trendobjekt konfiguriert ist, werden die Einstellungen beim erneuten Öffnen des Bilds nicht automatisch aus der temporären Datei geladen, es sei denn, vor dem Schließen des Bilds wurde der Befehl **Auslöser laden** ausgeführt.
	Dateiname	Bleibt dieses Feld leer, wird die temporäre Datei im Unterordner **Web** Ihres Projekts mit der Syntax **BildNameObjektIDTrendObjekt.stmp** (z.B. **MeineBild10TrendObjekt.stmp**) gespeichert. Die Thin-Client-Station speichert/lädt die temporäre Datei in das Standardverzeichnis "Temp" des Betriebssystems (z.B. **\Dokumente und Einstellungen\AktuellerBenutzer\Lokale Einstellungen\Temp**). In diesem Feld kann ein benutzerdefinierter Dateiname für die temporäre Datei konfiguriert werden oder sogar ein String-Tag in geschweiften Klammern konfiguriert werden, so dass der Benutzer den Namen der Konfigurationsdatei während der Laufzeit durch Ändern des Werts für diesen Tag dynamisch ändern kann. Wird kein Pfad angegeben, wird die Datei standardmäßig im Unterordner **Web** Ihres Projekts gespeichert.
	Automatisch Speichern	Wenn dieses Kontrollkästchen aktiviert ist, werden die aktuellen Einstellungen für den Trend automatisch in der temporären Datei gespeichert, wenn während der Laufzeit des Bilds, in der der Trend konfiguriert ist, geschlossen wird. Wird das Kästchen nicht aktiviert, werden die Einstellungen erst gespeichert, wenn der **Trigger für Speichern** aktiviert wird.
Benutzerdefinierte Punktauswahl	**Bild**	Mit dieser Schnittstelle kann ein benutzerdefinierter Dialog zum Ändern oder Einfügen von Stiften in das Objekt erstellt werden. Name des zu öffnenden Bilds, wenn der Benutzer während der Laufzeit einen Befehl zum Verändern oder Einfügen eines neuen Stifts in das Objekt auslöst.
	Punktnummer	Punktnummer (aus dem Dialog Punkte auf Seite 314) zur Anzeige des mit dem Stift verbundenen Punkts, der während der Laufzeit eingefügt oder verändert wird.
	Indikator hinzufügen	Merker, der angibt, dass der Benutzer eine Aktion zum Einfügen eines neuen Stifts (Wert 1) ausgelöst hat, anstatt einen Stift zu verwenden, der bereits visualisiert ist (Wert 0).

Bereich / Elementname		Beschreibung
In Datei exportieren	**Trigger**	Wenn der in diesem Feld konfigurierte Tag seinen Wert ändert (z.B. umschaltet), wird der aktuelle Zustand der Trendanzeige in eine Bilddatei exportiert. Werkzeugleiste und Bildlaufleiste werden nicht eingeschlossen. Legende und Zeitanzeige werden nur eingeschlossen, wenn die Trendanzeige für deren Anzeige konfiguriert ist.
	Dateiname	Pfad und Name der exportierten Datei. Wird kein Pfad angegeben, wird die Datei im Projektverzeichnis gespeichert. Wird keine Dateierweiterung angegeben, wird diese durch **Format** bestimmt. **Tipp:** Zum Ändern des Dateinamens während der Laufzeit kann ein Projekt-Tag in geschweiften Klammern angegeben werden (z.B. **{tagname}**).
	Zustand	Der in diesem Feld konfigurierten Tag erhält Zustandscodes, die erfolgreiches oder nichterfolgreiches Exportieren angeben.
	Format	Das Grafik-Format der exportierten Datei. Wenn **Auto** gewählt ist, wird das Format durch die in **Dateiname** festgelegte Erweiterung bestimmt. Wenn **Auto** gewählt ist aber keine Erweiterung angegeben ist, wird BMP als Standardformat verwendet.
	Größe	Standardmäßig wird die Bilddatei in voller Größe exportiert. Breite und Höhe können aber festgelegt werden (in Pixel).
Dezimierung	**Aktivieren**	Wird diese Option gewählt, werden die Daten der Trends im Trendobjekt, die für die Anzeige von historischen Daten konfiguriert sind, vor dem Zeichnen der Trends dezimiert. Das bedeutet für jeden Trend, dass die x-Achse des Trends in ein bestimmte Anzahl von Intervallen unterteilt wird (bestimmt durch **Max Punkte**) und es wird aus allen Datenpunkten innerhalb jedes Intervalls der Durchschnitt gebildet und als einzelner Punkt gezeichnet. Wenn eine große Menge an historischen Daten zur Anzeige zur Verfügung steht, kann dies die Laufzeitleistung verbessern. Außerdem kann es die Trends besser lesbar machen. Dezimierung funktioniert nur, wenn das Trendobjekt im Stopp-Modus (oder Historischen Modus) ist. Bitte beachten Sie bei Auswahl dieser Optionen, dass die Dezimierung für alle Trends durchgeführt wird, die zur Anzeige historischer Daten konfiguriert wurden. Zum Ausführen für einen Einzeltrend verwenden Sie den **Modus Zeichnen** in den Optionen auf Seite 317.
	Max Punkte	Die Höchstzahl an Datenpunkten für das Zeichnen jedes Trends. Standard ist 2048.
Optionen	**VK** (Virtuelles Keyboard)	Für dieses Objekt wird der Typ Virtuelle Tastatur auf Seite 80 verwendet.
	X-Filter ignorieren	Wird diese Box aktiviert, wird der X-Filter ignoriert, damit der Datenquelle kein WO oder Abfragen hinzugefügt werden.
	Übersetzung aktivieren	Aktiviert die externe Übersetzung für den von diesem Objekt angezeigten Text.
	Auto-Format	Wenn dieses Kästchen aktiviert ist, werden Dezimalwerte in den Spalten auf Seite 330 "Aktuell", "Cursor", "Max", "Min" und "Skale" gemäß der von der Funktion SetDecimalPoints auf Seite 681 erzeugten virtuellen Tabelle formatiert. **Hinweis:** Damit **Auto Format** funktioniert, muss die Dezimalformatierung auf der x-Achse deaktiviert sein – d.h. das Feld **Dezimalzahlen** im Dialog Achsen auf Seite 320 muss leer bleiben.

9.6.4 Verwenden einer Textdatei als Datenquelle

Die Trendanzeige kann Trendgrafiken aus jeder Textdatei erzeugen, in der die Werte in Spalten und Zeilen organisiert sind. Die Spalten sollten durch Sonderzeichen (normalerweise Komma) voneinander getrennt werden. Jedes Beispiel (Wertepaar für einen Punkt im Graph) wird durch eine Zeile dar-

gestellt. Angenommen, der Benutzer möchte eine Grafik mit den Informationen aus der folgenden Tabelle anzeigen:

X-Wert	Y1-Wert	Y2-Wert
0	0	10
1	1	20
2	2	30
3	3	40

Wir haben eine Variable für die x-Achse und zwei Variablen (Y1 und Y2) für die verschiedenen Linien in der Grafik. Im ersten Schritt werden die Daten in eine Textdatei konvertiert. Wird das Komma als Trennzeichen verwendet, sieht die Datei wie unten gezeigt aus:

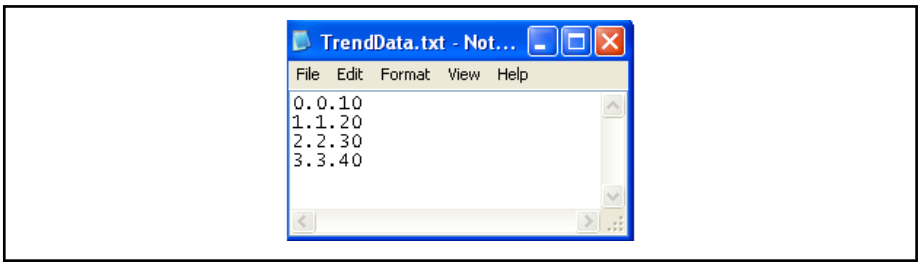


Abb.9-33: Text Trenddaten

Es wird dringend empfohlen, die Datei im gleichen Ordner wie das Projekt zu speichern. Dadurch muss nicht der gesamte Pfad angegeben werden und Ihr Projekt funktioniert auch noch, wenn es auf einen anderen Rechner kopiert wurde.

Nach dem Hinzufügen des Trendobjekts zu ihrem Bild führen Sie zum Öffnen der "Objekteigenschaften" einen Doppelklick auf das Objekt aus und klicken auf **Achse...**.

Ändern Sie den "Datentyp" der "x-Achse" auf **Numerisch** und legen Sie die Bereiche wie unten gezeigt fest:

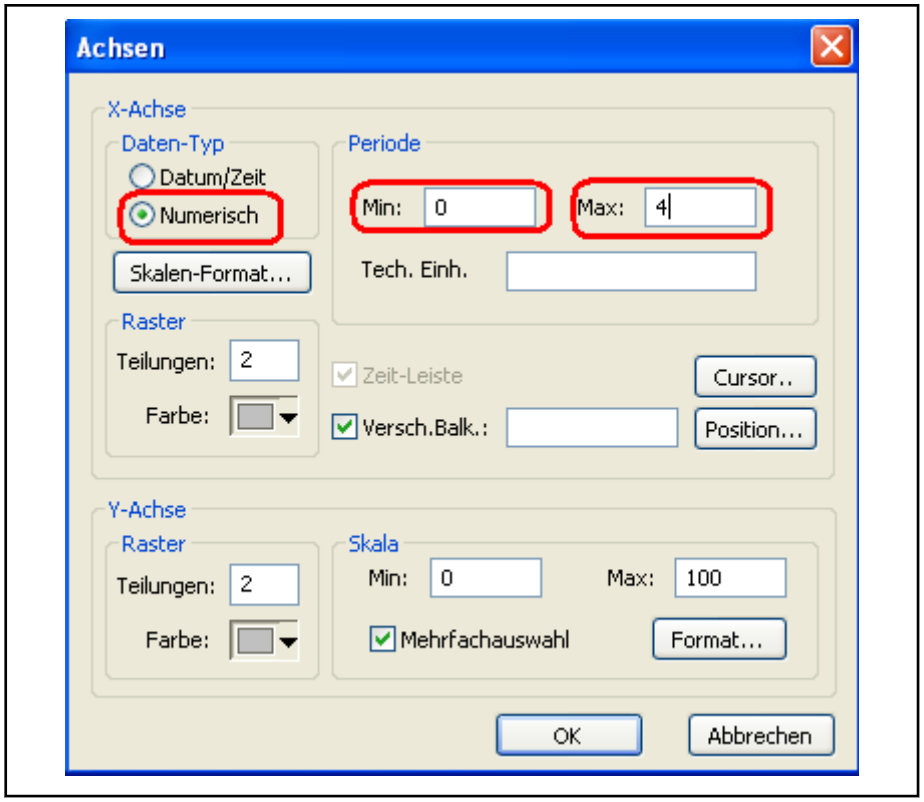


Abb.9-34: Dialog Achsen

Klicken Sie in diesem Fenster auf **OK** und klicken Sie dann im Fenster "Objekteigenschaften" auf die Schaltfläche **Datenquellen...**. Das folgende Fenster wird angezeigt:



Abb.9-35: Trendobjekt – Dialog Datenquellen

Für den Zugriff auf die Textdatei muss eine Datenquelle erstellt werden. Klicken Sie auf die Schaltfläche **Neu...**, geben Sie den **Datenquellen**-Namen **MeineTextDatei** ein und klicken Sie dann auf **Erstellen**. Jetzt sollten die folgenden Informationen zu sehen sein:



Abb.9-36: Feld x-Achse auf 0 einstellen

Im Feld **x-Achse** muss angegeben werden, welche Spalte in der Textdatei den Werten für die x-Achse entspricht. In dem Beispiel wird die Spalte Null verwendet. Geben Sie also für dieses Feld "0" ein und klicken dann auf **Einstellungen Datenquelle...**. Das folgende Fenster wird angezeigt:

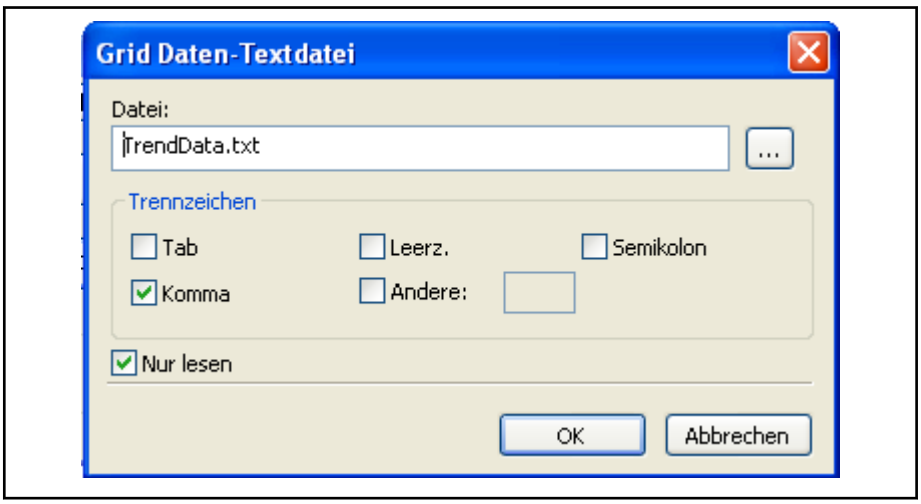


Abb.9-37: Auswahl der Textdatei

Wenn Sie die Textdatei in den Projektordner kopiert haben, müssen Sie nur den Dateinamen angeben, andernfalls geben Sie den kompletten Pfad für den Speicherort ein (bei Bedarf Schaltfläche Suchen verwenden). Klicken Sie in diesem Fenster auf **OK** und nochmals auf **OK**, um die Konfiguration der Datenquelle abzuschließen und das Konfigurationsfenster "Datenquelle" zu schließen.

Jetzt müssen wir unser Y1 und unser Y2 definieren. Diese werden durch Punkte in der Trendanzeige dargestellt. Führen Sie nochmals einen Doppelklick auf das Trendobjekt aus, um zum Fenster "Objekteigenschaften" zu gelangen, und klicken Sie dann auf **Punkte...**. Im nächsten Schritt werden die Punkte gemäß der folgenden Abbildung definiert:

Nach Befolgen dieser Schritte führen Sie Ihr Projekt aus. Das Ergebnis sollte in etwa der folgenden Abbildung entsprechen:

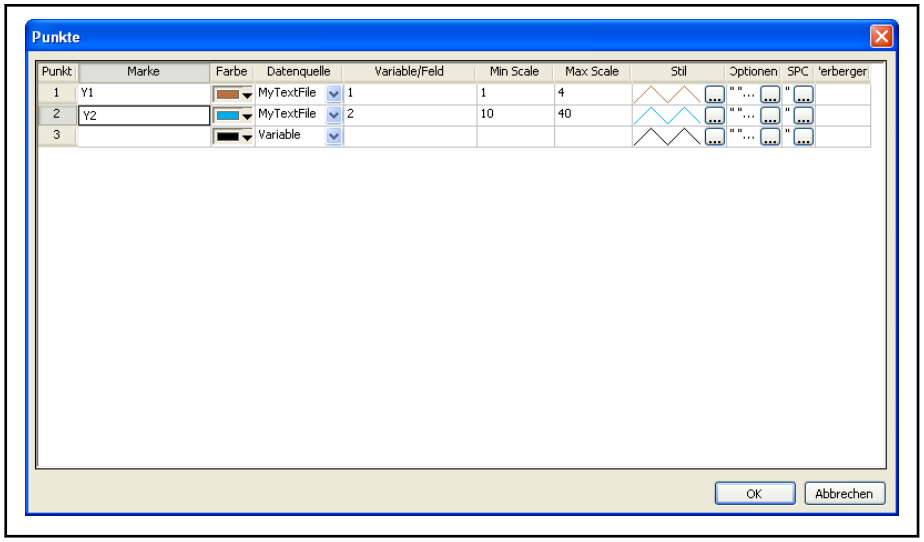


Abb.9-38: Punktetabelle

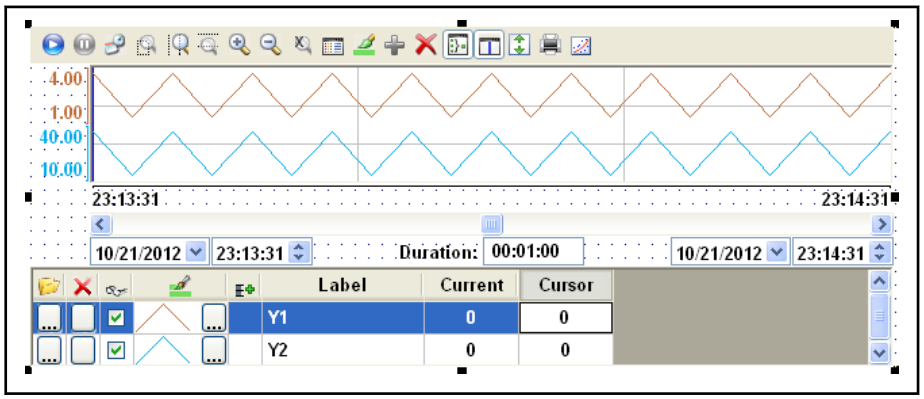


Abb.9-39: Diagramm

9.6.5 Verwenden einer Datenbank als Datenquelle

Die Trendanzeige kann Trendgrafiken aus jeder relationalen Datenbank erstellen, auf die mit der ADO.Net-Technologie zugegriffen werden kann.

Dieser Anhang erläutert den Zugang zu einer Microsoft Access Datenbank.

Wird eine andere Art von Datenbank verwendet, treffen fast alle Definitionen zu, allerdings muss die Verbindung anders konfiguriert werden. Informationen zur Konfiguration anderer Datenbanken finden Sie in den Anhängen im Abschnitt Datenbankschnittstelle dieses Handbuchs.

Angenommen, Sie haben eine Access-Datenbank auf Ihrem C-Laufwerk mit dem Namen "meinedaten.mdb" und Sie möchten eine Grafik auf Grundlage der Informationen in der folgenden Tabelle erstellen:

Alarme, Ereignisse und Trends

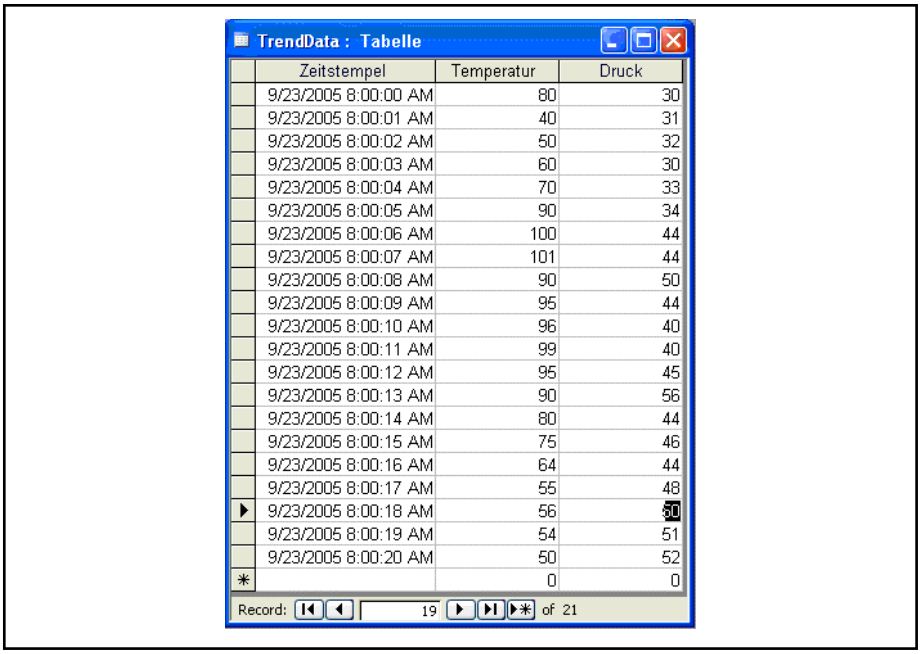


Abb.9-40: Tabelle Trenddaten

Im ersten Schritt wird das Trendobjekt zu Ihrem Bild hinzugefügt. Führen Sie jetzt einen Doppelklick auf das Objekt aus, um die Objekteigenschaften zu öffnen, und klicken Sie auf **Datenquellen...**. Das folgende Fenster wird angezeigt:



Abb.9-41: Trendobjekt – Dialog Datenquellen

Für den Zugriff auf die Datenbank muss eine Datenquelle erstellt werden. Klicken Sie auf die Schaltfläche **Neu...**, geben Sie den **Datenquellen**-Namen **MeineDB** ein und klicken Sie dann auf **Erstellen**. Jetzt sollten die folgenden Informationen zu sehen sein:

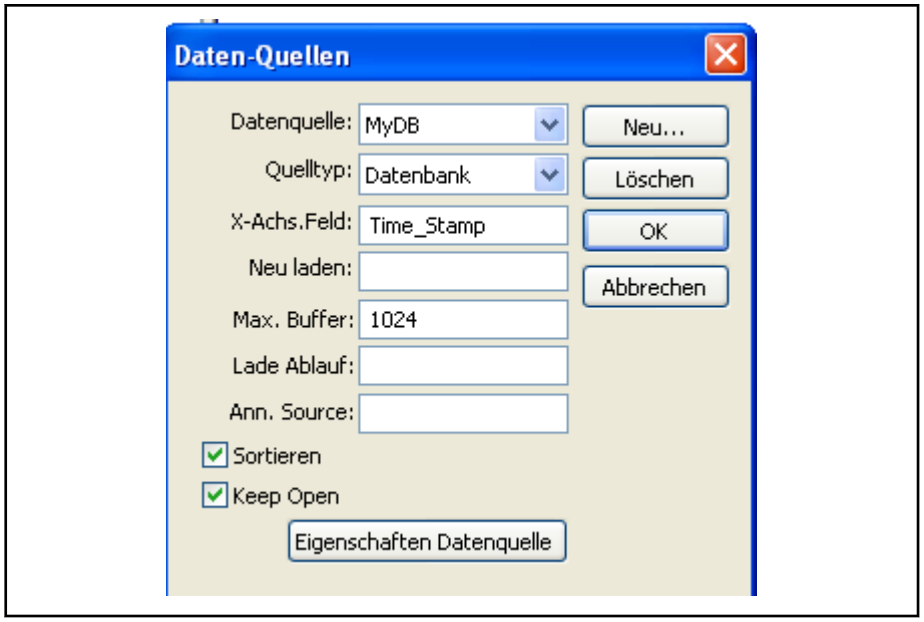


Abb.9-42: Feld x-Achse auf Time_Stamp einstellen

Den **Quellentyp** auf **Datenbank** einstellen und im Feld **x-Achse** "Time_Stamp" festlegen. Klicken Sie dann auf die Schaltfläche **Einstellungen Datenquelle...**. Das folgende Fenster wird angezeigt:

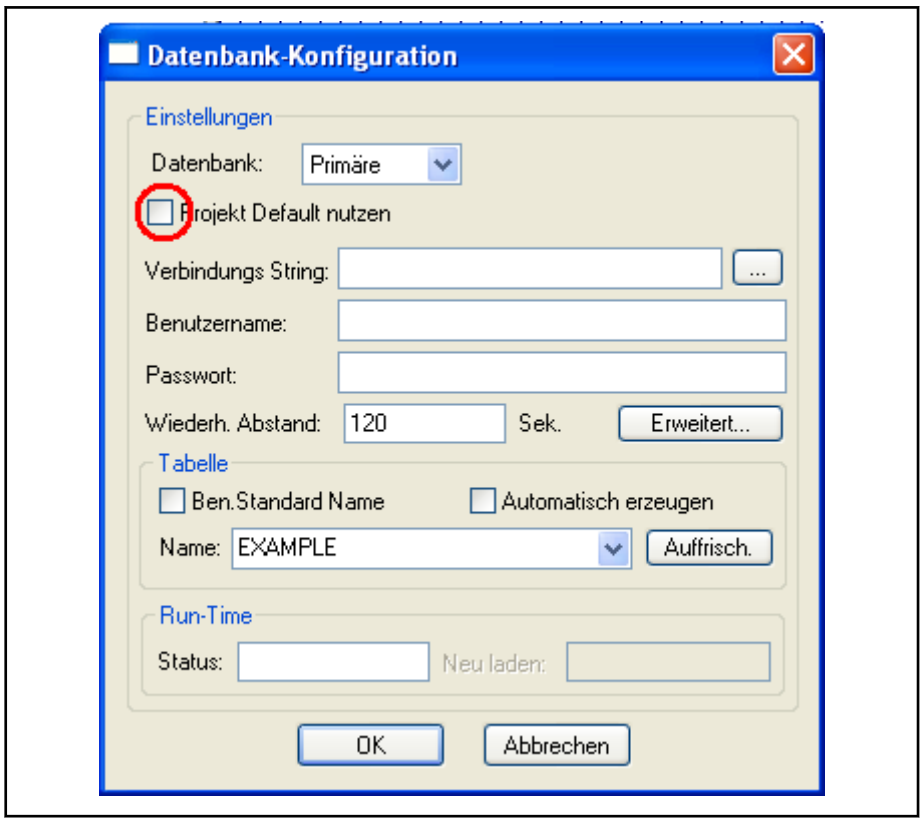


Abb.9-43: Löschen der Option Projektstandard verwenden

Deaktivieren Sie das Kontrollkästchen **Projektstandard verwenden** und klicken Sie auf die Suchschaltfläche **...** zum Konfigurieren des Verbindungsstrings. Das folgende Fenster wird angezeigt:

Alarme, Ereignisse und Trends

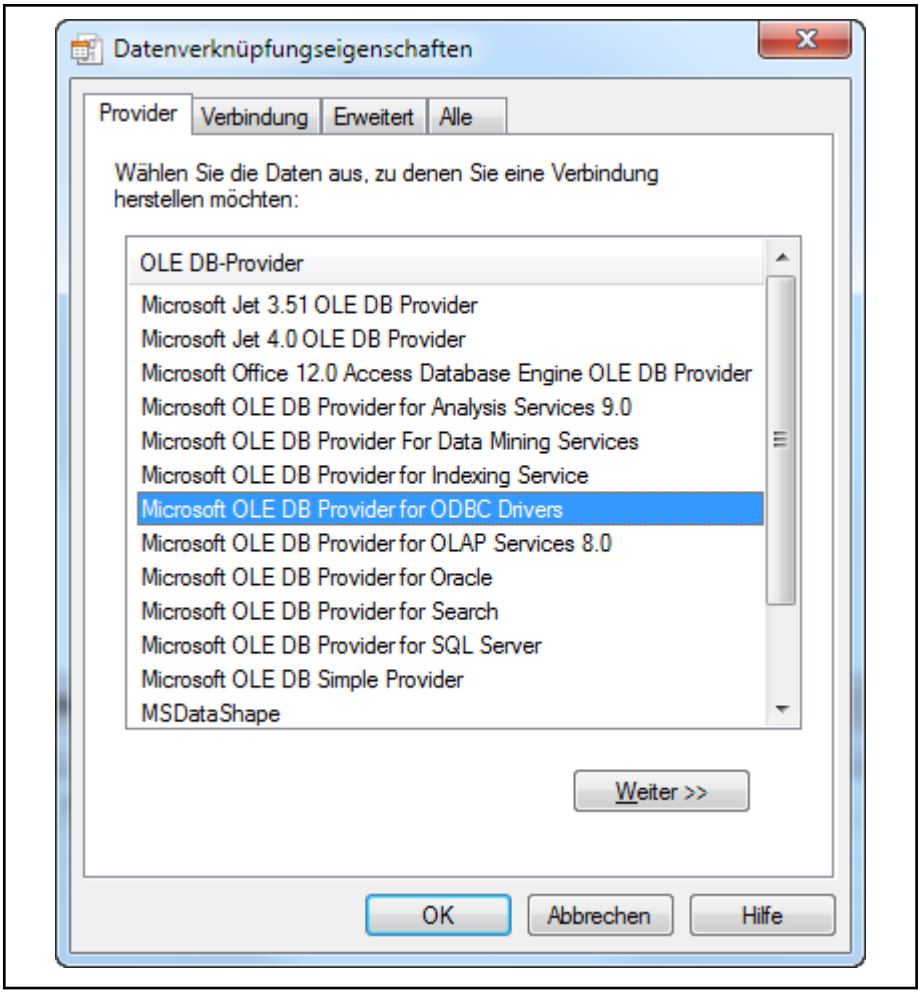


Abb.9-44: Auswahl des OLE Datenbankanbieters

Wählen Sie den Anbieter Microsoft 4.0 OLE DB aus und klicken Sie auf **Weiter »**. Im folgenden Fenster sollte der Datenbankpfad angegeben werden:

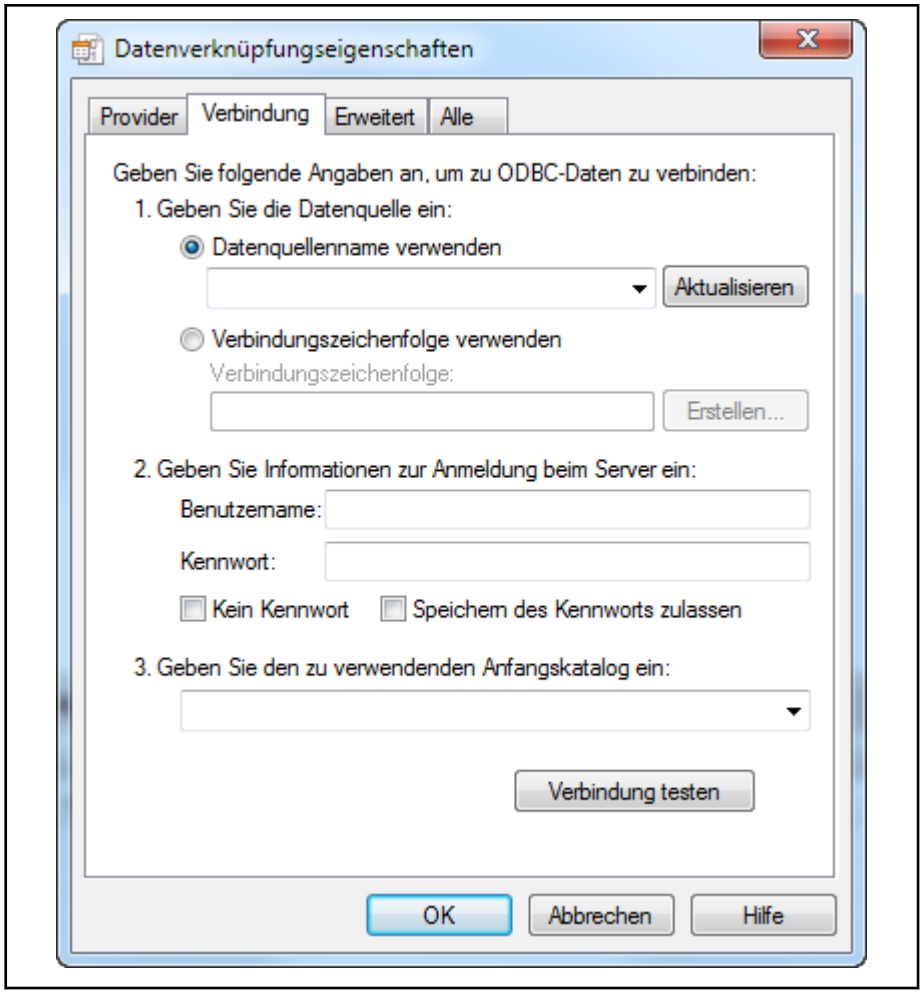


Abb.9-45: Auswahl der Datenbankdatei

Zum Beenden der Konfiguration des Verbindungsstrings klicken Sie auf **OK**. Deaktivieren Sie jetzt die Option **Standardname verwenden** und wählen Sie wie nachfolgend gezeigt die Tabelle aus Ihrer Datenbank aus:

Alarme, Ereignisse und Trends

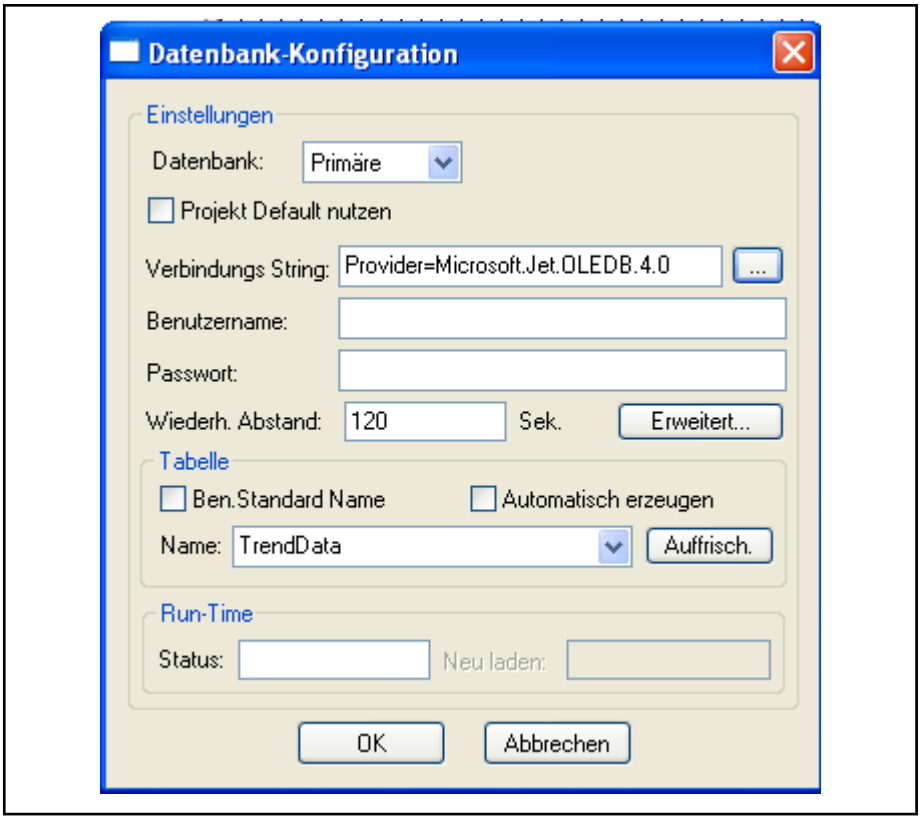


Abb.9-46: Auswahl der Tabelle in der Datenbank

Klicken Sie in diesem Fenster auf **OK** und nochmals auf **OK**, um die Konfiguration der Datenquelle abzuschließen und das Konfigurationsfenster "Datenquelle" zu schließen.

Jetzt müssen Temperatur und Druck definiert werden. Diese werden durch Punkte in der Trendanzeige dargestellt. Führen Sie nochmals einen Doppelklick auf das Trendobjekt aus, um zum Fenster Objekteigenschaften zu gelangen, und klicken Sie dann auf **Punkte...**. Im nächsten Schritt werden die Punkte gemäß der folgenden Abbildung definiert:

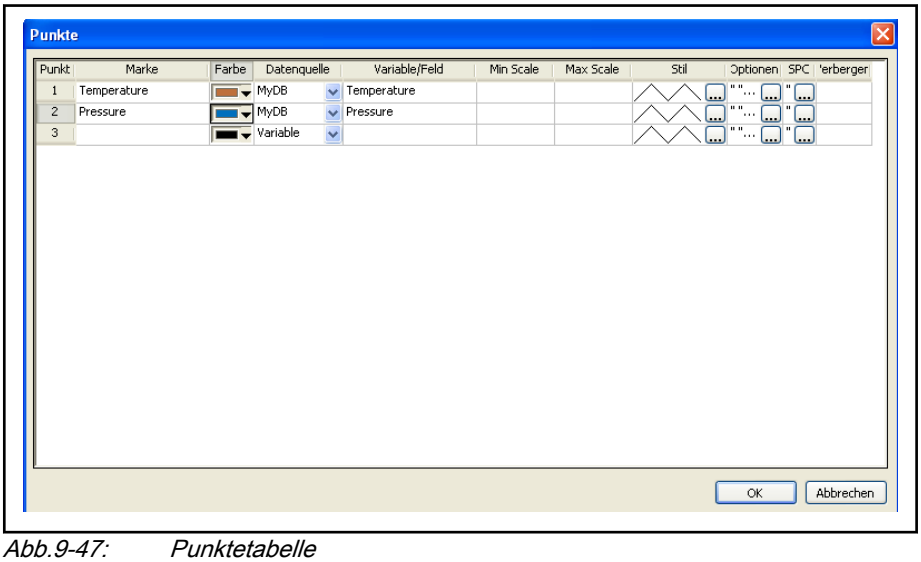


Abb.9-47: Punktetabelle

Wenn die Trendanzeige ausgeführt wird, startet diese mit aktuellem Datum/ Uhrzeit. Um die Daten in der Grafik darzustellen, müssen Startdatum/ -uhrzeit wie unten gezeigt korrekt konfiguriert werden:

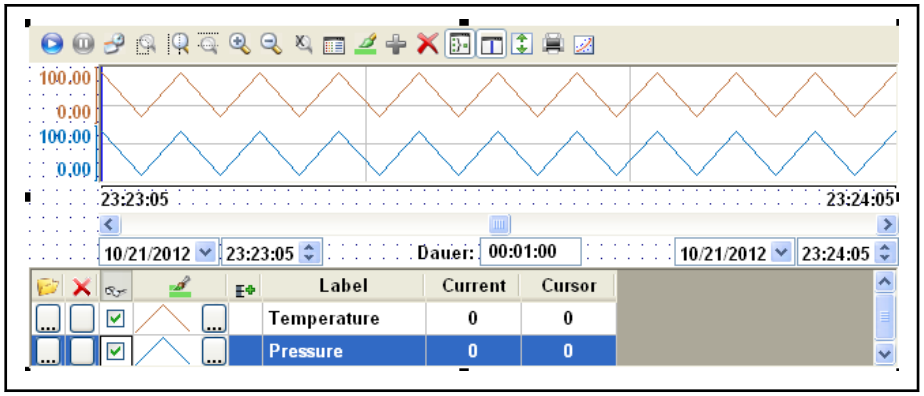


Abb.9-48: Diagramm

9.7 Tabellenobjekt

9.7.1 Tabellenobjekt, Allgemeines

Mit dem Tabellenobjekt können Daten im Tabellenformat aus der im Objekt konfigurierten Datenquelle gelesen oder dorthin geschrieben werden.

Um ein Tabellenobjekt zu zeichnen, gehen Sie wie folgt vor:

1. Klicken Sie im Werkzeugfenster "WinStudio Screen Tools" auf das Element "Tabelle"
2. Klicken Sie in das Bild, klicken Sie mit der linken Maustaste und ziehen die Maus über den Bildschirm, um einen Kasten in gewünschter Größe zu erstellen (gedrückt halten).
3. Wenn Sie die Maustaste loslassen, wird das Tabellenobjekt angezeigt.

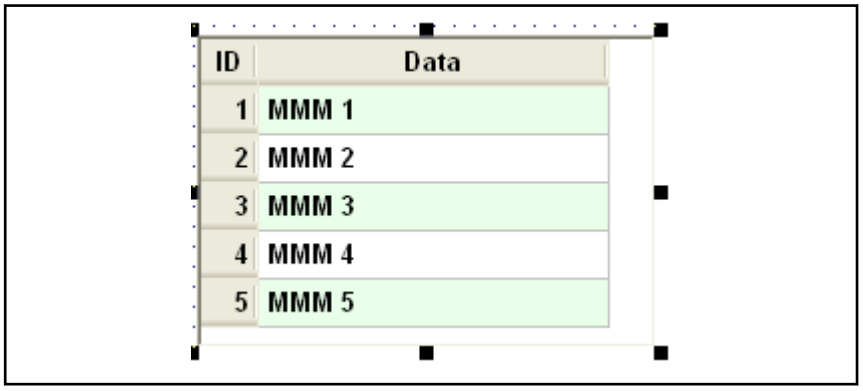


Abb.9-49: Beispiel für Tabellenobjekt

Führen sie einen Rechtsklick auf das Tabellenobjekt aus und wählen Sie aus dem Menü **Eigenschaften** aus. Der Dialog Objekteigenschaften wird geöffnet. Verwenden Sie diesen Dialog zum Konfigurieren der Parameter des Tabellenobjekts:

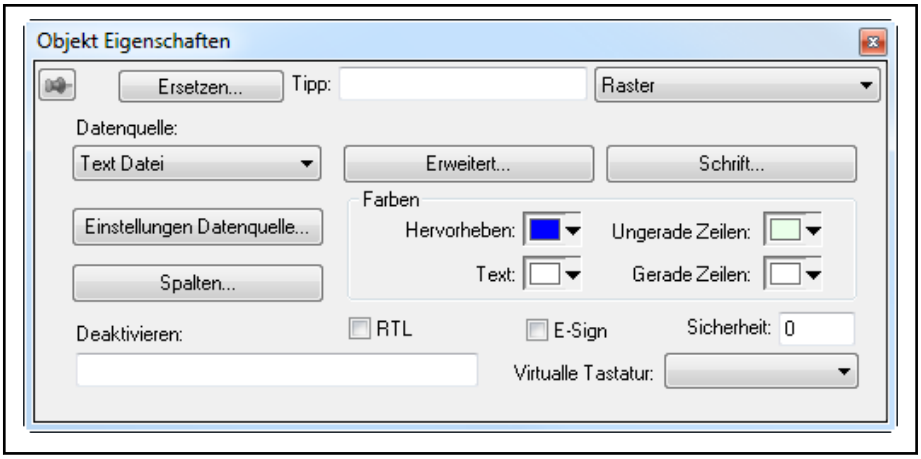


Abb.9-50: Objekteigenschaften: Tabelle

- **Datenquelle**: Wählen Sie den Datenquellentyp aus. Das Objekt unterstützt drei Datenquellen:

Datenquelle	Beschreibung
Textdatei	Zeigt Daten aus einer Textdatei im ASCII- oder Unicode-Format an (z.B. CSV-Textdateien).
Klassenvariable	Zeigt Werte aus einer Klassenvariable an, wobei die Klassenelemente des Tags Felder (Spalten) des Tabellenobjekts sind und jede Array-Position eine Zeile im Tabellenobjekt ist.
Datenbank	Zeigt Daten aus einer relationalen SQL-Datenbank an und verwendet ADO (ActiveX Database Object) für den Datenaustausch mit der Datenbank.

- **E-Sign**: Wird diese Option ausgewählt, wird der Benutzer aufgefordert, vor Eingabe oder Veränderung von Daten auf dem Objekt eine elektronische Signatur einzugeben.
- **VT**: Für dieses Objekt wird der Typ "Virtuelle Tastatur" verwendet. Die Option <Standard verwenden> wählt das standardmäßige virtuelle Keyboard aus, das in den Einstellungen "Viewer" (Register **Viewer** im Eigenschaftendialog des Ordners "WinStudio") konfiguriert ist.

  Sie können für dieses Tabellenobjekt auch eine andere virtuelle Tastatur festlegen.
- **Sicherheit**: Für das Objekt bzw. die Animation erforderliche Zugangsebene.
- **Deaktivieren**: In diesem Feld kann ein Ausdruck zum Deaktivieren von Dateneingabe oder Aktion durch den Benutzer eingegeben werden.
- **Markierungsfarbe**: Wählen Sie eine Hintergrundfarbe für die markierte Zeile, während der Laufzeit.
- **Textfarbe**: Wählen Sie eine Textfarbe für die markierte Zeile, während der Laufzeit.
- **Ungerade Zeilen**: Wählen Sie eine Hintergrundfarbe für die ungeraden Zeilen aus.
- **Gerade Zeilen**: Wählen Sie eine Hintergrundfarbe für die geraden Zeilen aus.
- **Schrift**: Klicken zum Öffnen des Dialogs "Schriftarten", in dem die Schriftarteinstellungen für den im Tabellenobjekt angezeigten Text konfiguriert werden können.
- **Spalten**: Klicken zum Aufrufen des Dialogs Spalten auf Seite 347, in dem die Einstellungen (z.B. Beschriftungen, Spalte, Breite usw.) für die Spalten des Tabellenobjekts konfiguriert werden können.

- **Einstellungen Datenquelle**: Klicken zum Öffnen des Dialogs Daten auf Seite 349, in dem eine Datenquelle für das Tabellenobjekt festgelegt werden kann.
- **Erweitert**: Klicken zum Öffnen des Dialogs Weitere Einstellungen auf Seite 351, in dem verschiedene Einstellungen für das Tabellenobjekt festgelegt werden können.

9.7.2 Dialog Spalten

Sie können die Einstellungen für jede vom Rasterobjekt auf Seite 345 angezeigte Spalte während der Laufzeit wie folgt konfigurieren:

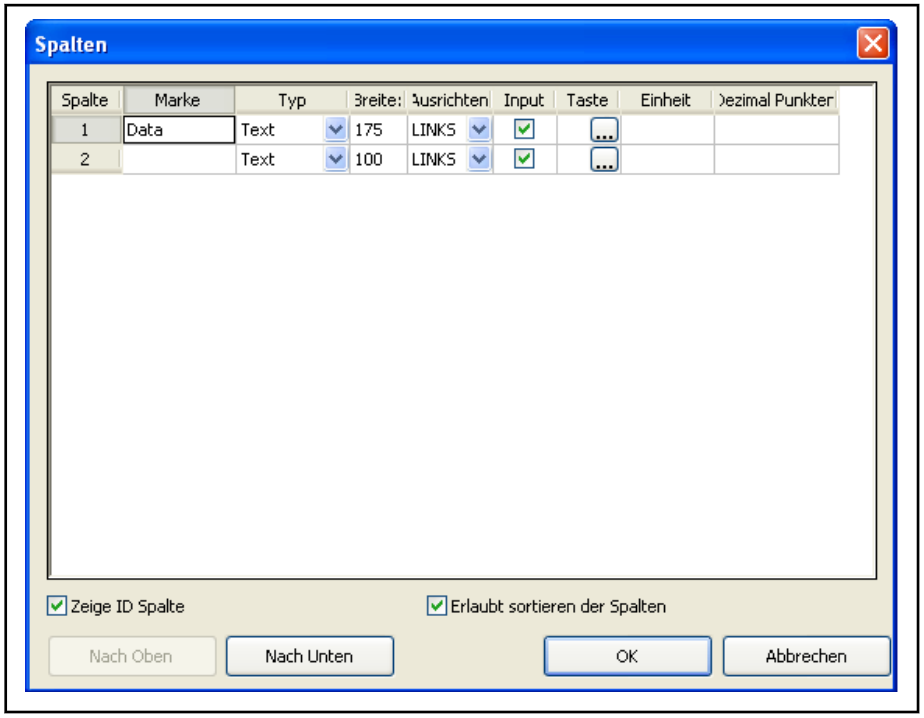


Abb.9-51: Dialog Spalten

- **Spalte**: Die Kennnummer definiert die Position der Spalte in der Tabelle.
- **Titel**: Geben Sie für jede Spalte einen Titel ein, der als oberste (erste) Zeile des Tabellenobjekts angezeigt wird.

Im Feld **Tabellenobjekt** können Tags in geschweiften Klammern konfiguriert werden, um das Feld während der Laufzeit dynamisch zu ändern. Bleibt die Beschriftung leer (z.B. ""), so wird die Breite der Spalte während der Laufzeit auf 0 gesetzt. Diese Option ist nützlich zum Verbergen von Spalten während der Laufzeit.

- **Feld**: Geben Sie den Namen des Feldes (der Spalte) in der relationalen SQL-Datenbank ist ein, mit der das Tabellenobjekt verknüpft ist. Bleibt das Feld leer, wird der im Feld **Bezeichnung** konfigurierte Text als Feldname verwendet. (Diese Einstellung ist nur verfügbar, wenn der **Datenquellen**-Typ auf Datenbank eingestellt ist.)
- **Typ**: Geben Sie die Art der Schnittstelle zur Verwendung in dieser Spalte an. Die folgenden Optionen sind verfügbar:

Alarme, Ereignisse und Trends

Typ	Beschreibung
Text	Zeigt alphanumerische Werte an
Numerisch	Zeigt numerische Werte an
Bild	Zeigt das Bild (***.bmp** oder ***.ico** Format) aus der Datenquelle an. Wenn zum Beispiel der Wert aus der Datenquelle **MeineDatei.bmp** ist, zeigt das Tabellenobjekt das Bild aus der Datei **MeineDatei.bmp** in Ihrem Projektordner an. Die Bildgröße wird automatisch an die Zelle des Tabellenobjekts angepasst. Die Bilddatei(en) muss/müssen im Unterordner **Web** Ihres Projektordners gespeichert werden, um diese Funktion auf den Thin-Client-Stationen zu unterstützen. CEView-Projekte unterstützen Bilder im Bitmap-Format (***.bmp**), aber nicht im Icon-Format (***.ico**).
Kontrollkästchen	Zeigt ein Kontrollkästchen an. Dieses Kontrollkästchen wird deaktiviert, wenn der aus der Datei ausgelesene Wert 0, <NULL> oder „FALSCH“ ist, andernfalls wird das Kontrollkästchen aktiviert. Standardmäßig verwendet "WinStudio" den Wert 0 für deaktiviert und den Wert 1 für aktiviert.
Zeit	Zeigt den Wert im Zeitformat an (z.B. HH:MM:SS). Diese Einstellung ist nur verfügbar, wenn der **Datenquellen**-Typ auf **Datenbank** eingestellt ist.
Datum	Zeigt den Wert im Format Datum an (z.B. MM/TT/JJJJ). Diese Einstellung ist nur verfügbar, wenn der **Datenquellen**-Typ auf **Datenbank** eingestellt ist.
Datum/ Uhrzeit	Zeigt den Wert im Format Datum/Uhrzeit an (z.B. MM/TT/JJJJ HH:MM:SS). Diese Einstellung ist nur verfügbar, wenn der **Datenquellen**-Typ auf **Datenbank** eingestellt ist.



- Wenn der Typ **Datenquelle** auf Datenbank eingestellt ist, muss unbedingt sichergestellt werden, dass der **Typ** für jede im Objekt konfigurierte Spalte dem Type des entsprechenden Felds in der Datenbank entspricht.
- Wenn der Typ **Datenquelle** auf Datenbank eingestellt ist, können gültige SQL-Aussagen direkt im Feld konfiguriert werden (z.B.

  List(DISTINCT [Cell_Name]) AS [Cell_Name]).

  Sie können zum Ändern dieser Einstellung während der Laufzeit auch Tag-Namen in geschweiften Klammern konfigurieren (z.B. **{MeinFeldName}**).



Wenn der Spaltentyp als Bild festgelegt ist, zeigt das Rasterobjekt ein Standardsymbol an ( ), wenn die Bilddatei während der Laufzeit nicht gefunden wird. Sie können ein anderes Bild zur Anzeige bei Nichtauffinden der Datei konfigurieren, indem Sie die Bilddatei in den Unterordner "Web" Ihres Projektordners kopieren und deren Namen in der Datei "projekt_name.app" wie folgt konfigurieren:

[Objekte]

GridDefaultPicture=BildDateiName

- **Breite**: Geben Sie eine Breite für die Spalte ein, in Pixeln
- **Ausrichtung**: Wählen Sie eine Ausrichtung für die in der Spalte angezeigten Daten aus. Dazu gibt es drei Möglichkeiten: Links, rechts oder zentriert
- **Eingabe**: Durch Anklicken (Aktivieren) kann der Benutzer während der Laufzeit in dieser Spalte Daten eingeben

- **Taste**: Verwenden Sie dieses Feld zum Zuweisen eines Tastaturkürzels zum Sortieren der Werte. Ein Tastaturkürzel ist eine Tastenkombination, die gleichzeitig auf einer Tastatur gedrückt wird (z.B. <STRG>+<C>, <STRG>+<V> usw.). Diese Option ist besonders hilfreich beim Erstellen von Projekten für Laufzeitgeräte, die keine Maus oder keine Touchscreen-Schnittstelle zur Verfügung stellen und bei denen zum Eingreifen während der Laufzeit nur die Tastatur zur Verfügung steht.
- **Einheit**: Geben Sie den Namen der Einheit (d.h. der Maßeinheit) ein, die auf die angezeigten Daten zutrifft. Es kann auch ein String-Tag mit der Syntax **{tagname}** eingegeben werden, mit dem der Wert von **Einheit** während der Laufzeit geändert werden kann.
- **Dezimalstellen**: Geben Sie die Anzahl der Dezimalstellen ein, die angezeigt werden sollen. Es kann auch ein Integer-Tag mit der Syntax **{tagname}** eingegeben werden, mit dem der Wert von **Dezimalstellen** während der Laufzeit geändert werden kann



Wenn der **Datenquellentyp** auf Klassen-Tag festgelegt ist und der Dialog Spalten leer bleibt, zeigt das Objekt die Werte aller Elemente des Klassen-Tags mit den folgenden Standardeinstellungen für die Spalten an:

- Titel = <Name des Members aus dem Klassen-Tag>
- Type = Text
- Breite= < Mindestgröße zur Anzeige des Namens des Members aus dem Klassen-Tag im Kopfbereich des Rasterobjekts>
- Ausrichtung = Mitte
- Eingabe = aktiviert (angekreuzt)
- Taste = <Keine>
- Einheit = <Einheit des Elements der Klassenvariable>

Die Einheit eines Klassen-Members oder -Tags wird mithilfe des Werkzeugs Tag-Eigenschaften auf Seite 136 festgelegt.

- **Spalten-ID anzeigen**: Aktivieren, um die Nummer jeder Zeile automatisch anzuzeigen.
- **Spaltensortierung erlauben**: Aktivieren, um dem Benutzer das Sortieren der Werte in den Spalten während der Laufzeit zu ermöglichen, entweder durch Klicken auf die Bezeichnung oder unter Verwendung des für jede Spalte konfigurierten Tastaturkürzels. Diese Option ist deaktiviert, wenn die Option **Titelbereich anzeigen** aus dem Dialog weitere Einstellungen auf Seite 351 nicht aktiviert ist.



Verwenden Sie die Schaltflächen **Nach oben** und **Nach unten**, um die Anzeigereihenfolge der Spalten neu zu ordnen.

9.7.3 Dialog Daten

Mit diesem Dialog kann die Datenquelle für ein Tabellenobjekt konfiguriert werden.

Tabellendaten – Textdatei

Wenn **Datenquelle** auf Textdatei gesetzt ist, können die folgenden Einstellungen konfiguriert werden:

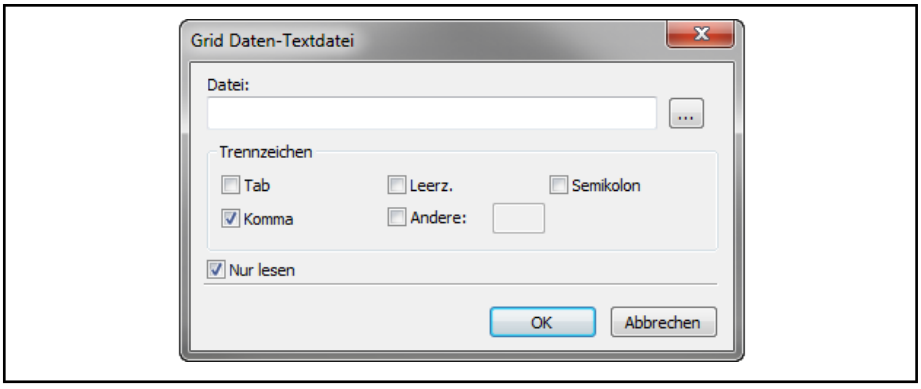


Abb.9-52: Textdatei Tabellendaten

- **Datei**: Namen der Quelltextdatei angeben. Sie können entweder den Dateinamen und dessen Pfad eingeben oder zum Suchen auf die Schaltfläche **...** klicken. (Wenn die Datei in Ihrem Projektordner gespeichert ist, kann der Pfad im Namen ausgelassen werden.)



Im Feld **Datei** können Tag-Namen in geschweiften Klammern eingegeben werden (z.B. {TagName}).

- **Trennzeichen**: Legen Sie den/die in der Daten Quelldatei verwendeten Trennzeichen fest. Wenn zum Beispiel die Daten aus einer CSV-Datei (comma separated values) gelesen werden, ist die Option **Komma** auszuwählen. Durch Auswählen der Option **Andere** und Eingabe des benutzerdefinierten Trennzeichens im benachbarten Feld kann sogar ein benutzerdefiniertes Trennzeichen ausgewählt werden.
- Kontrollkästchen **Nur lesen**: Wenn diese Option aktiviert ist, liest das Tabellenobjekt nur Daten aus der angegebenen Datei aus. Das Objekt schreibt nichts in die Datei.

Tabellendaten – Klassenvariable

Wenn **Datenquelle** auf Klassenvariable gesetzt ist, können die folgenden Schnittstellen konfiguriert werden:

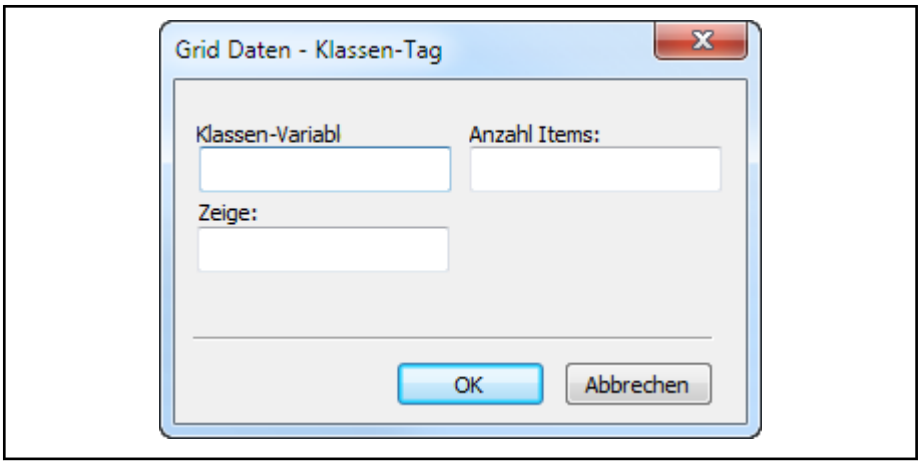


Abb.9-53: Tabellendaten – Klassenvariable

- **Klassenvariable**: Geben Sie den Namen der Hauptquelle für Klassen-Tag ein. (Geben Sie kein spezifisches Member vom Typ Klasse ein.) In diesem Feld kann die Ausgangsbereichsposition festgelegt werden (z.B. "MeinTag[10]"); andernfalls wird standardmäßig 0 (null) als Ausgangsposition verwendet.

- **Anzahl der Elemente**: Geben Sie die Anzahl der Array-Positionen der **Klassevariable** ein, die angezeigt werden sollen.
- **Ansicht**: Wenn der im optionalen Feld konfigurierte Tag während der Laufzeit seinen Wert ändert (z.B. umschaltet), öffnet das Tabellenobjekt einen Dialog, mit dem der Benutzer jede Spalte anzeigen/verbergen oder deren Positionen verändern kann.

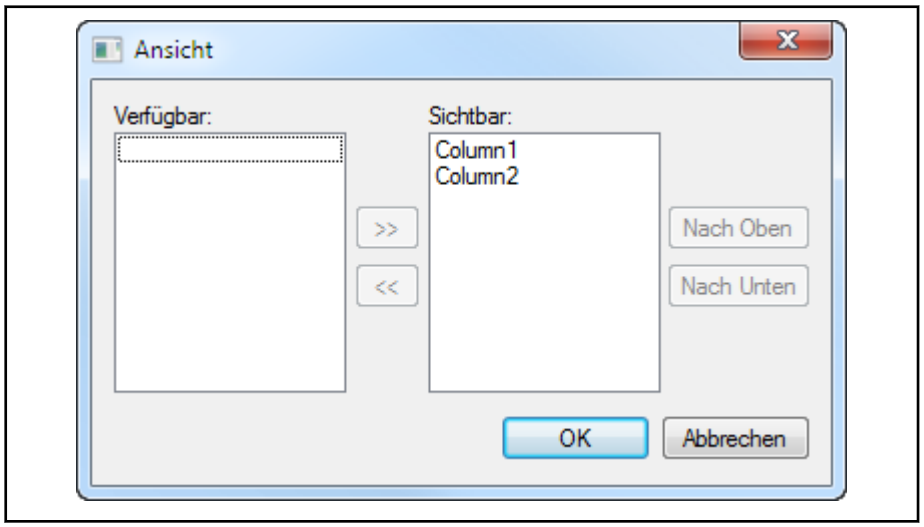


Abb.9-54: Dialog: Ansicht

Tabellendaten – Datenbankkonfiguration

Wenn **Datenquelle** auf Datenbank gesetzt ist, können die folgenden Einstellungen konfiguriert werden:

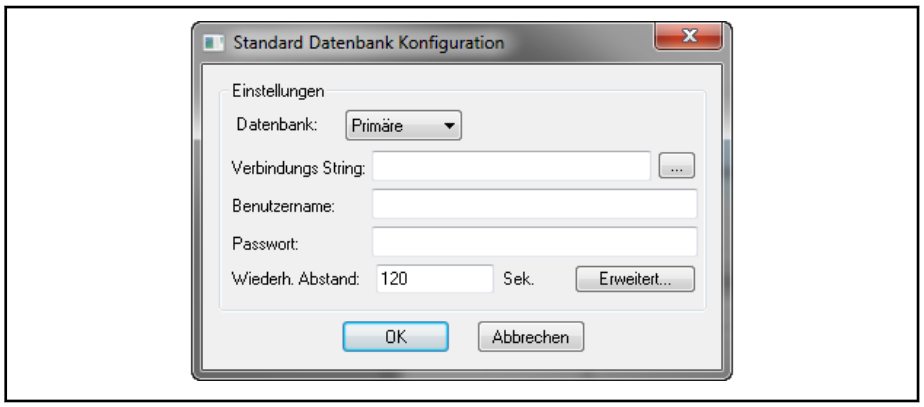


Abb.9-55: Tabellendaten – Datenbankkonfiguration

Weitere Informationen zu diesem Dialog finden Sie unter Dialog Datenbankkonfiguration auf Seite 502.

9.7.4 Dialog Erweiterte Einstellungen

Mit diesem Dialog können die erweiterten Einstellungen für ein Tabellenobjekt konfiguriert werden.

Alarme, Ereignisse und Trends

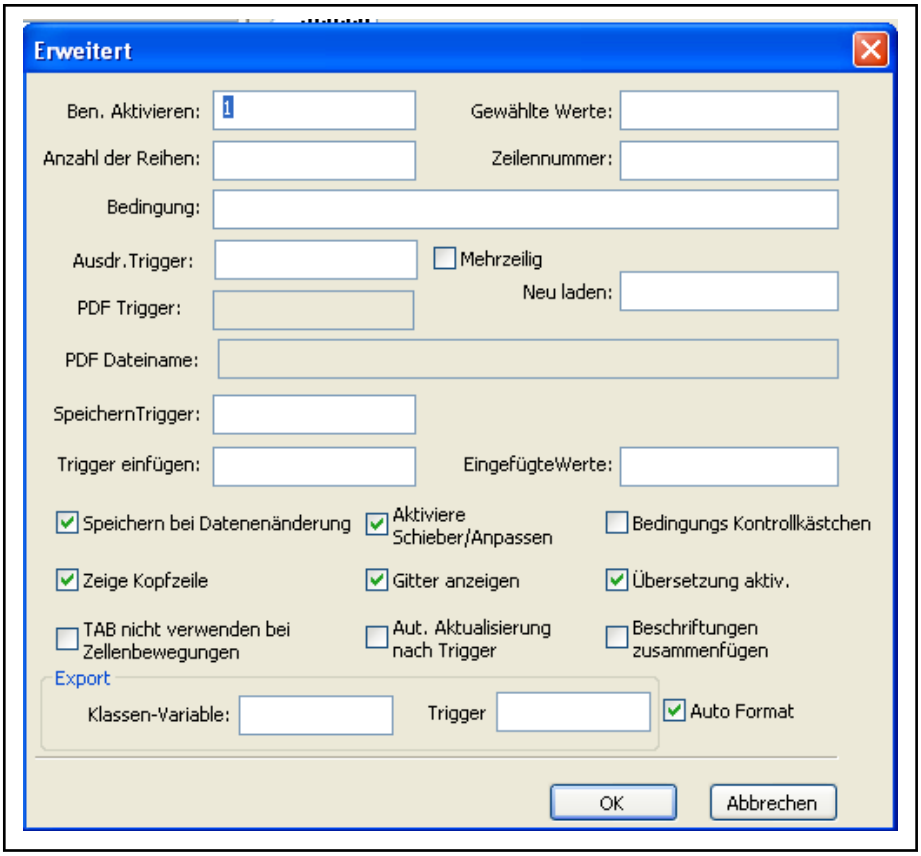


Abb.9-56: Dialog Weitere Einstellungen

- **Benutzer freigeben**: Wenn der Wert dieses Tags WAHR ist (ungleich 0), kann der Benutzer während der Laufzeit verschiedene Zeilen des Objekts durch Anklicken auswählen. Dieses Feld kann mit einem Tag oder mit einem numerischen Wert konfiguriert werden.
- **Ausgewählte Werte**: Die Werte aus jeder Spalte der ausgewählten Zeile werden in jede Position des in diesem Feld konfigurierten Array-Tags geschrieben. Sie können außerdem den Wert der aktuell im Tabellenobjekt markierten Zellen durch Ändern des Werts des in diesem Feld konfigurierten Array-Tags verändern. Die Ausgangsposition (Versatz) kann in diesem Feld konfiguriert werden.
- **Zeilenanzahl** : Das Tabellenobjekt schreibt die Anzahl der aktuell im Tabellenobjekt verfügbaren Zeilen in den in diesem Feld konfigurierten Tag.
- **Zeilennummer**: Das Tabellenobjekt schreibt während der Laufzeit die Nummer der aktuell markierten Zeile. Zusätzlich können verschiedene Zeilen durch Schreiben ihrer Werte in diesen Tag ausgewählt werden.
- **Bedingung**: Geben Sie einen Ausdruck zum Filtern der Tabellendaten ein; es werden nur Zeilen angezeigt, die dem Ausdruck entsprechen. Der Ausdruck muss die folgende Syntax verwenden:

  Wert Spaltenoperator

  Zum Beispiel...

  SpalteX > 200

Wenn **Datenquelle** (im Dialog Tabellenobjekteigenschaften auf Seite 345) auf Textdatei oder Klassenvariable eingestellt ist, ist die "Spalte" der in der Bezeichnung festgelegte Wert. Wenn **Datenquelle** auf Datenbank eingestellt ist, ist die Spalte der im Feld festgelegte Wert. (Wenn in diesem Fall das Feld leer bleibt, ist der angegebene Spaltenwert die Bezeichnung.)

Außerdem müssen Ausdrücke für Datenbank wie ein SQL-Where-Ausdruck formatiert werden. Die folgende Tabelle zeigt, welche Operatoren verwendet werden sollten:

Operatoren für Bedingungsausdrücke

Vergleich	Datenquelle ist Textdatei...	Datenquelle ist Datenbank...
ist gleich	=	LIKE
ist ungleich	<>	NOT LIKE
Platzhalter, einzelnes Zeichen	?	_
Platzhalter, unbegrenzte Zeichen	*	%

Somit bedeutet der folgende Ausdruck für Textdatei...

```
C1 = 'ab?d'
```

... das gleiche wie der folgende Ausdruck für Datenbank...

```
C1 LIKE 'ab_d'
```

Schließlich können verschiedene Ausdrücke gleichzeitig im Feld **Bedingung** kombiniert werden, unter Verwendung der logischen Operatoren "AND", "OR" und "NOT" z. B.:

```
SpalteAlter> '10' OR SpaltenName= 'John' AND SpalteDatum>
'05/20/2003'
```



Sie können im Feld **Bedingung** Tags in geschweiften Klammern konfigurieren **{TagName}**, um die Filterbedingungen während der Laufzeit zu ändern.

- **Trigger für Drucken**: Wenn der in diesem Feld konfigurierte Tag umgeschaltet wird, wird der aktuelle Status des Tabellenobjekts an den Standarddrucker gesendet.
- Feld **Trigger für PDF**: Wenn der in diesem Feld konfigurierte Tag umgeschaltet wird, wird der aktuelle Status des Tabellenobjekts als PDF-Datei an dem unter **PDF Dateiname** festgelegten Ort gespeichert.
- Feld **PDF Dateiname**: Geben Sie einen vollständigen Dateipfad für das Speichern der PDF-Datei ein. Es kann auch ein Tag-Name mit der Syntax **{tag}** eingegeben werden.



Trigger für PDF und **PDF Dateiname** werden in Projekten, die auf Windows Embedded oder Thin Client laufen, nicht unterstützt.

- Kontrollkästchen **Mehrzeilig**: Wenn diese Option angewählt ist, wird der Ausdruck oder das PDF gemäß der verfügbaren Spaltenbreite formatiert und der Text in jeder Zelle wird umgebrochen, so dass der gesamte Text zu sehen ist.

- **Neu laden**: Wenn der in diesem Feld konfigurierte Tag umgeschaltet wird, lädt das Objekt die Daten erneut aus der Datenquelle und zeigt sie an.
- **Trigger für Speichern**: Wenn der in diesem Feld konfigurierte Tag umgeschaltet wird, wird die Datenquelle (Textdatei oder Datenbank) mit den aktuellen Werten des Rasterobjekts aktualisiert. Dieses Feld ist nicht verfügbar, wenn die Datenquelle vom Typ Klassenvariable ist, da beim Ändern einer Zelle in der Tabelle die Werte in den Tags automatisch aktualisiert werden.
- **Trigger für Einfügen**: Wenn **Automatisch aktualisieren nach Auslöser Einfügen** aktiviert (ausgewählt) ist, wird der in diesem Feld konfigurierte Tag als Auslöser zum Aktualisieren der Datenbanktabelle verwendet. Wenn sich der Tag-Wert ändert, wird der Tabelle eine neue Zeile hinzugefügt und die im Feld **Eingefügte Werte** konfigurierten Werte des Arrays werden automatisch eingefügt.
- **Einzufügende Werte**: Bei Verwendung von **Trigger für Einfügen** liefert der in diesem Feld konfigurierte Trigger die einzufügenden Werte. Dieses Feld darf nur ein Array-Tag enthalten, kann aber eine beliebige Größe haben.
- **Speichern bei Datenänderung**: Wenn diese Option gewählt ist, werden die Werte in der Datenquelle (Textdatei oder Datenbank) aktualisiert, sobald der Benutzer während der Laufzeit einen neuen Wert in der Tabelle eingibt. Diese Option ist deaktiviert, wenn die Datenquelle vom Typ Klassenvariable ist, da beim Ändern der Zellwerte in der Tabelle durch den Benutzer die Werte in den Tags automatisch aktualisiert werden.
- **Aktiviere Schieber/Größenänderung**: Wenn dieses Kästchen nicht aktiviert ist, kann der Benutzer während der Laufzeit die Schieber-Schaltfläche nicht zum Blättern in der Liste verwenden oder die Größe der Zelle verändern.
- **Bedingtes Kontrollkästchen**: Wenn diese Option aktiviert ist, kann der Benutzer während der Laufzeit ein Kontrollkästchen im Raster erst dann deaktivieren, wenn alle vorangehenden Kontrollkästchen in der gleichen Spalte bereits deaktiviert sind. Diese Option ist besonders hilfreich, wenn der Benutzer einer vorbestimmten Sequenz folgen soll. Dieses Feld ist nicht verfügbar, wenn für die Datenquelle der Typ Klassenvariable gewählt ist.
- **Titelbereich anzeigen**: Wenn diese Option aktiviert ist, ist der Titelbereich des Tabellen während der Laufzeit sichtbar und zeigt die Beschriftung für jede Spalte an.
- **Gitter anzeigen**: Wenn diese Option aktiviert ist, sind die Rasterlinien des Tabellenobjekts während der Laufzeit sichtbar.
- **Übersetzung aktiv**: Wenn diese Option aktiviert ist, unterliegt der vom Rasterobjekt angezeigte Text während der Laufzeit dem Übersetzungswerkzeug.
- **TAB für Zellennavigation deaktivieren**: Wenn diese Option aktiviert ist, kann der Benutzer sich nur mit den Pfeiltasten durch die Zellen des Tabellenobjekts bewegen, und nicht mit der Tabulatortaste. Die Tabulatortaste sollte für die Navigation deaktiviert werden, wenn sie zum Wechseln zum nächsten aktivierbaren Objekt im Bild verwendet werden soll.
- **Beschriftung für Bild verknüpfen**: Wenn diese Option aktiviert ist, ist der Bezugsname für das Bild das Ergebnis der Verknüpfung des Namens in der Spalte Feld mit dem Wert der Spalte Bezeichnung. Das Ergebnis ist <Bezeichnung Name>_<Feld Wert>.

- **Export**: Mit dieser Schnittstelle können die Daten aus dem Tabellenobjekt in einen Klassen-Array-Tag exportiert werden, ungeachtet der für das Objekt gewählten Datenquelle. Für diese Funktion müssen die folgenden Felder konfiguriert werden:

Feld	Beschreibung
Array-Tag	Geben Sie den Haupt-Tag-Namen des Klassen-Array-Tags ein, der die exportierten Werte erhält. Jede Zeile aus dem Rasterobjekt wird in eine Bereichsposition des Bereichs Times exportiert, über die entsprechenden Spaltenbezeichnungen. Die Ausgangsbereichsposition kann in diesem Feld konfiguriert werden; der Standardwert ist 0.
Trigger	Wenn der in diesem Feld konfigurierten Tag seinen Wert ändert (z.B. umschaltet), werden die Daten von dem Tabellenobjekt in den im Feld Klassen-Tag konfigurierten Klassen-Array-Tag exportiert.



Die Exportfunktion ist ein einfaches und leistungsstarkes Werkzeug für die Datenübertragung von verschiedenen Datenquellen in Tags. Nach dem Exportieren der Daten in Tags können verschiedene Tasks zur Handhabung der Daten verwendet werden, zum Beispiel die Funktion FileWrite auf Seite 659 oder die Tasks Rezeptur oder Bericht zum Speichern der Daten in Textdateien (z.B. CSV-Dateien).

- **Auto-Format**: Wenn diese Option aktiviert ist, werden Dezimalwerte in Spalten vom Typ **Numerisch** gemäß der von der Funktion SetDecimalPoints auf Seite 681 erzeugten virtuellen Tabelle formatiert. Diese Option funktioniert nur in Spalten, für die noch keine **Dezimalstellen** konfiguriert sind. Weitere Informationen unter Dialog Rasterobjekt: Spalten auf Seite 347.

10 Hintergrundtasks

10.1 Hintergrundtasks, Allgemeines

Hintergrundtasks sind, wie der Name schon sagt, im Hintergrund ausgeführte Projektbestandteile, im Gegensatz zu den grafischen Bildern, mit denen der Benutzer interagiert.

Die Hintergrundtasks werden vom Modul Hintergrundtasks (siehe <Taskausführung>) ausgeführt und werden durch Task-Arbeitsblätter im Projekt-Explorer definiert.

10.2 Alarmarbeitsblatt, Allgemeines

Der Ordner Alarme ermöglicht die Konfiguration von Alarmgruppen und Tags für jede Gruppe. Das Arbeitsblatt Alarm definiert die von der Anwendung erzeugten Alarmmeldungen. Der Hauptzweck eines Alarms besteht darin, den Bediener über Probleme oder ungewöhnliche Zustände während des Prozesses zu informieren, damit er korrigierende Maßnahmen einleiten kann.

Das Arbeitsblatt Alarm wird durch das Modul Hintergrundtask ausgeführt (siehe <Ausführung Tasks>). Es verwaltet den Zustand aller Alarme und speichert die Alarmmeldungen in der Historie, falls dies konfiguriert wurde. Das Alarm-Arbeitsblatt selbst zeigt jedoch nicht die Alarmmeldungen in der Laufzeit an. Zum Anzeigen von Alarmen muss das Bildobjekt Alarm-/Ereignissteuerung auf Seite 290 aus dem Werkzeugfenster "WinStudio Screen Tools" in ein WinStudio-Bild eingefügt und konfiguriert werden.

Zum Erstellen eines neuen Alarmarbeitsblattsöffnen Sie das Kontextmenü des Ordners "Tasks" und wählen Sie **Hinzufügen ▸ Alarm**.

Zum Bearbeiten eines bestehenden Alarmarbeitsblatts führen sie im Projektexplorer einen Doppelklick auf das Element aus.

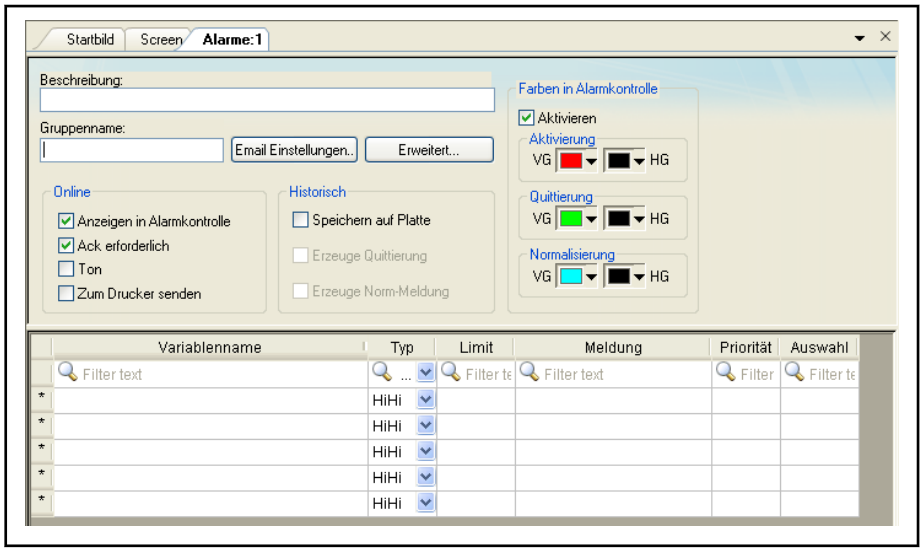


Abb. 10-1: Alarmarbeitsblatt

Es können mehrfache Alarmgruppen (Arbeitsblätter) erstellt werden und jede Gruppe kann mit unabhängigen Einstellungen wie Meldungsfarbe, Aktivierung/Deaktivierung historisches Protokoll usw. konfiguriert werden.

Jedes Alarmarbeitsblatt besteht aus zwei Bereichen:

- **Titelbereich:** Auf alle in der gleichen Alarmgruppe konfigurierten Tags und Alarme angewendete Einstellungen. Diese Einstellungen erlauben die Konfiguration der Meldungsformatierung und der Aktionen, die aufgrund von Alarmereignissen ausgelöst werden müssen (z.B. Alarm drucken, Alarm per E-Mail verschicken usw.). Weitere Informationen finden Sie in Einstellungen Titelbereich auf Seite 275.
- **Hauptteil:** Tags auswählen und Bedingungen definieren, bei denen Alarmmeldungen erzeugt werden. Weitere Informationen finden Sie in Einstellungen Hauptbereich auf Seite 280.

☞
- Die Alarmgruppe kann zum automatischen Versenden von E-Mail-Benachrichtigungen aufgrund von Alarmereignissen konfiguriert werden. Weitere Informationen finden Sie in E-Mail-Einstellungen auf Seite 276.
- Die mit jedem Tag verbundenen Alarmeigenschaften (im Hauptteil der Alarmgruppe konfiguriert) können auch mit dem Dialog Tag-Eigenschaften auf Seite 153 bearbeitet werden (**Tag-Eigenschaften** im Werkzeugfenster "Querverweise"). Allerdings ist es vor dem Verbinden eines Tags mit einer Alarmgruppe erforderlich, die Alarmgruppe zu erstellen und die Einstellungen in deren Kopfteil zu konfigurieren. Diese werden auf alle mit der Gruppe verbundenen Tags angewendet.
- Ab "WinStudio" v6.1+SP2 wurde der Alarm-Task verändert, um das automatische Bestätigen eines Alarms durch einen anderen Alarm zu verhindern. Zum Beispiel sollte der Hi- bzw. Lo-Alarm nicht automatisch bestätigt werden, wenn der HiHi- bzw. LoLo-Alarm und aktiviert wird. Zum Aktivieren des vorgenannten Verhaltens setzen Sie Ihrer Projektdatei (**.APP**) den folgenden Schlüssel:

  [Alarm]

  UseLegacyPriorityAck=1

HINWEIS

Diese im Hauptteil jedes Alarmarbeitsblatts konfigurierten Einstellungen werden in den Tag-Datenbank-Archiven gespeichert. Daher können Änderungen in der Tag-Datenbank den Inhalt der Alarm-Arbeitsblätter (Hauptteil) beeinflussen.

Beachten Sie, dass jeder Tag/Typ nicht in mehr als einer Alarmgruppe gleichzeitig verfügbar sein kann, da die Alarmgruppe eine mit jedem Tag-/Alarmtyp verbundene Eigenschaft ist

(z.B. Tag: Stufe; Alarm-Typ: Hi; Alarmgruppe: 2).

10.3 Trendarbeitsblatt, Allgemeines

Der Ordner "Trend" ermöglicht das Konfigurieren von Historiengruppen, die Trendkurven speichern. Sie können das Trendarbeitsblatt zum Deklarieren verwenden, welche Tags ihre Werte auf die Festplatte speichern, und zum Anlagen von Historiedateien für Trendgraphen. Ihr Projekt speichert die

Stichproben in einer binären Historiedatei (*.hst) und zeigt die historischen und die Online-Stichprobenwerte in einem Trendgraph-Bild an.

Das Trendarbeitsblatt wird durch das Modul Hintergrundtask ausgeführt. Es verwaltet das Speichern von Trenddaten in der Historie, zeigt diese Daten aber nicht für den Bediener an. Zum Anzeigen von Trenddaten muss das Trendobjekt auf Seite 313 aus dem Werkzeugfenster "WinStudio Screen Tools" in ein WinStudio-Bild eingefügt und konfiguriert werden.

Zum Erstellen eines neuen Trendarbeitsblatts öffnen Sie das Kontextmenü des Ordners "Tasks" und wählen aus dem Menü "Hinzufügen" den Eintrag "Trend."

Zum Bearbeiten eines bestehenden Trendarbeitsblatts führen sie im Projektexplorer auf ein Trendarbeitsblatt einen Doppelklick aus.

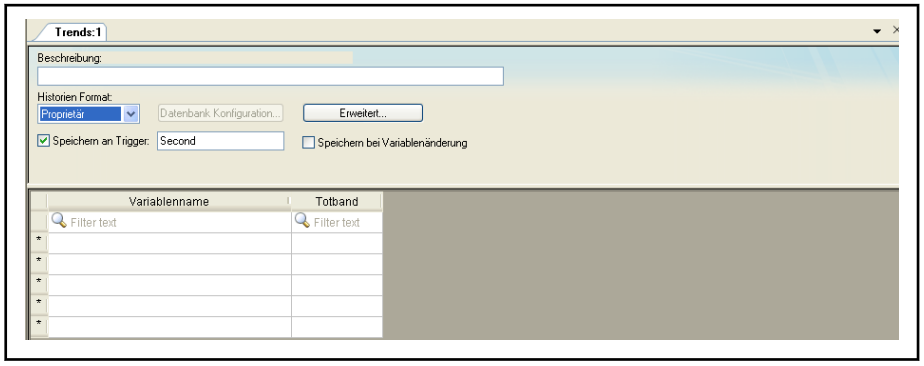


Abb.10-2: Trendarbeitsblatt

Das Trendarbeitsblatt gliedert sich in zwei Bereiche:

- Titelbereich (oberer Teil), der Informationen für die gesamte Gruppe enthält
- Hauptbereich (unterer Teil), in dem jeder Tag in der Gruppe definiert wird. Dieser Bereich enthält mehrere Spalten (in der obigen Abbildung sind nur zwei dargestellt).

Die Titel-Parameter auf diesem Arbeitsblatt werden wie folgt verwendet:

- Feld **Beschreibung**: Geben Sie zu Dokumentationszwecken eine Beschreibung für das Arbeitsblatt ein.
- **Historienformat**: Klicken Sie zum Auswählen eines Trendhistorienformats. Die folgenden Optionen sind verfügbar:
  - **Proprietär**
    - **Dateiformat:** Binär
    - **Standardpfad: …\projekt_name\Hst\GGJJTTMM.HST**, wobei:
      - **JJ** = die letzten beiden Stellen der Jahreszahl
      - **MM** = Monat
      - **TT** = Tag

☞ "WinStudio" stellt die Programme **HST2TXT.EXE** und **TXT2HST.EXE** zur Verfügung, mit denen Trendhistoriendateien von binär (**.hst**) in Text (**.txt**) konvertiert werden können und umgekehrt. Weitere Informationen über diese Programme unter Konvertieren von Trendhistoriendateien von binär in Text auf Seite 303 und Konvertieren von Trendhistoriendateien von Text in binär auf Seite 304.

- **Datenbank**
  - **Datenbanktyp:** Vom Benutzer gewählt
  - **Standard-Tabellenname:TRENDGGG** (**GGG** = Trendarbeitsblattnummer; z.B., **TREND001** für das Trendarbeitsblatt 001)

  Weitere Informationen zur Struktur der Datenbanktabelle, die "WinStudio" zum Speichern von Historiendateien verwendet, unter Datenbankschnittstelle auf Seite 493.

HINWEIS

In vielen Feldern des Trendarbeitsblatts können String-Tags zum Ändern dieser Werte während der Laufzeit festgelegt werden. Allerdings kann dies beeinflussen, wie diese Werte in der Trendhistorie gespeichert werden:

- Wenn das Historienformat auf **Proprietär** eingestellt ist, wird der Wert des String-Tags in einen numerischen Wert konvertiert (falls möglich) und dann in der Historiendatei gespeichert. Wenn die numerische Konvertierung nicht möglich ist, wird ein Nullwert gespeichert.
- Wenn das Historienformat auf **Datenbank** eingestellt ist, wird der tatsächliche Wert des String-Tags in der Datenbank gespeichert.

- **Datenbankkonfiguration**: Öffnet den Dialog Datenbankkonfiguration auf Seite 502, in dem die notwendigen Einstellungen zum Verknüpfen des Projekts mit einer externen relationalen SQL-Datenbank für das Speichern der Trendhistorie eingegeben werden können.
- Kontrollkästchen und Feld **Trigger für Speichern**: Dieses Kontrollkästchen anklicken (aktivieren) und einen Tag-Namen eingeben, um Trendbeispielwerte zu speichern, wenn der angegebene Tag geändert wird. (Die Tag-Änderung kann ein Ereignis aus dem Scheduler sein.)
- Kontrollkästchen **Speichern bei Variablenänderung**: Durch Anklicken (Aktivieren) wird der Trendbeispielwert immer gespeichert, wenn in einem der Tags aus dieser Gruppe eine Wertänderung stattfindet.
- **Erweitert**: Durch Klicken wird der Dialog "Erweiterte Einstellungen" angezeigt. Informationen zum Ausfüllen der Felder in diesem Fenster unter Stapelkonfiguration Historie auf Seite 305.

Die Haupt-Parameter auf diesem Arbeitsblatt werden wie folgt verwendet:

- Feld **Variablenname**: Geben Sie einen Variablennamen zum Speichern in der Historiendatei ein.
- Kontrollkästchen **Totband**: Geben Sie einen Wert zum Filtern annehmbarer Änderungen bei Verwendung von **Speichern bei Variablenänderung** ein.

  Zum Beispiel:

  Das Totband hat einen Wert = 5. Wenn der Tag-Wert 50 ist und sich auf 52 ändert, registriert das System diese Änderung nicht in der Datenbank, weil sie kleiner als 5 ist. Wenn die Änderung größer oder gleich 5 ist, wird der neue Wert in der Historiendatei gespeichert.
- **Feld**: Name des Felds in der Datenbank, in der der Tag gespeichert wird. Wenn das Feld leer bleibt, wird der Name des Tags als Tag-Name verwendet. Bei Array-Tags und Klassen werden die Zeichen "[", "] " und " . " durch " _ " ersetzt.

Beispiele:

Tag-Name	Standardfeld
MeinArray[1]	MeinArray_1
MeineKlasse.Member1	MeineKlasse_Member1
MeineKlasse[3].Member2	MeineKlasse_3_Member2



Der Trend-Task kann nur maximal 240 Tags in einem einzigen Arbeitsblatt aufnehmen. Wenn manuell mehr als 240 Tags im gleichen Arbeitsblatt konfiguriert werden, erzeugt der Trend-Task einen Fehler beim Ausführen des fertigen Projekts.

Siehe auch:

- Konvertieren von Trendhistoriendateien von binär in Text auf Seite 303
- Konvertieren von Trendhistoriendateien von Text in binär auf Seite 304
- Stapelerstellung Historie auf Seite 305
- Konfigurieren einer Standarddatenbank für die gesamte Task-Historie auf Seite 508

10.4 Rezeptur-Arbeitsblatt

Ein Rezeptur-Arbeitsblatt dient zum Laden von Tag-Werten oder zum Speichern von Tag-Werten in einer externen Datei während der Laufzeit. Es wird normalerweise zum Ausführen von Prozessrezepturen verwendet, es können aber beliebige Arten von Informationen in der externen Datei gespeichert werden (z.B. Betriebsprotokolle, Passwörter usw.).

Zum Erstellen eines neuen Rezeptur-Arbeitsblatts öffnen Sie das Kontextmenü des Ordners "Tasks" und wählen den Eintrag **Hinzufügen...** ▸ **Rezept**

Zum Bearbeiten eines bestehenden Rezepturen-Arbeitsblatts führen Sie im Projekt-Explorer darauf einen Doppelklick aus.

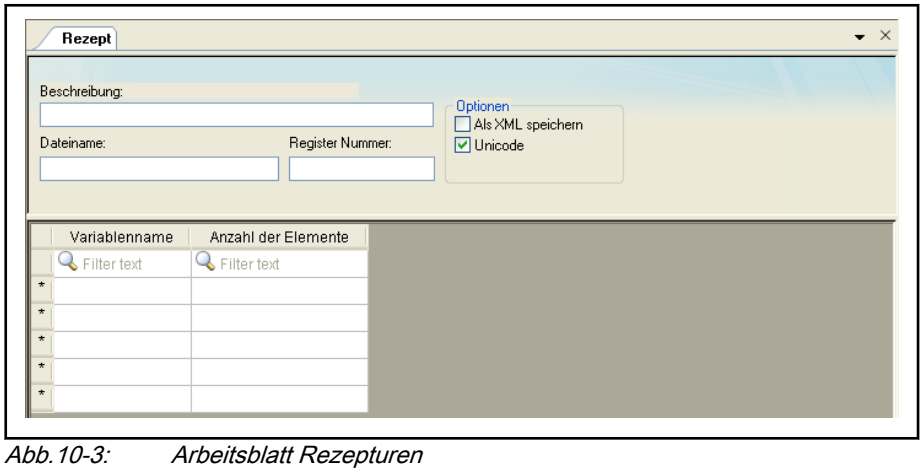


Abb. 10-3: Arbeitsblatt Rezepturen

Das Arbeitsblatt gliedert sich in zwei Bereiche:

- "Titelbereich" (oberer Teil), der Informationen für die gesamte Gruppe enthält
- "Hauptbereich" (unterer Teil), in dem jeder Tag in der Gruppe definiert wird.

Die "Titel"-Parameter auf diesem Arbeitsblatt werden wie folgt verwendet:

- **Beschreibung**: Geben Sie zu Dokumentationszwecken eine Beschreibung für das Arbeitsblatt ein.
- **Dateiname**: Geben Sie den Namen der externen Datei als statischen Text (**Datei1**) oder als indirekten Tag (**{DateiNameTag}**) ein.
- **Registernummer**: Geben Sie einen Tag zur Definition der Registernummer ein, die aus einer DBF-Datei gelesen bzw. dorthin geschrieben wird. (Dieses Feld ist nur für Altsysteme bestimmt und wird nicht mehr verwendet.)
- **Als XML speichern**: Anklicken (Aktivieren) zum Speichern von Informationen im XML-Format oder Abwählen (Deaktivieren) zum Speichern im DAT-Format. Mithilfe eines zweiten Rezepturen-Arbeitsblatts können Sie Informationen aus einer **.DAT**-Datei in verschiedene Tags laden. Informationen in einer **.XML**-Datei müssen dagegen in Tags geladen werden, die den gleichen Namen wie der Tag mit den Ursprungsdaten tragen.

☞ Wie bei HTML-Seiten muss der Web-Server ausgeführt werden, um XML-Daten aus dem Web anzuzeigen. Anders als bei HTML-Seiten im Laufzeitsystem muss bei XML-Seiten zur Anzeige der XML-Daten das Projekt nicht ausgeführt werden. (Zur Anzeige von XML-Daten muss der Internet Explorer 5.0 oder höher ausgeführt werden.)

- **Unicode**: Anklicken (Aktivieren) zum Speichern der Rezeptur im Unicode-Format (zwei Bytes pro Zeichen) oder Abwählen (Deaktivieren) zum Speichern der Rezeptur im ANSI-Format (ein Byte pro Zeichen).

☞ Ein Arbeitsblatt kann unser einem beliebigen Namen gespeichert werden (es muss kein vorbestimmter Dateiname verwendet werden). Eine Konfigurationsdatei mit der Standarderweiterung **.RCP** (oder **.XSL** wenn **Speichern als XML** gewählt wurde) enthält die Rezepturkonfiguration und das Feld **Dateiname** enthält den Namen der Datendatei, aus der gelesen bzw. in die geschrieben wird.

Die "Haupt"-Parameter auf diesem Arbeitsblatt werden wie folgt verwendet:

- **Tag-Name**: Eingabe von Tag-Namen zum Aktualisieren mit Dateiinhalten oder mit in eine Datei zu schreibenden Werten. Wenn der Tag ein Array ist, muss die erste zu verwendende Position angegeben werden. Wenn der Tag ein Array oder eine Klasse (oder beides) ist, werden standardmäßig automatisch alle Array-Positionen und Klassenelemente aktiviert. Zum Konfigurieren von bestimmten Array-Positionen und/oder Klassenelementen geben Sie diese wie gewohnt in das Feld **Tag-Name** ein. Zum Beispiel **ebene[3].member**.
- Feld **Anzahl der Elemente**: Festlegung, wie viele Positionen des Array-Tags verwendet werden.



In diesem Feld kann ein Tag-Name in geschweiften Klammern konfiguriert werden (**{tagname}**), damit der Benutzer während der Laufzeit die Anzahl der für jeden Array-Tag in der Rezeptur konfigurierten Elemente dynamisch verändern kann.

Zum Ausführen eines Rezepturen-Arbeitsblatts kann die Funktion Recipe auf Seite 634 an jeder zulässigen Stelle verwendet werden.

10.5 Report-Arbeitsblatt

Ein Report-Arbeitsblatt dient der Gestaltung eines Berichts, der während der Laufzeit dynamisch erzeugt (mithilfe der aktuellen Werte der eingeschlossenen Tags) und dann entweder an einen Drucker gesendet oder in einer Datei gespeichert wird.

Zum Erstellen eines neuen Report-Arbeitsblatts öffnen Sie das Kontextmenü des Ordners "Tasks" und wählen den Eintrag **Hinzufügen...** ▸ **Report**

Zum Bearbeiten eines bestehenden Report-Arbeitsblatts führen Sie im Projekt-Explorer darauf einen Doppelklick aus.

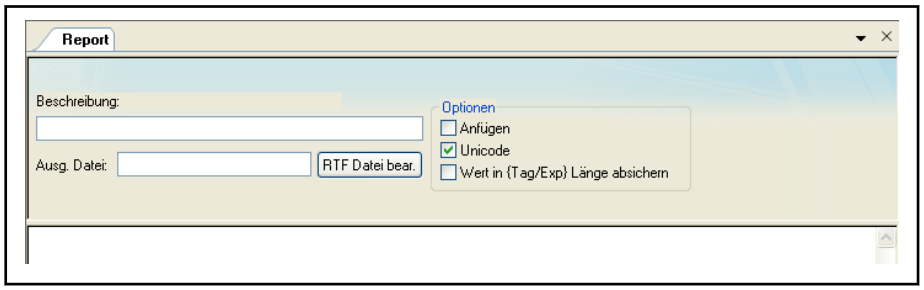


Abb. 10-4: Report-Arbeitsblatt

Das Arbeitsblatt "Report" gliedert sich in zwei Bereiche:

- "Titelbereich" (oberer Teil), der Informationen für die gesamte Gruppe enthält, und
- "Hauptbereich" (unterer Teil), in dem jeder Tag in der Gruppe definiert wird.

Die Titel-Parameter auf diesem Arbeitsblatt werden wie folgt verwendet:

- **Beschreibung**: Geben Sie zu Dokumentationszwecken eine Beschreibung für das Arbeitsblatt ein.
- **Ausgabedatei**: Geben Sie einen Tag-Namen für die Ausgabedatei ein (mit der Syntax **{tag}**), in der die Daten bei Druckausgabe in eine Datei gespeichert werden. Wobei der Tag-Wert Teil des Dateinamens ist.

  Zum Beispiel: **bericht{Tag}.out**. Wobei die erzeugte Datei je nach dem Tageswert des Tags **bericht1.out**, **bericht2.out**, **bericht3.out** usw. wäre.

☞ Eine Berichtskonfigurationsdatei verwendet ".RCP" als Standarderweiterung. Die Daten werden im Feld **Ausgabedatei** gespeichert.

- Schaltfläche **RTF-Datei bearbeiten**: Anklicken, um den Bericht als RTF-Datei zum Bearbeiten des Layouts usw. zu öffnen.
- Kontrollkästchen **An Datei anfügen**: Beim Drucken in eine Datei
  - Kontrollkästchen aktivieren, um den neuen Report am Ende einer bestehenden Datei anzuhängen (zu ändern)
  - Kontrollkästchen deaktivieren, um den bestehenden Report in dieser Datei durch den neuen Report zu ersetzen
- Kontrollkästchen **Unicode**: Anklicken (Aktivieren) zum Speichern des Berichts im Unicode-Format (zwei Bytes pro Zeichen) oder Abwählen (Deaktivieren) zum Speichern des Berichts im ASCII-Format (ein Byte pro Zeichen).

- Kontrollkästchen **Wert auf Länge von {Tag/Ausdr} beschränken**: Anklicken (Aktivieren) zum automatischen Beschränken der Werte von Tags/ Ausdrücken im Report auf den Bereich zwischen den geschweiften Klammern, so wie diese im Hauptteil des Reports positioniert sind (siehe unten). So kann das Layout des Reports erhalten werden. Wenn diese Option nicht angewählt wird, werden die vollständigen Werte von Tags/Ausdrücken im Report angezeigt.

Verwenden Sie den "Hauptteil" dieses Arbeitsblatts zum Formatieren des Reports. Ein Report kann unter Verwendung der Daten im System und unter Angabe des Zielorts zum Drucken der Tag-Werte konfiguriert werden. Jeder Tag-Name ersetzt den Tag-Namen "{"tag_name"}". Für Tags vom Typ Real wird die folgende Syntax verwendet: **{tag_name n}**, wobei **n** die Anzahl der zu druckenden Dezimalstellen ist.



Bei Verwendung des Standard-Berichts-Editors (nur Text: ASCII oder Unicode) ist die Anzahl der für den Tag-Wert reservierten Zeichen gleich der Anzahl der Zeichen zur Eingabe des Tag-Namens (einschließlich der zwei geschweiften Klammern). Wenn zum Beispiel **{TagA}** im Hauptteil des Berichts konfiguriert wird, müssen für den Tag-Wert in der Berichtsdatei sechs Zeichen reserviert werden. Diese Funktionalität ist für Berichte im RTF-Format nicht gültig.

Zum Ausführen eines Report-Arbeitsblatts kann die Funktion Report auf Seite 635 an jeder zulässigen Stelle verwendet werden.

10.6 ODBC-Arbeitsblatt

Die ODBC-Schnittstelle läuft in einer Netzwerkumgebung und verwendet die Windows-ODBC-Standardkonfiguration. Der "ODBC"-Task ist zum Datenaustausch zwischen "WinStudio" und allen Datenbanken, die diese Schnittstelle unterstützen, fähig.



Zusätzlich zur Konfiguration des "ODBC"-Arbeitsblatts muss der Windows-ODBC-Standardtreiber konfiguriert werden. "WinStudio" bezieht sich auf den Benutzer-DSN (Data Source Name = Datenquellenname), der über die Systemsteuerung konfiguriert wird. Mehr Informationen finden Sie in Ihrer Windows-Dokumentation.

Außerdem ist die ODBC-Schnittstelle nicht für Projekte verfügbar, die auf Windows-Embedded-Zielsystemen laufen.

Zum Erstellen eines neuen ODBC-Arbeitsblatts öffnen Sie das Kontextmenü des Ordners "Tasks" und wählen den Eintrag **Hinzufügen... ▸ ODBC-Arbeitsblatts**

Zum Bearbeiten eines bestehenden ODBC-Arbeitsblatts führen sie im Projekt-Explorer darauf einen Doppelklick aus.

Ein neues Arbeitsblatt wird wie folgt angezeigt.

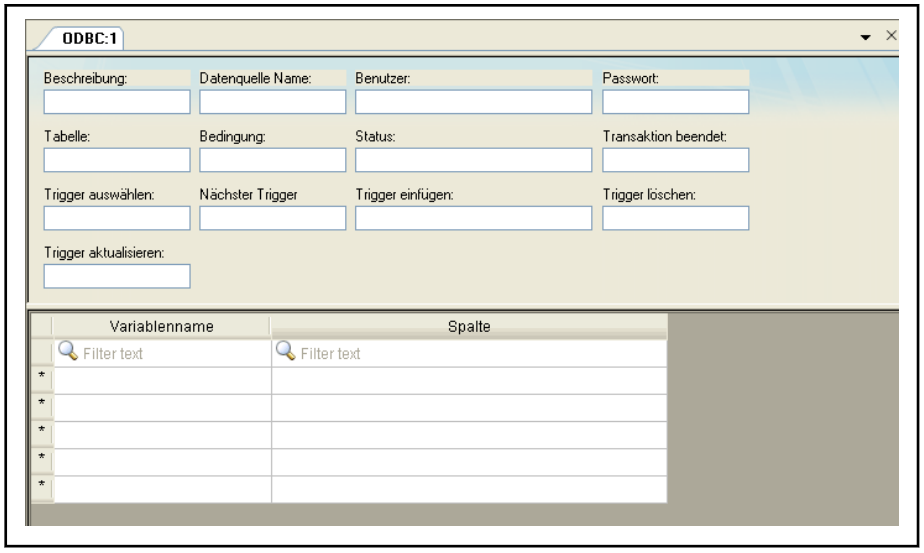


Abb.10-5: ODBC Arbeitsblatt

ODBC-Arbeitsblätter werden unter dem Task "ODBC-Laufzeit" ausgeführt. Das Erstellen eines neuen Arbeitsblatts aktiviert diesen Task allerdings nicht automatisch. Der Task muss in der Taskkonfiguration (im Kontextmenü WinStudio unter **Werkzeuge ▸ Status**) zum Start während der Laufzeit aktiviert werden. Weitere Informationen finden Sie unter <Task-Ausführung>.

Das Arbeitsblatt "ODBC" gliedert sich in zwei Bereiche:

- "Titelbereich" (oberer Teil), der Informationen für die gesamte Gruppe enthält, Tags zum Starten von Lese- und Schreibereignissen definiert, Rückgabewerte festlegt, Datenbank-Zugangsparameter handhabt usw., und
- "Hauptbereich" (unterer Teil), in dem jeder Tag in der Gruppe definiert wird und Tags mit Feldern im aktuellen Register der Datenbanktabelle verbunden werden.

Die "Titel"-Parameter auf diesem Arbeitsblatt werden wie folgt verwendet:

- **Beschreibung**: Geben Sie zu Dokumentationszwecken eine Beschreibung für das Arbeitsblatt ein.
- **Datenquellenname**: Geben Sie den in der Windows-Systemsteuerung festgelegten Datenquellennamen (DNS) mit Informationen zum spezifischem Datenbankzugriff ein.
- **Benutzer**: Geben Sie einen Benutzernamen für den Datenbankzugriff ein.
- **Passwort**: Geben Sie das Benutzerpasswort ein.
- **Tabelle**: Geben Sie einen Tabellennamen in der Datenbank ein.
- **Bedingung**: Geben Sie eine Suchbedingung oder einen Filter ein.
- **Status**: Geben Sie einen Tag ein, in den Statuswert bei ODBC-Zugriffen geschrieben werden soll. Der Tag sollte 0 für erfolgreich ausgeben und einen anderen Wert für einen Fehlercode verwenden.
- **Transaktion beendet**: Geben Sie einen Tag ein, der bei Ausführung der Transaktion seinen Wert ändert.
- **Trigger für Auswählen**, **Nächster**, **Einfügen**, **Löschen** oder **Aktualisieren**: Geben Sie einen Tag als Trigger ein, wobei WinStudio bei jeder Wertänderung den entsprechenden Befehl ausführt. Es wird mindestens ein Triggerfeld benötigt.

Die "Haupt"-Parameter auf diesem Arbeitsblatt werden wie folgt verwendet:

- **Variablenname**: Geben Sie die Tag-Namen zum Aktualisieren mit Dateiinhalten oder Tags ein, deren Werte in eine Datei geschrieben werden sollen.
- **Spalte**: Geben Sie den Ort ein, an dem Daten in der Datei gefunden werden können (z.B. "R3CH" entspricht Zeile 3, Spalte H in einem Excel-Arbeitsblatt)

Zum Einrichten der ODBC-Schnittstelle für Excel-Dateien muss die Windows-Systemsteuerung verwendet werden. Die Vorgehensweise ist wie folgt:

1. Klicken Sie auf **Start ▸ Einstellungen ▸ Systemsteuerung**.
2. Wenn das Fenster "Systemsteuerung" angezeigt wird, führen Sie zum Öffnen des Dialogs "ODBC Datenquellenverwaltung" einen Doppelklick auf das ODBC-Symbol aus.
3. Im Dialog "ODBC Datenquellenverwaltung" wählen Sie aus der Liste Benutzerdatenquellen den Eintrag Excel-Dateien aus und klicken Sie dann auf die Schaltfläche Konfigurieren.
4. Wenn der Dialog "Einrichten ODBC Microsoft Excel" angezeigt wird, geben Sie den im DSN-Feld im "ODBC"-Arbeitsblatt zu verwendenden Windows-Konfigurationsnamen in das Feld Datenquellename ein.
5. Klicken Sie auf die Schaltfläche Arbeitsmappe auswählen, um die zu verwendende Excel-Datei zu konfigurieren.
6. Kehren Sie zum Dialog ODBC Datenquellenverwaltung zurück und stellen Sie sicher, dass Ihr Benutzer-DSN in der Liste angezeigt wird. Schließen Sie den Dialog durch Klicken auf OK.
7. Nach dem Konfigurieren der ODBC-Windows-Schnittstelle müssen die "ODBC"-Arbeitsblätter des Projekts konfiguriert werden.
8. Fügen Sie aus der Registerkarte "Tasks" ein neues ODBC-Arbeitsblatt ein.
9. Achten Sie unbedingt darauf, in der Task-Konfiguration (Menüeintrag **Werkzeuge ▸ Status** im Kontextmenü von WinStudio) das automatische Starten der ODBC-Laufzeit festzulegen.

Zum Starten dieser Konfiguration wird einfach das Projekt ausgeführt. Ihr Projekt handhabt die Trigger **Auswählen**, **Nächste**, **Einfügen**, **Löschen** und **Aktualisieren**, um den Datenaustausch in Excel-Zeilen und im Arbeitsblatt konfigurierten Tags zu ermöglichen.

Die Bedeutung spezifischer Fehlermeldungen finden Sie in Ihrer Windows-Dokumentation.

Im folgenden finden Sie eine Liste von "WinStudio"-Fehlermeldungen:

- *Befehl Auswählen*
  - 1 - Fehler in der Funktion ODBCPREPARE
  - 2 - Fehler in der Funktion ODBCBINDCOL
  - 3 - Fehler in der Funktion ODBCEXECUTE
  - 4 - Fehler in der Funktion ODBCSETCH
- *Befehl Nächste*
  - 5 - Fehler in der Funktion ODBCSETCH
- *Befehl Einfügen*
  - 6 - Fehler in der Funktion ODBCPREPARE
  - 7 - Fehler in der Funktion ODBCEXECUTE
  - 8 - Fehler in der Funktion ODBCCOMMITE

- *Befehl Aktualisieren*
  - 9 - Fehler in der Funktion ODBCPREPARE
  - 10 - Fehler in der Funktion ODBCEXECUTE
  - 11 - Fehler in der Funktion ODBCCOMMITE
- *Befehl Löschen*
  - 12 - Fehler in der Funktion ODBCPREPARE
  - 13 - Fehler in der Funktion ODBCEXECUTE
  - 14 - Fehler in der Funktion ODBCCOMMITE

10.7 Mathematik-Arbeitsblatt

Ein Mathematik-Arbeitsblatt dient mehr der Implementierung von Programmlogik (mithilfe der internen Skriptsprache), die während der Laufzeit kontinuierlich ausgeführt werden soll, als von spezifischen Aktionen wie dem Drücken einer Schaltfläche auf einem Bild durch den Benutzer.

Das Mathematik-Arbeitsblatt gleicht in seinen Funktionen dem Arbeitsblatt Skript auf Seite 368, außer dass es anstelle von VBScript die interne Skriptsprache verwendet.

Zum Erstellen eines neuen Mathematik-Arbeitsblatts öffnen Sie das Kontextmenü des Ordners "Tasks" und wählen den Eintrag **Hinzufügen...** ▸ **Mathematikblatt**

Zum Bearbeiten eines bestehenden Mathematik-Arbeitsblatts führen Sie im Projekt-Explorer darauf einen Doppelklick aus.

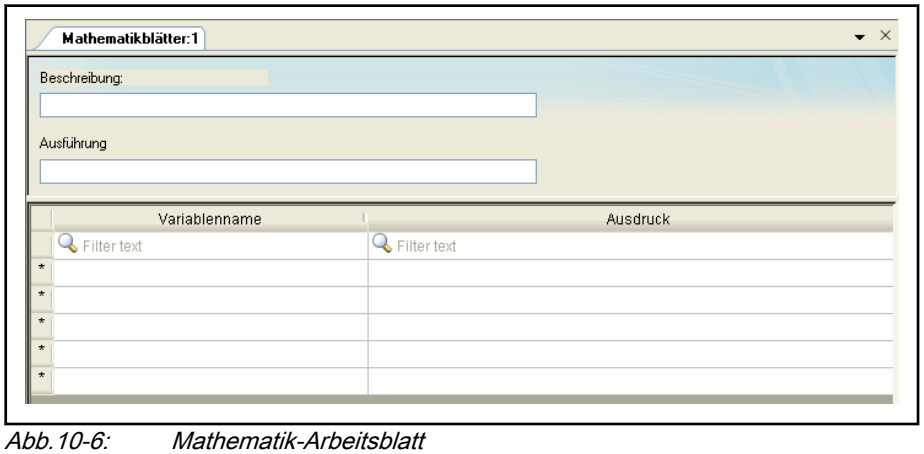


Abb. 10-6: Mathematik-Arbeitsblatt

Das Mathematik-Arbeitsblatt gliedert sich in zwei Bereiche:

- "Titelbereich" (oberer Teil), der Informationen für die gesamte Gruppe enthält, und
- "Hauptbereich" (unterer Teil), in dem alle Tags, Ausdrücke und Programmzeilen (logische Routinen und mathematische Berechnungen durch Funktionen und logische Operationen) in der Gruppe definiert werden.

Die "Titel"-Parameter auf diesem Arbeitsblatt werden wie folgt verwendet:

- **Beschreibung**: Geben Sie zu Dokumentationszwecken eine Beschreibung für das Arbeitsblatt ein.
- **Ausführung**: Geben Sie einen Ausdruck, einen einzelnen Tag oder einen konstanten Wert ein, um festzulegen, wann das Arbeitsblatt ausgeführt werden soll.



Das Projekt führt das Arbeitsblatt nur dann aus, wenn das Ergebnis im Feld **Ausführung** "nicht null" ist. Wenn das Arbeitsblatt immer ausgeführt werden soll, geben Sie im Feld **Ausführung** eine 1 (konstanter Wert) ein.

Die "Haupt"-Parameter auf diesem Arbeitsblatt werden wie folgt verwendet:

- Feld **Variablenname**: Geben Sie einen Tag ein, um einen Rückgabewert von der angegebene Berechnung in der Spalte **Ausdruck**zu erhalten.
- Feld **Ausdruck**: Geben Sie einen Ausdruck ein, um einen Rückgabewert an den angegebenen Tag in der Spalte **Variablenname** zurückzugeben.

Zur Ausführung eines Mathematik-Arbeitsblatts zu einem bestimmten Zeitpunkt unabhängig von der im Feld **Ausführung** konfigurierten Einstellung, kann die Funktion Math auf Seite 633 überall dort ausgeführt werden, wo ein Ausdruck erlaubt ist.

10.8 Skript-Arbeitsblatt

10.8.1 Skript-Arbeitsblatt, Allgemeines

Ein Skript-Arbeitsblatt dient mehr der Implementierung von Programmlogik (mithilfe von VBScript), die während der Laufzeit kontinuierlich ausgeführt werden soll, als von spezifischen Aktionen wie dem Drücken einer Schaltfläche auf einem Bild durch den Benutzer.



Das Skript-Arbeitsblatt gleicht in seinen Funktionen dem Arbeitsblatt Berechnung auf Seite 367, außer dass es anstelle der internen Skriptsprache VBScript verwendet.

Zum Erstellen eines neuen Skript-Arbeitsblatts öffnen Sie das Kontextmenü des Ordners "Skripte" und wählen den Eintrag **Neues Skript**

Zum Bearbeiten eines bestehenden Skript-Arbeitsblatts führen Sie im Projekt-Explorer darauf einen Doppelklick aus.

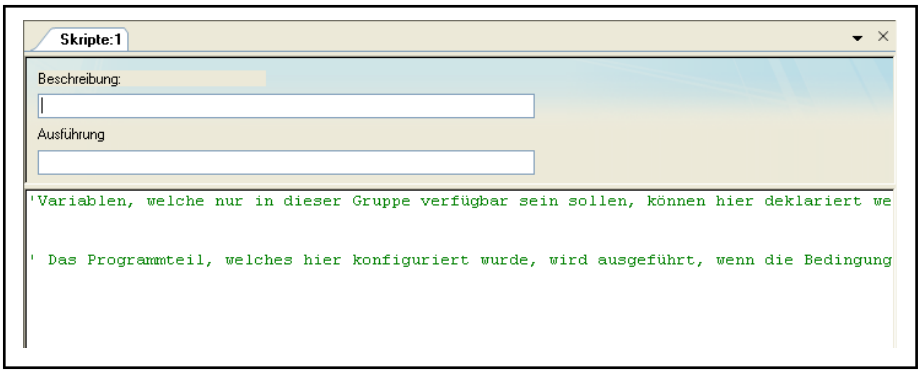


Abb.10-7: Skript-Arbeitsblatt

Der in jedem Skript-Arbeitsblatt konfigurierte Code wird vom Hintergrund-Task ausgeführt. Das Projekt überprüft die Arbeitsblätter sequenziell (aufgrund der Arbeitsblattnummer) und führt nur die Gruppen aus, bei denen die im Feld **Ausführung** des Arbeitsblatts konfigurierte Bedingung WAHR ist (d.h. nicht null).



Es muss die von der internen Skriptsprache im Feld **Ausführung** unterstützte Syntax verwendet werden. Nur der Hauptteil des Arbeitsblatts unterstützt VBScript.

Im Arbeitsblatt deklarierte Variablen haben einen lokalen Gültigkeitsbereich nur für diese bestimmte Gruppe. Sie stehen für keine anderen VBScript-Schnittstellen zur Verfügung.

Im Skript-Arbeitsblatt können keine Prozeduren (d.h. Funktionen und Unterprogramme) definiert werden. Es können aber unter Globale Prozeduren auf Seite 799 oder im Startup-Skript auf Seite 369 definierte Prozeduren aufgerufen werden.

Beispiel:

```
'Nur für diese Gruppe verfügbare Variablen können hier deklariert werden
Dim myVar, myTest
myTest = 1

'Der hier konfigurierte Code wird ausgeführt,
'während die im Feld Ausführung konfigurierte Bedingung WAHR ist
myVar = $FindFile("c:\*.txt")
If MyVar > 0 Then
  $TagNumOfFiles = myVar
End If
```

HINWEIS

Wenn ein Skript-Arbeitsblatt während der Laufzeit gespeichert wird (Online-Konfiguration), wird das Startup-Skript erneut ausgeführt

Der aktuelle Wert der lokalen Variablen aller Skript-Arbeitsblätter wird zurückgesetzt.

10.8.2 Skript-Arbeitsblatt Startup

Das Skript-Arbeitsblatt Startup ist eine VBScript-Schnittstelle, die bei Ausführung des Projekts automatisch ausgeführt wird.

Zum Bearbeiten des Skript-Arbeitsblatts Startup führen Sie im Projekt-Explorer darauf einen Doppelklick aus. (Es befindet sich im Ordner "WinStudio \Tasks\Skripte")

Das Arbeitsblatt wird angezeigt:

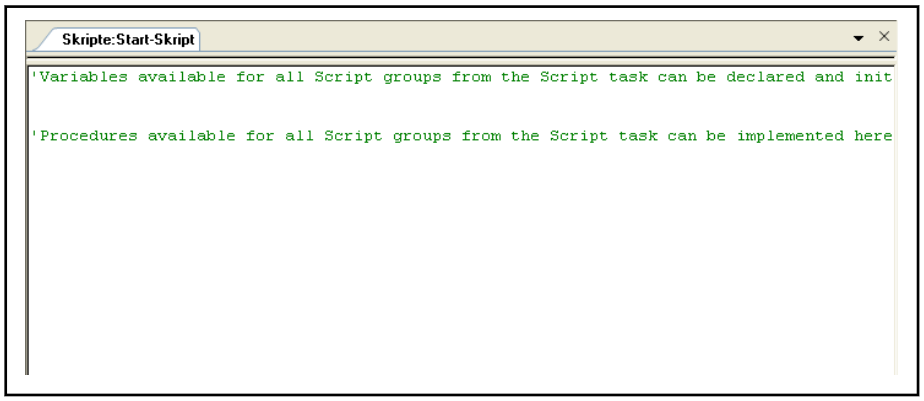


Abb.10-8: Skript-Arbeitsblatt Startup

Der in diesem Arbeitsblatt konfigurierte Code wird nur einmal ausgeführt, wenn das Modul Hintergrund-Task (BG_Task) ausgeführt wird. Diese Schnittstelle ist hilfreich zur Initialisierung von Variablen oder zum Ausführen von Logiken, die bei Ausführung des Projekts implementiert werden müssen.

Sie können Variablen deklarieren und initialisieren und Prozeduren definieren. In dieser Schnittstelle deklarierte Variablen oder Prozeduren sind allerdings NUR in den vom Hintergrund-Task-Modul ausgeführten Skript-Arbeitsblättern verfügbar – für VBScript-Schnittstellen aus dem Grafikmodul sind sie nicht verfügbar.

Hintergrundtasks

Beispiel:

```
'Hier können für alle Skriptgruppen aus dem Skript-Task verfügbaren Variablen deklariert und initialisiert werden
Dim MyVar, Counter
MyVar = 100

'Hier können für alle Skriptgruppen aus dem Skript-Task verfügbaren Prozeduren implementiert werden

Function AreaEquTriangle(base, high)
  AreaEquTriangle = (base * high) / 2
End Function

Sub CheckLimits(myValue, myHiLimit, myLoLimit)
  If (myValue > myHiLimit Or myValue < myLoLimit) Then
    MsgBox("Wert außerhalb des Bereichs")
  End If
End Sub

'Der hier konfigurierte Code wird nur einmal beim Starten des Hintergrundtasks ausgeführt
If $GetOS() = 3 Then
  MsgBox ("Willkommen! Dieses Projekt wird unter dem Betriebsyystem Microsoft Windows Embedded ausgeführt.")
Else
  MsgBox ("Willkommen! Dieses Projekt wird unter dem Betriebsyystem Microsoft Windows Desktop ausgeführt.")
End If
```

10.9 Scheduler-Arbeitsblatt

Ein Scheduler-Arbeitsblatt dient zur Ausführung von Programmlogik (mithilfe der internen Skriptsprache) zu einem bestimmen Datum/Zeitpunkt, in regelmäßigen Zeitabständen oder beim Auslösen eines Ereignisses.

Zum Erstellen eines neuen Scheduler-Arbeitsblatts öffnen Sie das Kontextmenü des Ordners "Tasks" und wählen den Eintrag **Hinzufügen...** ▸ **Scheduler**

Zum Bearbeiten eines bestehenden Scheduler-Arbeitsblatts führen Sie im Projekt-Explorer darauf einen Doppelklick aus.

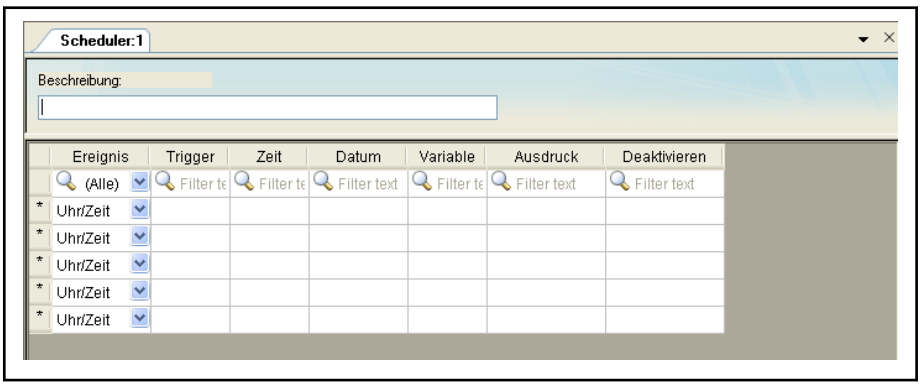


Abb. 10-9: Scheduler-Arbeitsblatt

Das Arbeitsblatt "Scheduler" gliedert sich in zwei Bereiche:

- **Titelbereich** (oberer Teil), der Informationen für die gesamte Gruppe enthält
- **Hauptbereich** (unterer Teil), in dem alle Tags, Ausdrücke und Bedingungen in der Gruppe definiert werden.

Die Parameter auf diesem Arbeitsblatt werden wie folgt verwendet:

- **Beschreibung**: Geben Sie zu Dokumentationszwecken eine Beschreibung für das Arbeitsblatt ein.
- **Ereignis**

  Auswahlliste: Klicken zum Auswählen eines Ereignistyps aus den folgenden Optionen:

– **Kalender**: Erzeugt Zeitspannen größer als 24 Stunden. Zum Beispiel kann ein Ereignis definiert werden, das jeden Freitag zu einer bestimmten Uhrzeit einen Bericht druckt.



Das Feld **Datum** muss ausgefüllt sein, wenn ein bestimmtes Datum für die Ausführung eines Ereignisses gewünscht ist.

– **Zeit**: Erzeugt Zeitspannen kleiner als 24 Stunden (Intervalle in Minuten oder Sekunden). Diese Funktion wird häufig für Trendgrafiken verwendet. Zum Beispiel kann ein Tag definiert werden, der jeder Stunde inkrementell erhöht wird.
– **Änderung**: Ereignis, das mit der Änderung eines Tags im Feld **Trigger** verbunden ist.



Dies funktioniert nur für Tag-Änderungen auf dem Projekt-Server, ungeachtet des für den Tag definierten Bereichs.

- **Trigger**: Dieses Feld wird nur mit dem Ereignistyp **Änderung** verwendet. Geben Sie in diesem Feld den Namen eines Projekt-Tags ein. Wenn der Wert des Tags sich ändert, wird **Ausdruck** ausgewertet.
- **Zeit**: Dieses Feld wird mit den Ereignistypen **Kalender** und **Uhr** verwendet.

  Wenn **Kalender** als Ereignistyp gewählt ist, ist **Zeit** eine bestimmte Uhrzeit am Tag **Datum**. Bei Eintreten von **Datum** und **Zeit** wird **Ausdruck** ausgewertet. Wenn **Uhr** als Ereignistyp gewählt ist, ist **Zeit** ein Zeitintervall ab Ausführung des Projekts. Bei jedem Auftreten des Intervalls wird **Ausdruck** ausgewertet.

  In beiden Fällen wird die Zeit im Format "HH.MM:SS.ms" eingegeben. Zulässige Werte sind "00" bis "23" für Stunden, "00" bis "59" für Minuten, "00" bis "59" für Sekunden und "1" bis "9" für Millisekunden. (Millisekunden sind optional.) Beispiele: "03:00:00" bedeutet alle 3 Stunden, "00:00:00.1" bedeutet alle 100 Millisekunden.
- **Datum**: Dieses Feld wird nur mit dem Ereignistyp **Kalender** verwendet. Geben Sie ein bestimmtes Datum im Format "MM/TT/YYYY" ein. Bei Auftreten von **Datum** und **Zeit** wird **Ausdruck** ausgewertet.

  Zulässige Werte sind "01" bis "12" für Monate, "01" bis "31" für Tage und "1900" bis "2099" für Jahre. Bleibt das Feld leer, tritt das Ereignis täglich zur festgelegten **Zeit** ein.
- **Variable**: Geben Sie einen Tag ein, der den ggf. vom **Ausdruck** zurückgegebenen Wert erhält.
- **Ausdruck**: Geben Sie einen Ausdruck zur Auswertung ein. Dieses Feld wird von allen Ereignissen verwendet.
- **Deaktivieren**: Enthält eine Deaktivierungsbedingung für die angegebene Funktion. Zum Ausführen der Funktion lassen Sie dieses Feld leer oder verwenden Sie einen Ausdruckswert gleich null. Bei Verwendung eines Ausdruckswerts gleich eins wird die Funktion nicht ausgeführt (deaktivieren = 1).

10.10 Datenbank-/ERP-Arbeitsblatt

Neben ODBC unterstützt "WinStudio" auch Microsoft .NET ActiveX Datenobjekte (ADO.NET) als Schnittstelle zwischen Projekt-Tag-Datenbank und an-

deren externen Datenbanken. Ein Datenbank-/ERP-Arbeitsblatt dient der Verknüpfung von Projekt-Tags mit externen Datenbankfeldern.



Weitere Informationen über ADO.NET-Unterstützung in "WinStudio" – einschließlich der Kommunikation mit Remote-Datenbanken mithilfe der "WinStudio"-Datenbank Gateway-Software – finden Sie unter Datenbankschnittstelle auf Seite 493.

Zur Kommunikation mit einer externen Datenbank müssen Sie zuerst eine Verbindung konfigurieren auf Seite 372 zur Datenbank und dann ein Arbeitsblatt erstellen auf Seite 374, das Projekt-Tags mit den Datenbankfeldern verknüpft.

Datenbankverbindungen

Zum Erstellen einer neuen Verbindung zu einer Zieldatenbank:

1. Öffnen Sie im Projekt-Explorer auf Seite 59 das Kontextmenü des Ordners **Externe Datenbanken** und wählen Sie dann den Eintrag **Datenbankverbindung konfigurieren** aus.
2. Es öffnet sich der Konfigurationsdialog "Datenbankverbindungen", in dem alle bereits konfigurierten Datenbankverbindungen angezeigt werden.

   Drücke Sie die Schaltfläche **Hinzufügen**, um eine neue Datenbankverbindung zu erstellen. Der Dialog "Datenbankverbindung" wird angezeigt.

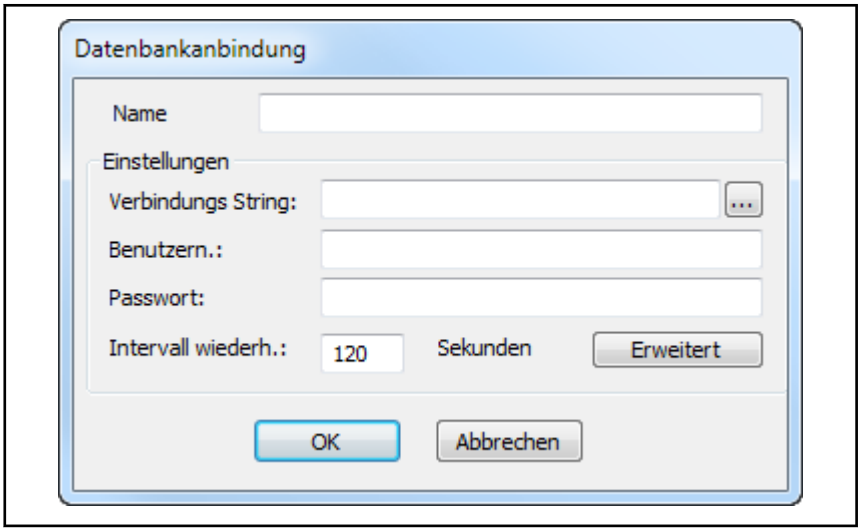


Abb. 10-10: Dialog Datenbankverbindung

3. Geben Sie im Feld **Name** den Namen ein, der als Bezug für die Datenbank verwendet werden soll. Sie können mehrfache Datenbankverbindungen erstellen, aber jede Verbindung muss einen eindeutigen Namen haben.
4. Klicken Sie im Feld **Verbindungsstring** auf die Suchschaltfläche **...** zum Öffnen eines Dialogs "Datenverbindungseigenschaften". Verwenden Sie den Dialog zum Konfigurieren eines Verbindungsstrings für die Zieldatenbank.



Die im Dialog "Datenverbindungseigenschaften" aufgeführte Liste der Datenbankanbieter ist abhängig von den tatsächlich auf der Station mit der Entwicklungsanwendung installierten und verfügbaren Anbietern. Weitere Informationen zur Verwendung der "Datenverbindungseigenschaften" finden Sie in der Windows-Hilfe.

5. Geben Sie in den Feldern **Benutzername** und **Passwort** ein gültiges Login für die Zieldatenbank ein. Das Login sollte bereits auf dem Datenbank-Server erstellt worden sein und sollte über ausreichend Berechtigungen zum Lesen und Schreiben von bzw. zu den Datenbanktabellen verfügen.
6. Wenn Sie sich über den

   Studio Database Gateway auf Seite 496 mit einer Remote-Datenbank verbinden, klicken Sie auf die Schaltfläche **Weitere Einstellungen** zum Öffnen des Dialogs Weitere Einstellungen, wie unten gezeigt.

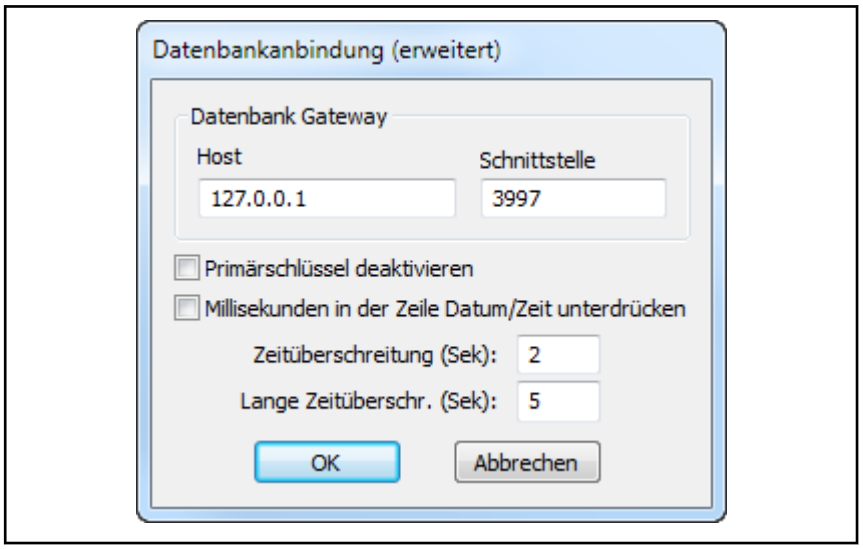


Abb.10-11: Dialog Datenbankverbindung (Weitere Einstellungen)

7. Geben Sie im Feld **Host** die IP-Adresse der Station ein, auf der die "WinStudio"-Datenbank Gateway-Software läuft (**STADOSvr.exe**). Im Feld **Port** geben Sie die Port-Nummer ein, auf der die Software laut Konfiguration läuft.

   Andere bei Bedarf zu konfigurierende Einstellungen:

   - Kontrollkästchen **Primärschlüssel deaktivieren**: "WinStudio" versucht, zum Beschleunigen der Abfragen einen Primärschlüssel für die Tabelle zu definieren. Bei Verwendung einer Datenbank ohne Primärschlüsselunterstützung (z.B. Microsoft Excel) sollte diese Option ausgewählt werden.
   - Kontrollkästchen **Millisekunden in Spalten Datum/Zeit deaktivieren**: "WinStudio" wird versuchen, beim Speichern von Datum/Zeit in der Datenbank Millisekunden einzuschließen. Bei Verwendung einer Datenbank, die keine Millisekunden unterstützt, sollte dieses Kontrollkästchen aktiviert werden.
8. Klicken Sie auf **OK**, um die Verbindungskonfiguration zu speichern und den Dialog zu schließen.

Datenbankverbindungen werden im Unterordner **\projekt_name\Config** als XML-Dateien gespeichert. Jede Datei erhält den gleichen Namen wie die Verbindung (wie im Feld **Name** im Dialog "Datenbankverbindung" angegeben) und die Dateierweiterung **.XDC**. Zum Beispiel ist die Verbindungskonfiguration "DB1" gespeichert in der Datei...

\projekt_name\Config\DB1.XDC

Hintergrundtasks

Datenbank-Arbeitsblatt

Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Befehlen vertraut machen.

Datenbank-Arbeitsblätter ermöglichen die asynchrone Ausführung von Datenbankoperationen und bieten eine benutzerfreundliche Schnittstelle für den Aufbau von SQL-Befehlen. Zum Erstellen eines neuen Datenbank-Arbeitsblatts wählen Sie im Kontextmenü von "Tasks" oder "Externe Datenbanken" den Eintrag **Neues Datenbankblatt.**

Ein neues Arbeitsblatt wird wie folgt angezeigt:

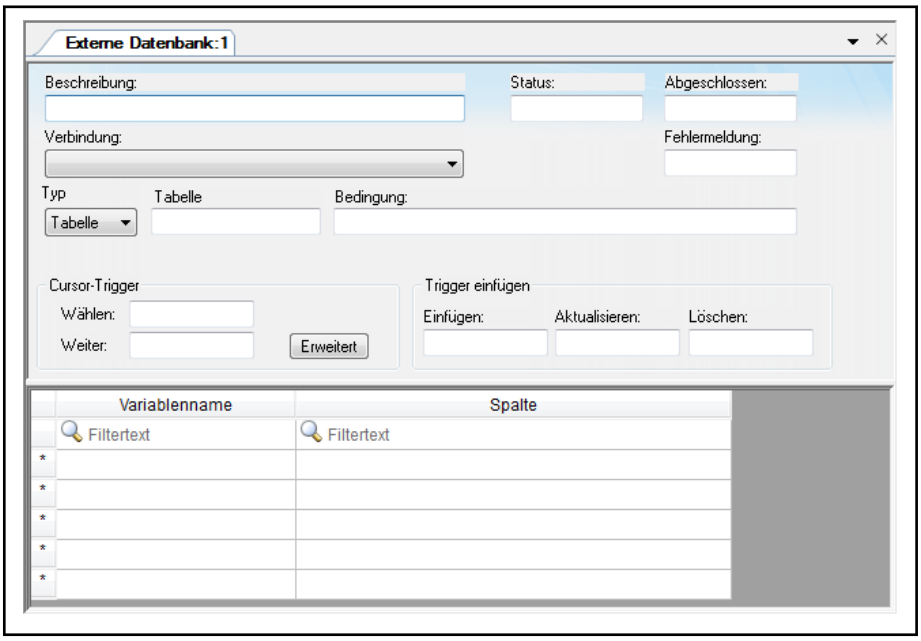


Abb. 10-12: Datenbank-Arbeitsblatt

Datenbank-Arbeitsblätter werden mit der Dateierweiterung **.XDB** im Verzeichnis **\projekt_name\Config** gespeichert. Jedes neue Arbeitsplatz wird automatisch in der Reihenfolge seiner Erstellung nummeriert. Zum Beispiel wird das zuerst erstellte Arbeitsblatt gespeichert in der Datei...

\projekt_name\Config\DB001.XDB

Datenbank-Arbeitsblätter werden unter dem Task "Datenbase/ERP-Runtime" ausgeführt. Das Erstellen eines neuen Arbeitsblatts aktiviert diesen Task allerdings nicht automatisch. Der Task muss in der Task-Konfiguration (Eintrag **Werkzeuge ▸ Status** im Kontextmenü des Ordners WinStudio) zum Start während der Laufzeit konfiguriert werden. Weitere Informationen finden Sie unter <Task-Ausführung>.

Außerdem laufen Datenbank-Arbeitsblätter nur auf dem Server und alle Trigger müssen mit Server-Tags konfiguriert werden.

Arbeitsblatt-Titel

Der Titelbereich des Datenbank-Arbeitsblatts wird wie folgt konfiguriert:

- **Beschreibung**: Geben Sie zu Dokumentationszwecken eine Beschreibung für das Arbeitsblatt ein.
- **Status**: Geben Sie einen numerischen Tag zum Empfang von Statuscodes für Datenbankoperationen während der Laufzeit ein:

Statuscodes für externe Datenbankoperationen

Statuscode	Beschreibung
4	Ergebnissatz ist leer
3	Cursor freigegeben und Abfrage erfolgreich geschlossen
2	Anfang des Ergebnissatzes erreicht, normalerweise beim Bewegen das Cursors zur vorherigen Zeile
1	Ende des Ergebnissatzes erreicht, normalerweise beim Bewegen das Cursors zur nächsten Zeile
0	Keine Fehler; Zustand normal
-1	Fehler beim Verbinden mit angegebener Datenbank (siehe **Verbindung** weiter unten)
-2	Fehler beim Auswählen des Ergebnissatzes
-3	Fehler beim Bewegen des Cursors zur nächsten Reihe (siehe Auslöser **Nächste** weiter unten)
-4	Fehler beim Bewegen des Cursors zur vorherigen Reihe (siehe Auslöser **Vorherige** weiter unten)
-5	Fehler beim Schließen der Abfrage (siehe Auslöser **Abfrage schließen** weiter unten)
-6	Fehler beim Einfügen von Zeilen im Ergebnissatz (siehe Auslöser **Einfügen** weiter unten)
-7	Fehler beim Aktualisieren des Ergebnissatzes (siehe Auslöser **Aktualisieren** weiter unten)
-8	Fehler beim Löschen des Ergebnissatzes (siehe Auslöser **Löschen** weiter unten)

- **Abgeschlossen**: Geben Sie den Namen eines numerischen Tags ein, der bei erfolgreicher Ausführung von Datenbankbefehlen umgeschaltet wird.
- **Fehlermeldung**: Geben Sie den Namen eines String-Tags ein, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.
- **Verbindung**: Klicken zum Auswählen einer Verbindung zur Zieldatenbank. Alle verfügbaren Verbindungen werden aufgeführt, konfiguriert wie im Dialog "Datenbankverbindung" zuvor beschrieben auf Seite 372.
- **Typ**: Klicken zum Festlegen, wie der Ergebnissatz für das Arbeitsblatt ausgewählt wird:
  - **Tabelle**: Geben Sie einen Tabellennamen und eine optionale Filterbedingung ein. (Die Filterbedingung entspricht dem SQL-Satz „Where“.) Alle den Bedingungen entsprechenden Tabellenzeilen werden ausgewählt.
  - **SQL**: Geben Sie eine benutzerdefinierte SQL-Anweisung „Select“ ein.

☞ Für **Tabelle**, **Bedingung** und **SQL-Anweisung** können die Namen von Projekt-Tags mit den gewünschten Daten eingegeben werden. Auf diese Weise kann die Auswahl während der Laufzeit über das Programm geändert werden. Allerdings müssen Tag-Namen in geschweifte Klammern gesetzt werden ("{}"), um sie von tatsächlichen Strings zu unterscheiden. Außerdem muss vor der Freigabe einer bestehenden Auswahl eine neue geöffnet werden; siehe **Abfrage schließen** weiter unten.

- *Bereich Cursor-Trigger...*
  - **Auswählen**: Geben Sie einen beliebigen Tag ein; wenn sich der Tag-Wert ändert, öffnet ein neuer Cursor die erste Zeile des Er-

gebnissatzes und kopiert diese Werte in die im Hauptteil des Arbeitsblatts konfigurierten Tags.

– **Nächste**: Geben Sie einen beliebigen Tag ein; wenn sich der Tag-Wert ändert, bewegt sich der Cursor in die nächste Zeile des Ergebnissatzes und kopiert diese Werte in die im Hauptteil des Arbeitsblatts konfigurierten Tags.

– **Erweitert**: Klicken Sie zum Öffnen des Dialogs "Weitere Cursor-Einstellungen":

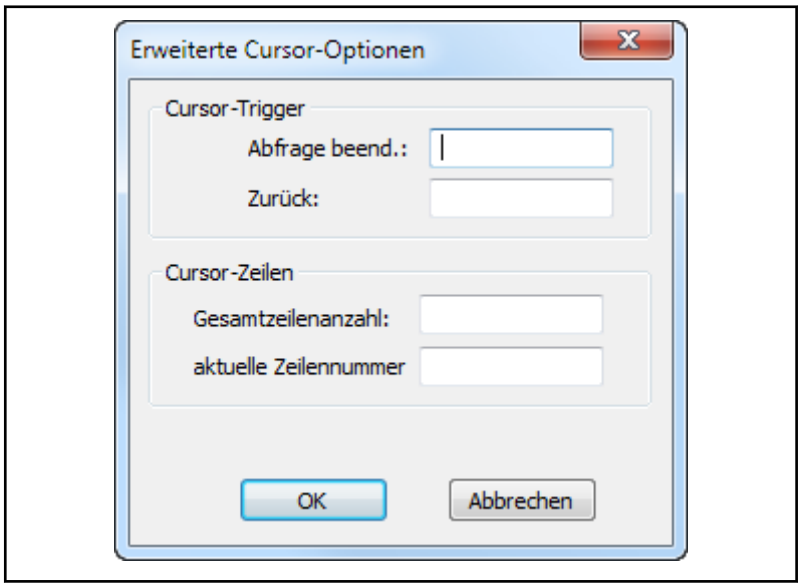


Abb.10-13: Dialog Weitere Cursor-Einstellungen

– **Abfrage beendet**: Geben Sie einen beliebigen Tag ein; wenn sich der Tag-Wert ändert, gibt der Cursor den Ergebnissatz frei.

– **Vorherige**: Geben Sie einen beliebigen Tag ein; wenn sich der Tag-Wert ändert, bewegt sich der Cursor in die vorherige Zeile des Ergebnissatzes und kopiert diese Werte in den im Hauptteil des Arbeitsblatts konfigurierten Tag.

– **Gesamtanzahl Zeilen**: Geben Sie einen numerischen Tag ein, der die Gesamtanzahl der Zeilen im Ergebnissatz erhält.

– **aktuelle Zeilennummer**: Geben Sie einen numerischen Tag ein, der die Nummer der aktuellen Zeile erhält (d.h. die Position das Cursors). Beim ersten Öffnen eines Ergebnissatzes mit dem Auslöser **Auswählen**, ist diese Zahl 1. Jeder Auslöser **Nächste** erhöht diese Zahl und jeder Auslöser **Vorherige** verringert sie.

- *Bereich Trigger einfügen*

– Feld **Einfügen**: Geben Sie einen beliebigen Tag ein; wenn sich der Tag-Wert ändert, wird eine neue Zeile mit den aktuellen Werten der im Hauptteil des Arbeitsblatts konfigurierten Tags eingefügt.

– Feld **Aktualisieren**: Geben Sie einen beliebigen Tag ein; wenn sich der Tag-Wert ändert, werden alle Zeilen des Ergebnissatzes mit den aktuellen Werten der im Hauptteil des Arbeitsblatts konfigurierten Tags überschrieben.

– Feld **Löschen**: Geben Sie einen beliebigen Tag ein; wenn sich der Tag-Wert ändert, werden alle Zeilen des Ergebnissatzes gelöscht.



Tabellen-Trigger sind nur verfügbar, wenn für **Typ** die Einstellung **Tabelle** ausgewählt wurde, da diese Operationen in der gesamten Tabellenzeile arbeiten.

Hauptteil des Arbeitsblatts

Im Hauptteil des Arbeitsblatts können Anwendungsvariablen auf Seite 116 auf die Spalten (Felder) des Ergebnissatzes abgebildet werden. Geben Sie für jede Zeile im Hauptteil einen **Variablennamen** und dessen entsprechende **Spalte** ein. Welche Spalten verfügbar sind, hängt davon ab, wie der Ergebnissatz ausgewählt wird.

Diese Auswahl kann sich während der Laufzeit ändern, es müssen also unbedingt alle notwendigen Spalten abgebildet werden.

11 Kommunikation mit anderen Geräten

11.1 Kommunikation mit anderen Geräten, Allgemeines

Kommunikations-Tasks/-Arbeitsblätter dienen dem Austausch von Tag-Werten mit Remote-Geräten (z.B. SPS-Steuerungen), anderen "WinStudio"-Projekten und allen Systemen, die unterstützte Protokolle wie OPC und DDE implementieren.

Die Kommunikation von Tags mit einer Datenquelle erfolgt mit Hilfe von Kommunikationstreibern und den zugehörigen Arbeitsblättern im Ordner **WinStudio ▸ Variablen ▸ Kommunikation ▸ <Treibername>** konfiguriert.

11.2 Konfiguration der direkten Kommunikation mit einem Remote-Gerät

11.2.1 Konfiguration der direkten Kommunikation mit einem Remote-Gerät, Allgemeines

Bei einem Kommunikationstreiber handelt es sich um eine DLL, die das Kommunikationsprotokoll eines bestimmten Remote-Gerätes implementiert und somit deren Daten Lesen und Schreiben kann. WinStudio bietet hierzu einige Standardkommunikationstreiber an. Für die Kommunikation mit VPx- und VEx-Geräten steht außerdem der BR_WS-Treiber zur Verfügung. Für zahlreiche Geräte existieren weitere Treiber, die Sie zu Ihrer Installation hinzufügen können, um mit diesen Geräten Daten auszutauschen.

Im Eigenschaftendialog eines Treibers können Sie die Kommunikationsschnittelle zwischen dem Projekt und den Remote-Geräten definieren.

☞ Im **Hilfe**-Menü finden Sie eine Beschreibung der Funktionen und Merkmale, die bei allen Treiber Standard sind. Bei der Entwicklung eines Projekts kann auch die für jeden Kommunikationstreiber zur Verfügung gestellte spezielle Dokumentation hinzugezogen werden. Diese Dokumentation befindet sich normalerweise im Verzeichnis **DRV**.

Bei der Konfiguration eines Kommunikationstreibers müssen Sie die Schnittstellenparameter angeben (z.B. die Stationsadresse und die Baudrate) sowie die Geräteadresse definieren und diese dann den Projekt-Tags zuordnen.

Zum Hinzufügen oder Entfernen eines Treibers gehen Sie folgendermaßen vor:

- Öffnen Sie das Kontextmenü des Ordners **WinStudio ▸ Variablen ▸ Kommunikation** und wählen Sie den Eintrag "Treiber hinzufügen/ entfernen"

Es öffnet sich folgender Dialog, der eine Liste der verfügbaren Treiber anzeigt.

Kommunikation mit anderen Geräten

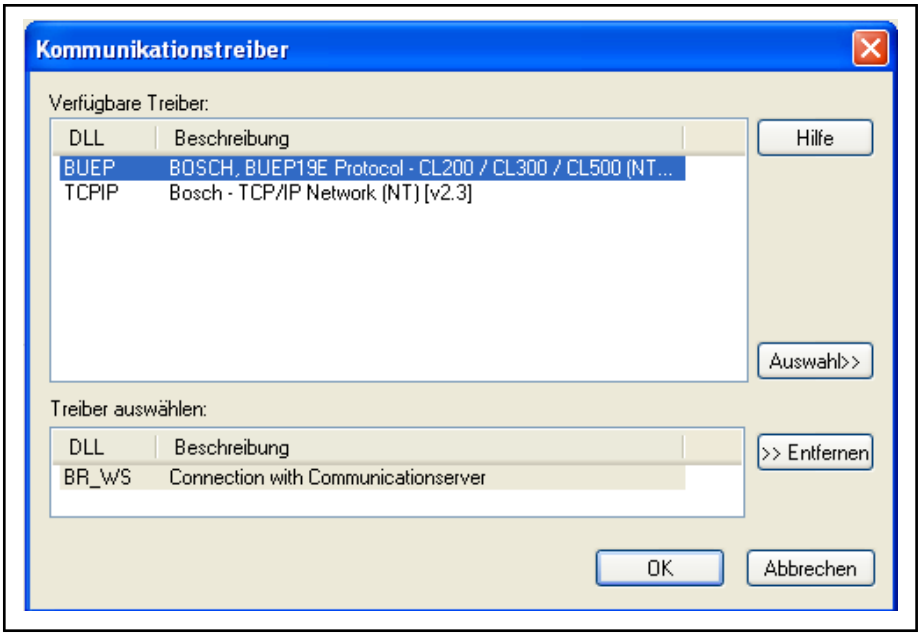


Abb.11-1: Dialog Kommunikationstreiber

Die Einstellungen in diesem Dialog besitzen folgende Bedeutung:

- **Verfügbare Treiber**: Liste aller verfügbaren Treiber mit einer kurzen Beschreibung
- **Hilfe**: Mit der Schaltfläche **Hilfe** können Sie für den unter "Verfügbare Treiber" oder "Ausgewählte Treiber" selektierten Treiber die zugehörige Hilfe mit einer ausführlichen Beschreibung der Konfiguration der Treibers öffnen. Falls Sie zusätzliche Treiber installiert haben ist u.U. für diese keine Hilfe verfügbar. In diesem Fall ist die Schaltfläche "Hilfe" deaktiviert
- **Auswählen**: Klicken Sie diese Schaltfläche an, wenn Sie den im Feld **Verfügbare Treiber** markierten Treiber in Ihrem Projekt verwenden möchten.
- **Ausgewählte Treiber**: Zeigt alle ausgewählten Treiber und ggf. deren Beschreibungen an.
- **Entfernen**: Klicken Sie diese Schaltfläche, wenn Sie den im Feld **Ausgewählte Treiber** markierten Treiber nicht mehr verwenden und aus Ihrem Projekt entfernen möchten.

Nach einer Änderung der Einstellungen können Sie diese mit der Schaltfläche **OK** übernehmen. Im Ordner "Kommunikation" wird ein entsprechender Unterordner für den Hinzugefügten Treiber erstellt bzw wieder entfernt.

Im Kontextmenü eines Treiberunterordners finden Sie die den Eintrag "Eigenschaften". Dieser öffnet Konfigurationsdialog für diesen Treiber, in dem Sie folgende Einstellungen vornehmen können:

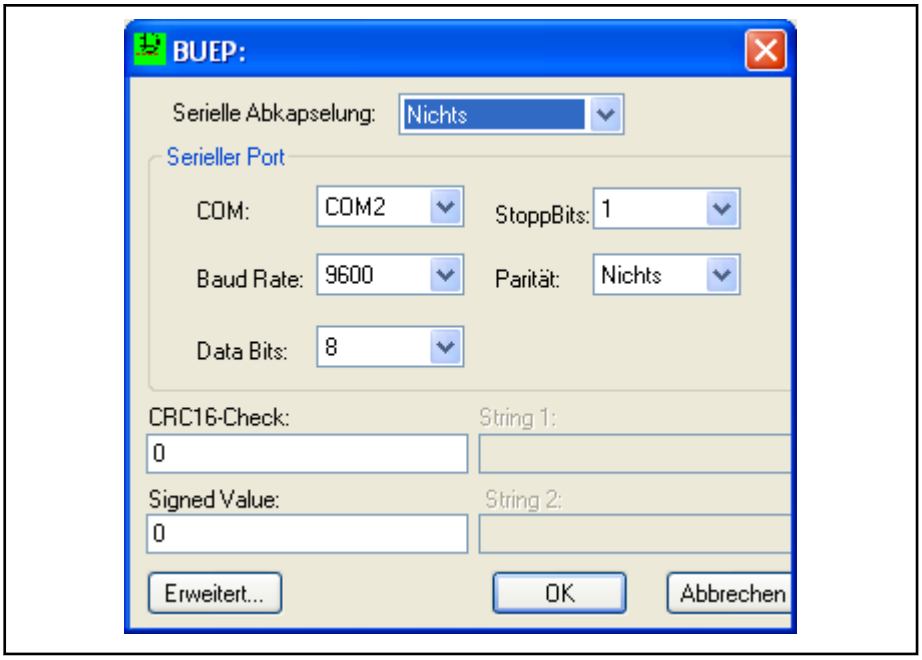


Abb.11-2: Dialog Beispiel für Kommunikationsparameter

Die einzelnen Treibereinstellungen besitzen folgende Bedeutung:

- **Serielle Datenkapselung**: Ermöglicht seriellen Treibern die Kommunikation mit Modem-, TCP/IP- oder UDP-Verbindungen. Diese Einstellung wird nur für serielle Treiber, die mit der UNICOMM-Bibliothek entwickelt wurden, unterstützt. Dies trifft auf die meisten in diesem Produkt verfügbaren seriellen Treiber zu.

☞ Dieser Abschnitt beschreibt nur die Option **"Nichts"**, bei der der Treiber sich mit Hilfe eines normalen seriellen Kanals verbinden kann. Weitere Informationen zu anderen Datenkapselungsmodi finden Sie nachfolgend unter „Verwendung von TCP/IP- und UDP-Datenkapselung“ und „Verwendung von Modemverbindungen“. Der Abschnitt „Tests für serielle Datenkapselung“ beschreibt die Treiber, die mit den Modi Modem, TCP/IP und UDP getestet wurden.

- **COM**: Auswahl eines seriellen Kommunikationsanschlusses.
- **Baudrate**, **Datenbits**, **Stopbits und Parität**: Diese Parameter werden für die Konfiguration eines seriellen Ports benötigt.
- **Long1**, **Long2**, **String1** und **String2**: Diese Einstellungen sind treiberspezifisch. Im obigen Beispiel verwendet der Treiber **Long1** zum Aufbau der Fehlererkennungsmethode und **String1** zum Definieren der Art der SPS-Familie.
- **Erweitert**: Mit dieser Schaltfläche öffnen Sie einen Dialogs in dem Sie weitere Einstellungen vornehmen können: Dieser Dialog wird zum Ändern der Standard-Treiberparameter verwendet.



Abb.11-3: Dialog Erweiterte Einstellungen

Festlegen oder Ändern der Standard-Treiberparameter wie folgt:

- *Bereich Zeitüberschreitung (ms):*
  - **Meldungsbeginn**: Festlegen der Zeitüberschreitung für den Meldungsbeginn.
  - **Meldungsende**: Festlegen der Zeitüberschreitung für das Meldungsende.
  - **Intervall zwischen Zeichen**: Festlegen der Zeitüberschreitung zwischen den einzelnen Zeichen.
  - **Warte CTS**: Festlegen der Zeitüberschreitung für die Wartezeit der Sendebereitschaft (CTS = Clear to send).
- *Bereich Handshake:*
  - **Steuerung RTS**: Festlegen, ob die Steuerung der Sendeanfrage (RTS = Request to send) verwendet wird.
  - **CTS überprüfen**: Festlegen, ob die Überprüfungsart „Sendebereitschaft“ verwendet wird.
  - **DTR deaktivieren**: Durch Anklicken (Aktivieren) dieses Kontrollkästchens wird die Funktion DTR (data terminal ready = Datenterminal bereit) deaktiviert (der Treiber setzt vor Kommunikationsstart nicht das DTR-Signal).
  - **IR aktivieren** ("nur auf Windows-Embedded-Zielsystemen verfügbar"): Durch Anklicken (Aktivieren) dieses Kontrollkästchens wird dem seriellen Treiber ermöglicht, eine Infrarotschnittstelle (COM2-Port) anstelle eines standardmäßigen seriellen Ports für die Kommunikation mit dem Gerät (z.B. SPS, E/A, Pocket-Computer usw.) zu verwenden.
- *Bereich Protokoll:*
  - **Station**: Einige Slave-Treiber wie z.B. der Modbus Slave (MODSL) benötigen eine Slave-Netzwerkadresse. Verwenden Sie dieses Feld zum Festlegen der Slave-Adresse.

  - **Wiederholungen**: Geben Sie einen numerischen Wert ein, um festzulegen, wie oft der Treiber versucht, den gleichen Kommunikationsbefehl auszuführen, bevor ein Kommunikationsfehler für diesen Befehl in Betracht gezogen wird.
- *Bereich Pufferlänge (Bytes):*
  - **TxPuffer**: Festlegen der Pufferlänge für die Übertragung (in Bytes):
  - **RxPuffer**: Festlegen der Pufferlänge für den Empfang (in Bytes):
- *Bereich Simultane Anfragen (nur für bestimmte Treiber verfügbar):*
  - **Maximum**: Festlegen der Höchstzahl der Anfragen, die gleichzeitig an alle angeschlossenen Geräte gesendet werden können.
  - **Maximum pro Station**: Festlegen der Höchstzahl der Anfragen, die gleichzeitig an ein einzelnes Gerät gesendet werden können.



Die Höchstzahl der gleichzeitigen Anfragen hängt vom jeweiligen Gerät und den Protokollspezifikationen ab. Bitte ziehen Sie die Dokumentation der Gerätedokumentation zu Rate.

Die Entwicklungsanwendung stellt zwei Schnittstellen zur Verfügung, die zum Konfigurieren der Treiber verwendet werden können (Verbinden von Projekt-Tags mit Geräteadressen):

- <HAUPTTREIBERBLATT> Stellt die einfachste Methode zur Konfiguration der Kommunikation zwischen Projekt-Tags und Geräteadressen dar. Mit dieser Schnittstelle können Tags automatisch gruppiert werden, um die bestmögliche Leistung während der Laufzeit zu ermöglichen. Diese Schnittstelle kann nicht zur Steuerung der Zeit verwendet werden, die benötigt wird, um eine Gruppe von Tags einzeln zu scannen.
- STANDARDTREIBERBLÄTTER auf Seite 387 Ermöglicht die Steuerung der Zeit, die benötigt wird, um eine Gruppe von Tags einzeln zu scannen.

Beide Blätter können gleichzeitig verwendet werden.

Verwendung von TCP/IP- und UDP-Datenkapselung

Die meisten seriellen Treiber gestatten die Verwendung von TCP/IP- und UDP-Datenkapselung. Der Datenkapselungsmodus wurde zur Bereitstellung von Kommunikation mit seriellen Geräten entwickelt, die an Terminal-Server in Ihren Ethernet- oder Drahtlosnetzwerken angeschlossen sind. Ein Terminal-Server kann als virtueller serieller Port betrachtet werden. Er konvertiert TCP/IP- oder UDP/IP-Meldungen in Ihrem Ethernet- oder Drahtlosnetzwerk in serielle Daten. Nachdem die Meldung in das serielle Format konvertiert wurde, können Standardgeräte, die serielle Kommunikation mit dem Terminal-Server unterstützen, angeschlossen werden.

Das folgende Bild zeigt ein Beispiel für die Anwendung dieser Lösung:

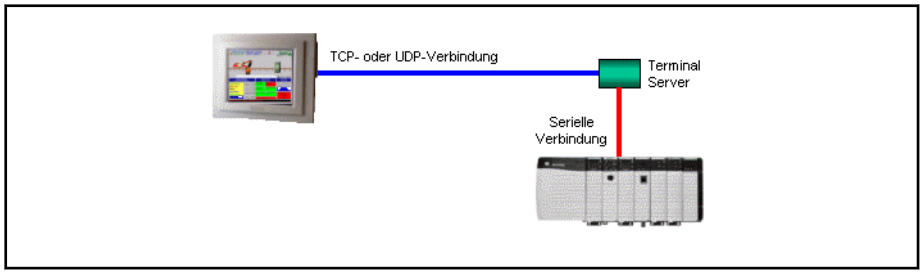


Abb.11-4: TCP/IP-Datenkapselung

Die Datenkapselung wird durch die folgenden Schritte freigegeben:

Kommunikation mit anderen Geräten

1. Führen Sie einen Rechtsklick auf den Treiberordner aus und wählen Sie dann **Einstellungen** aus dem Kontextmenü. Dadurch erhalten Sie Zugriff auf die Kommunikationsparameter.
2. Wählen Sie im Feld **Serielle Datenkapselung TCP/IP** oder **UDP/IP**:

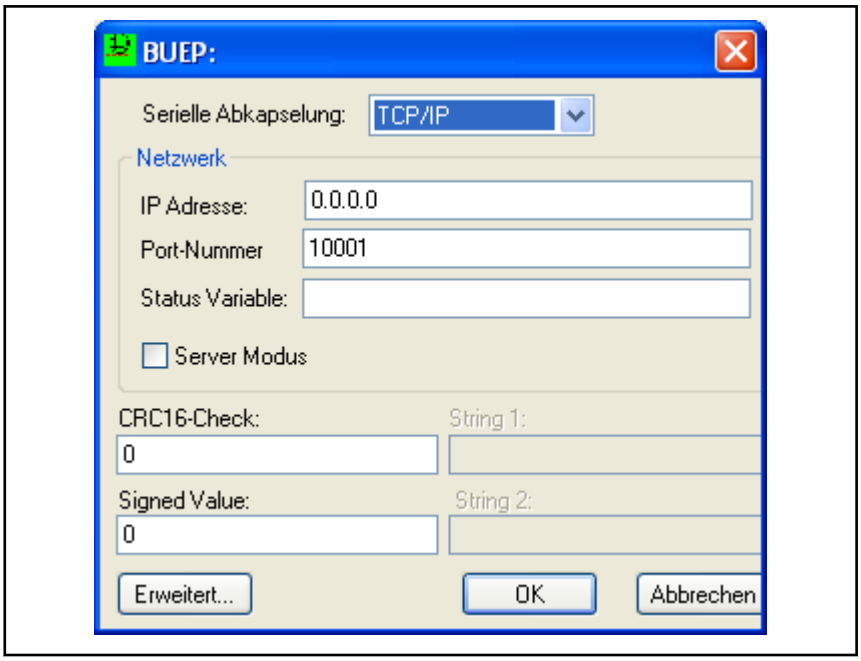


Abb.11-5: Dialog: Datenkapselung

Folgende Felder stehen zur Verfügung:

- **IP-Adresse**: Geben Sie die IP-Adresse für den Terminal-Server an. In diesem Feld können Tags in geschweiften Klammern eingegeben werden.
- **Port-Nummer**: Geben Sie die TCP/IP- oder UDP/IP-Port-Nummer ein.
- **Status-Tag**: Dieses Feld steht nur bei Verwendung von TCP/IP zur Verfügung. Der Tag in diesem Feld erhält den Wert 1, wenn die TCP/IP-Verbindung aufgebaut ist; andernfalls erhält er den Wert 0.
- **Server-Modus**: Die TCP/IP-Datenkapselung ermöglicht den Server-Modus, wodurch der Remote Client für den Aufbau der Verbindung zur Ermöglichung der Kommunikation verantwortlich ist.

Verwenden von Modemverbindungen

Die meisten seriellen Treiber erlauben die Verwendung von Modemverbindungen. Die Modemverbindung wurde entwickelt, um die Kommunikation mit seriellen Geräten über eine Telefonleitung zu ermöglichen.

Das folgende Bild zeigt ein Beispiel für die Anwendung dieser Lösung:

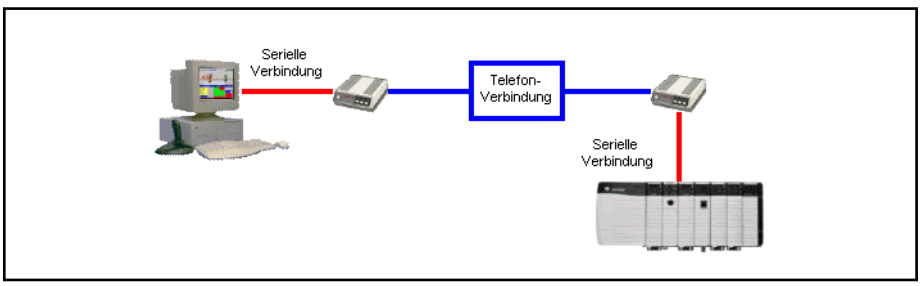


Abb.11-6: Modemverbindung

Die Modemverbindung wird durch die folgenden Schritte freigegeben:

1. Führen Sie einen Rechtsklick auf den Treiberordner aus und wählen Sie dann **Einstellungen** aus dem Kontextmenü. Dadurch erhalten Sie Zugriff auf die Kommunikationsparameter.
2. Wählen Sie nun im Menü **Serielle Datenkapselung** den Eintrag **Modem**:

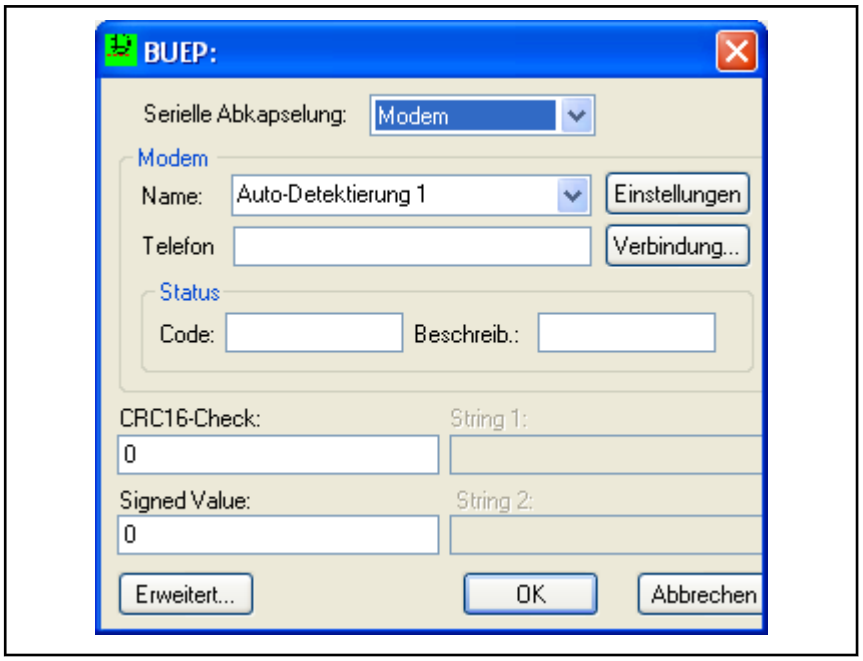


Abb.11-7: Dialog: Serielle Datenkapselung

HINWEIS **Die Option Modem wird für Pocket PC v3.00 oder früher nicht unterstützt.**

Folgende Einstellungen stehen zur Verfügung:

- **Name**: Wählen Sie das Modem, das der Treiber zum Aufbau der Verbindung verwenden soll.

  Wenn der Modemname nicht bekannt ist, verwenden Sie die Option Automatische Erkennung:

  – **Automatische Erkennung 1** verwendet das erste verfügbare Modem
  – **Automatische Erkennung 2** verwendet das zweite verfügbare Modem
  – **Automatische Erkennung 3** verwendet das dritte verfügbare Modem und
  – **Automatische Erkennung 4** verwendet das vierte verfügbare Modem
- **Telefon**: Geben Sie eine Telefonnummer ein, die der Treiber zur Verbindung mit dem Remote-Gerät verwendet. In diesem Feld können Tags in geschweiften Klammern eingegeben werden, wenn Sie die Telefonnummer zur Laufzeit ändern möchten.
- **Einstellungen**: Klicken Sie auf diese Schaltfläche, um die Modemeinstellungen zu konfigurieren. Welches Fenster nach dem Klicken auf diese Schaltfläche angezeigt wird, hängt vom verwendeten Betriebssystem und dem Modemtyp ab.

HINWEIS	**Die durch Klicken auf diese Schaltfläche konfigurierten Einstellungen werden nicht mit Ihrem Projekt gespeichert.**

Die Informationen werden in der Registry des Betriebssystems gespeichert und gelten nur für den Rechner, mit dem Sie arbeiten. Wenn Sie Ihr Projekt auf einem anderen Rechner installieren, müssen diese Einstellungen neu konfiguriert werden.

- **Verbindung**: Klicken zum Öffnen des Fensters "Verbindungssteuerung"

  Die Standard-Verbindungseinstellungen sollten für die meisten Projekte ausreichend sein. Sie können aber durch Klicken auf diese Schaltfläche die Verbindung vollständig steuern und auch eingehende Anrufe freigeben.

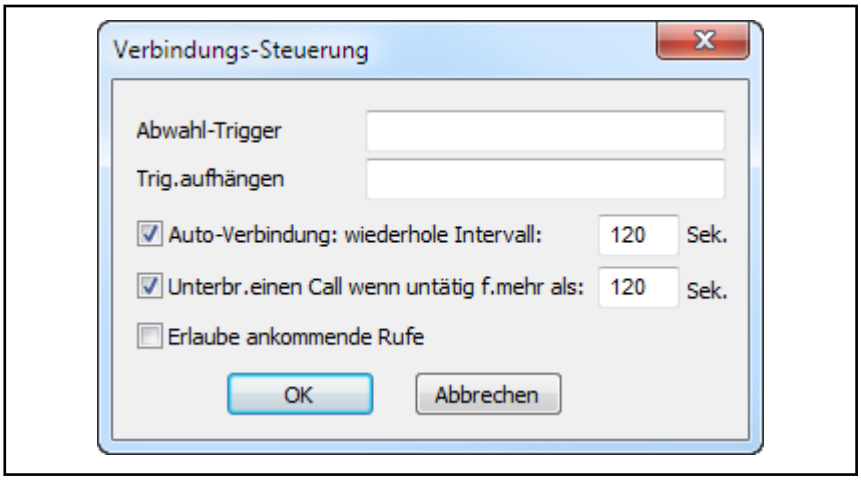


Abb.11-8: Dialog Verbindungssteuerung

  - **Auslöser Anwählen**: Wenn sich der in diesem Feld konfigurierte Tag ändert, versucht der Treiber, sich mit dem Remote-Gerät zu verbinden. Wenn die Verbindung bereits aufgebaut ist, wird der Befehl ignoriert. Bei Verwendung von" Automatisch Verbinden" muss dieses Feld nicht verwendet werden.
  - **Auslöser Auflegen**: Wenn sich der in diesem Feld konfigurierte Tag ändert, trennt der Treiber die Verbindung mit dem Remote-Gerät. Wenn keine Verbindung zu dem Gerät besteht, wird der Befehl ignoriert. Bei Verwendung von "Trennen wenn Verbindung ungenutzt länger als" muss dieses Feld nicht verwendet werden.
  - **Automatisch verbinden**: Wenn diese Option aktiviert ist, versucht der Treiber, sich vor dem Senden von Informationen mit dem Remote-Gerät zu verbinden. Wenn die Verbindung nicht zustande kommt, wird nach Ablauf des Intervalls für Wiederholungsversuche ein neuer Versuch unternommen.
  - **Trennen wenn Verbindung ungenutzt länger als**: Wenn diese Option ausgewählt ist, trennt der Treiber automatisch die Verbindung zum Remote-Gerät, wenn nach Ablauf der angegeben Zeit keine Kommunikation stattgefunden hat.
  - **Eingehende Anrufe ermöglichen**: Aktivieren Sie diese Option, wenn Sie den Treiber für eingehende Anrufe vom Remote-Gerät freigeben wollen. Sie können den Auslöser Auflegen verwenden, um aufgebaute Anrufe zu trennen. Bitte beachten Sie, dass ein

Treiber sowohl eingehende als auch ausgehende Anrufe verwenden kann.

- *Bereich Status*
  - **Code**: Geben Sie einen Tag ein, der den Statuswert der Verbindung erhält.

    Die Statuswerten haben folgenden Bedeutung:
    - 0 = Getrennt
    - 1 = Verbunden
    - 2 = Anwahl
    - 3 = Trennen
    - 4 = Leitung schließen
  - **Beschreibung**: Geben Sie einen Tag ein, der eine vollständige Beschreibung des aktuellen Status erhält. Die Beschreibung wird mit dem Feld **Code** verbunden, liefert aber zusätzliche Informationen zum aktuellen Status.

Tests Serielle Datenkapselung:

Die meisten seriellen Treiber sollten mit jedem seriellen Datenkapselungsmodus funktionieren. Allerdings wurden die meisten Treiber vor der Erstellung der Datenkapselungsmodi entwickelt. Die folgende Tabelle führt die Treiber auf, die mit bestimmten Datenkapselungsmodi getestet wurden. Wenn Sie einen nicht aufgeführten Treiber verwenden wollen und unsicher sind, ob dieser funktionieren wird, wenden Sie sich bitte an Ihren Händler.

Treiber	Modem	TCP/IP	UDP/IP
MODSL	X	X	X
ABKE	X	X	X
MODBU	X		X
OMETH	X		

X = Element wurde getestet

11.2.2 Treiberblätter

Für jeden Treiber können Sie mehrere Treiberblätter erstellen, in denen Sie Tags eintragen und diese mit einer Adresse auf dem Remote-Gerät verknüpfen können. Außerdem besitzt jedes Treiberblatt verschiedene Konfigurationsmöglichkeiten, mit denen Sie deren Ausführung zur Laufzeit steuern können.

Um ein neues Treiberblatt zu erstellen, öffnen Sie das Kontextmenü des Treiberordners und wählen Sie "Neues Treiberblatt". Jedes Treiberblatt besitzt eine eindeutige Nummer, die Sie jederzeit über die Kontextfunktion "Umbenennen" ändern können. Zum Öffnen eines Treiberblatts führen Sie einen Doppelklick auf das neue Element im Treiberordner aus.

Kommunikation mit anderen Geräten

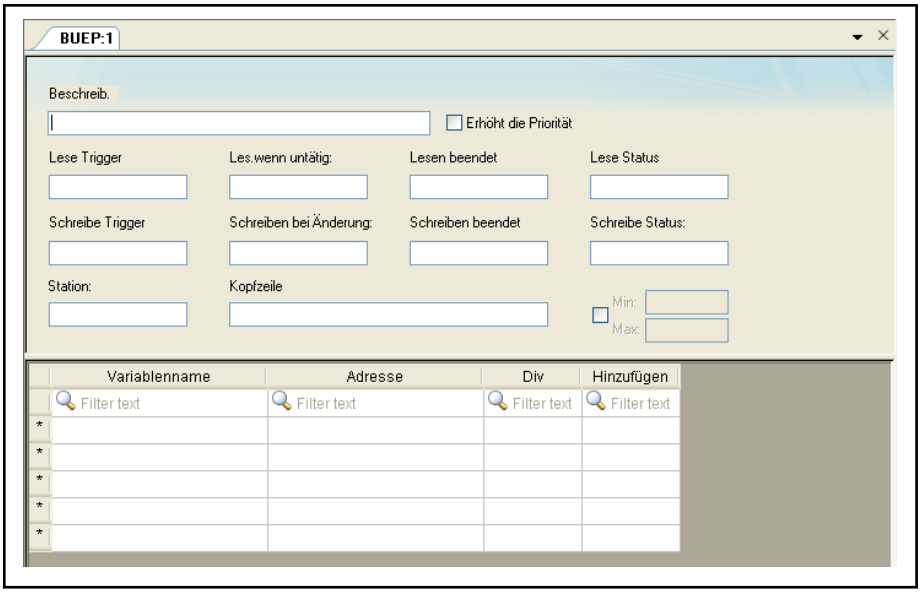


Abb.11-9: Treiberblatteditor

Der Treiberblatteditor gliedert sich in zwei Bereiche:

- Der "Titelbereich" (oberer Teil) enthält Parameter, die die Kommunikation aller im Hauptteil dieses Arbeitsblatts konfigurierten Tags beeinflussen.
- Der "Hauptteil" (unterer Teil) verknüpft Tags, die im Projekt vorhanden sind, mit Geräteadressen.

Die Parameter im "Titelbereich" werden wie folgt verwendet:

- **Beschreibung**: Geben Sie zu Dokumentationszwecken einen Beschreibungstext ein .
- **Priorität erhöhen**: Aktivieren Sie diese Option, wenn dieses Treiberblatt in jedem Ausführungszyklus zuerst bearbeitet werden soll.

HINWEIS	**Wenn die Option Priorität erhöhen aktiviert ist, muss diesem Arbeitsblatt besondere Aufmerksamkeit gewidmet werden.**

Wenn das Arbeitsblatt weiterhin Kommunikationsbefehle auslöst, ist das Projekt ggf. nie in der Lage, andere Treiberblätter auszuführen.

- **Lese-Trigger**: Geben Sie einen Tag ein, wenn bei jeder Wertänderung des Tags ein Lese-Befehl für alle Tags um Hauptteil ausgeführt werden soll.
- **Lesen wenn untätig**: Geben Sie einen Tag-Wert oder eine Konstante ein. Verwenden Sie einen Tag-Wert (oder eine Konstante) größer Null, um das Lesen vom Gerät freizugeben.

HINWEIS	**Prüfen Sie bei Verwendung einer Konstante (ungleich Null) sicher, ob Ihr Projekt kontinuierliches Lesen erfordert.**

Kontinuierliches Lesen beeinträchtigt unter Umständen die Ausführung Ihres Projekts.

- **Lesen beendet**: Geben Sie einen Tag ein, wenn Sie die Ausführung des Lese-Befehls überwachen möchten. Der Kommunikationstreiber ändert den Tag nach jedem "Lese"-Befehl.
- **Status lesen**: Geben Sie einen Tag ein und der Kommunikationstreiber aktualisiert den Tag mit dem Status auf Seite 841 des letzten "Lese"-Befehls.
- **Schreibe-Trigger**: Geben Sie einen Tag ein, wenn bei jeder Wertänderung des Tags ein Schreib-Befehl für alle Tags um Hauptteil ausgeführt werden soll.
- **Schreiben bei Tag-Änderung**: Geben Sie einen Tag-Wert oder eine Konstante (ungleich null) ein, um den Kommunikationstreiber für die kontinuierliche Überprüfung auf Veränderungen in diesem Tag freizugeben. Wenn eine Veränderung des Tag-Werts auftritt, schreibt das Projekt diesen Wert in eine Adresse im Außengerät.
- **Schreiben beendet**: Geben Sie einen Tag ein und der Kommunikationstreiber aktualisiert den Wert des Tags, wenn ein "Schreib"-Befehl beendet wurde.
- **Schreibe-Status**: Geben Sie einen Tag ein und der Kommunikationstreiber aktualisiert den Tag mit dem Status auf Seite 841 des letzten "Schreib"-Befehls.
- **Station**: Geben Sie die Nummer der Gerätestation im Netzwerk ein. Die Syntax in diesem Feld ist je nach Kommunikationstreiber unterschiedlich. Weitere Informationen finden Sie in der Dokumentation für den entsprechenden Treiber.



Wenn der Treiber für serielle Datenkapselung über TCP/IP oder UDP/IP konfiguriert ist, dann kann die Station für einige Treiber mit Hilfe des folgenden Formats festgelegt werden:

IP_adresse:port_nummer|station

Zum Beispiel:

10.169.25.18:1234|Station5

Informationen zur Verfügbarkeit dieser Funktion für den von Ihnen gewählten Treiber finden Sie in der Dokumentation für den Treiber.

- Feld **Kopfzeile**: Legen Sie den Datentyp und/oder die Anfangsadresse zum Lesen oder Schreiben im Gerät fest. Die Syntax in diesem Feld ist je nach Kommunikationstreiber unterschiedlich. Weitere Informationen finden Sie in der entsprechenden <Treiberdokumentation>.



In den Feldern **Station** und **Kopfzeile** kann Text mit Tag-Werten mit der Syntax **text {tag}** verwendet werden.

- **Min** und **Max**: Aktivieren Sie diese Option, um die Minimal- und Maximalwerte für Außengerätedaten festzulegen.
- **Min** und **Max** (werden erst bei Aktivieren des Kontrollkästchens **Min und Max** freigegeben): Geben Sie einen Wertebereich zum Konvertieren in eine technische Einheit ein. Diese Felder bestimmen den Minimal- und Maximalbereich von Werten. Diese Werte beeinflussen alle Tags im Arbeitsblatt. Zum Beispiel enthält Speicher Werte von 0 bis 4095, d.h. 0% bis 100% in der Benutzerschnittstelle. Für dieses Beispiel muss also 0 bis 100 für die Min- und Max-Tag-Parameter eingegeben werden.

Die Parameter im "Hauptteil" werden wie folgt verwendet:

- **Variablenname**: Geben Sie einen Namen eines Tags an, der mit einer Geräteadresse verknüpft werden soll.
- **Adresse**: Geben Sie eine Außengeräteadresse (oder einen Adressversatz) mit Bezug auf den Projekt-Tag ein. Die Syntax in diesem Feld ist je nach Kommunikationstreiber unterschiedlich. Weitere Informationen finden Sie in der Dokumentation für den entsprechenden Treiber.
- **Div**: Festlegen einer Divisionskonstante, wenn eine Anpassung der Skalierung erforderlich ist. Das Projekt verwendet diesen Wert als Divisionsfaktor in einer Lese-Operation und als Multiplikationsfaktor in einer Schreib-Operation. Dieses Feld darf nicht verwendet werden, wenn im Hauptteil der Konfiguration bereits **Min** oder **Max** verwendet wird.
- **Addieren**: Festlegen einer Additionskonstante, wenn eine Anpassung der Skalierung erforderlich ist. Das Projekt verwendet diesen Wert als Additionsfaktor in einer Lese-Operation und als Subtraktionsfaktor in einer Schreib-Operation. Dieses Feld darf nicht verwendet werden, wenn im Hauptteil der Konfiguration bereits **Min** oder **Max** verwendet wird.



Die Höchstzahl an Tags in jedem Treiberkommunikationsarbeitsblatt beträgt 512. Für einige Treiber kann diese Zahl kleiner sein. Weitere Informationen sind in Ihrer <Treiberdokumentation>.

Für Leseoperationen gilt:

tag = (Wert im Gerät) / Div + Add

Für Schreiboperationen gilt:

Wert im Gerät = (tag - Add) * Div

Wenn die Zellen in den Feldern **Div** und **Add** leer bleiben, wird diese Funktion ignoriert.

11.3 Konfigurieren einer OPC-Client-Verbindung mit einem OPC-Server

OPC-Arbeitsblätter werden zur Kommunikation mit allen Systemen verwendet, die das OPC-Server-Protokoll implementieren.

Der OPC-Client implementiert den OPC-Standard wie im Dokument "OLE für Prozesssteuerung Datenzugriff Standardversion 1.0A" beschrieben. Dieser ist verfügbar auf der OPC Foundation Website.



Vor der Verwendung des OPC-Client-Tasks/-Arbeitsblatts in Ihrem Projekt müssen Sie sicherstellen, dass die OPC-Server-Software ordnungsgemäß auf dem System, mit dem Sie sich verbinden wollen, installiert und konfiguriert ist.

Zum Konfigurieren einer neuen Verbindung fügen Sie im Ordner **WinStudio ▸ Variablen ▸ Kommunikation ▸ OPC** im Projektexplorer ein neues OPC-Arbeitsblatt ein.

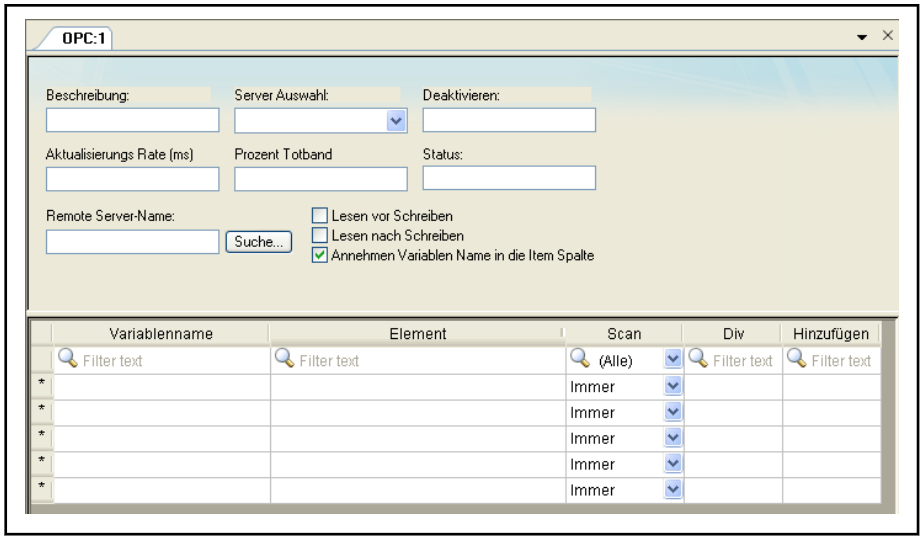


Abb.11-10: Beispiel OPC-Arbeitsblatt

Verwenden Sie die folgenden Parameter für die Konfiguration eines OPC-Treiberblatts:

- **Beschreibung**: Geben Sie zu Dokumentationszwecken eine Beschreibung für das OPC-Arbeitsblatt ein.
- **Server-Auswahl**: Geben Sie den Namen des Servers ein, mit dem Sie eine Verbindung herstellen möchten. Wenn der Server bereits auf dem Rechner installiert ist, kann der Servername aus der Liste ausgewählt werden.
- **Deaktivieren**: Geben Sie einen Tag oder eine Konstante mit einem Wert ungleich "0" ein, um die Kommunikation mit dem OPC-Server zu deaktivieren. Geben Sie "0" ein oder lassen Sie das Feld leer, um die Kommunikation zu aktivieren.
- **Aktualisierungsrate Lesen**: Legen Sie fest, wie oft der Server diese Gruppe aktualisieren soll (in Millisekunden). Bei Eingabe von "0" verwendet der Server die schnellste sinnvolle Rate.
- **Prozent Totband** (nur für analoge Elemente gültig): Legen Sie fest, bei welcher prozentualen Änderung eines Elementwerts eine Benachrichtigung durch den Server erfolgt.
- **Status**: Geben Sie den Namen eines Tags ein, um den Status der Verbindung zu erhalten. Der Status "Gut" entspricht dem Wert 1.
- **Remote-Server-Name**: Knotenname oder IP-Adresse des Servers im Knotennetzwerk.
- **Lesen vor Schreiben**: Bei Aktivierung dieser Option werden direkt vor dem Schreiben von neuen Variablenwerten auf den Server die Ursprungswerte vom OPC-Server gelesen. Das Projekt puffert dazu erst die neuen zu schreibenden Werte und liest dann die Ursprungswerte vom Server. Erst nach der Synchronisation des Projekts mit dem Server werden die neuen Werte vom Puffer zum Server geschrieben.
- **Lesen nach Schreiben**: Bei Aktivierung dieser Option werden direkt nach dem Schreiben von neuen Variablenwerten auf den Server die Werte vom OPC-Server zurückgelesen.

HINWEIS

Die Optionen "Lesen vor Schreiben" und "Lesen nach Schreiben" werden angeboten, weil die Spezifikationen von OPC Client/Server besagen, dass der Wert eines Elements auf dem Client – in diesem Fall Ihrem Projekt – sich nur ändern darf, wenn die Änderung vom Server kommt. Auf diese Weise bleibt der Client immer synchron mit dem Server.

Allerdings könnte Ihr Projekt so angelegt sein, dass diese Werte durch Laufzeitprozesse oder Benutzereingaben geändert werden. Daher besteht die beste Weise zum Ändern der Werte bei gleichzeitiger Synchronisation mit dem Server darin, den Anschein zu erwecken, dass die Änderungen vom Server stammen. Wenn beide Optionen aktiviert sind, findet bei jedem Scan des OPC-Arbeitsblatts der folgende Ablauf statt:

1. Die neuen Werte auf dem Client werden gepuffert.
2. Die Ursprungswerte auf dem Server werden vom Client gelesen – d.h. der Client wird mit dem Server synchronisiert.
3. Die neuen Werte werden vom Puffer zum Server geschrieben.
4. Die neuen Werte auf dem Server werden vom Client gelesen – d.h. der Client wird wieder mit dem Server synchronisiert.

Am Ende jedes Lese-/Schreibzyklus spiegeln die Werte den Zustand in Ihrem Projekt wider, obwohl das Projekt technisch gesehen einfach nur mit dem Server synchron bleibt.

In den meisten Projekten sollten beide Optionen aktiviert sein. In einigen Projekten kann dies allerdings dazu führen, dass Elemente zwischen den Ursprungswerten und den neuen Werten hin- und herspringen. Wenn dies ein Problem ist, versuchen Sie, diese Elemente in ein anderes OPC-Arbeitsblatt zu verschieben, in dem die Optionen **Lesen vor Schreiben** und **Lesen nach Schreiben** deaktiviert sind.

- **Annehmen Variablen-Name in Item-Spalte**: Wenn diese Option aktiviert ist, wird der in geschweiften Klammern eingeschlossene Text im Feld "Element" als Tag-Name (von Typ String) interpretiert. In diesem Fall wird der Wert dieses Tags als Name des Elements vom OPC-Server verwendet. Dadurch können Sie während der Laufzeit die Verknüpfung eines Tag mit OPC-Elementen ändern, indem Sie den Werte des konfigurierten Tags ändern. Wenn die Option **Annehmen Variablen-Name in Item-Spalte** nicht aktiviert ist, werden alle in der Spalte "Element" konfigurierten Zeichen als Bestandteil das Elementnamens betrachtet (einschließlich der Zeichen in geschweiften Klammern).
- **Variablenname**: Geben Sie die Namen der mit den Serverelementen zu verknüpfenden Tags ein.
- **Element**: Gegen Sie den Namen des Serverelements ein. Nach dem Auswählen des OPC-Servers können Sie mit Hilfe des OPC-Browsers Elemente vom Server auswählen. Führen Sie einen Rechtsklick auf das Feld **Element** aus und wählen Sie die Option **OPC-Browser**.



In diesem Feld kann ein Tag-Name in geschweiften Klammern verwendet werden (z.B. **{TagName}**). Auf diese Weise können Sie den Elementnamen während der Laufzeit dynamisch ändern.

- **Scan**: Legt die Bedingungen fest, unter denen der Tag-Wert vom Remote-Gerät oder -Server gelesen und dann in der Projektdatenbank aktualisiert wird. Dabei gelten die folgenden Optionen:
  - **Immer** bedeutet, dass der Tag bei jedem Scan des Kommunikationsarbeitsblatts gelesen und aktualisiert wird, ungeachtet dessen, ob der Tag in anderen WinStudio-Bildern, Skripten oder Arbeitsblättern verwendet wird. Diese Option wird empfohlen für Tags, die kontinuierlich im Hintergrund überwacht werden müssen, wie z.B. Tags zum Auslösen vor Alarmen, in Rezepturen verwendete Tags, in der historischen Datenbank erfasste Tags usw.
  - **Bild** bedeutet, dass der Tag nur dann gelesen und aktualisiert wird, wenn er in mindestens einem geöffneten Bild verwendet wird, entweder lokal oder auf einer anderen Client-Station. Diese Option wird für Tags empfohlen, die ausschließlich in Bildern verwendet werden, da diese nicht immer aktualisiert werden müssen. Durch Auswahl dieser Option kann die Leistungsfähigkeit des Projekts verbessert werden.
  - **Auto** bedeutet, dass das Projekt automatisch entweder **Immer** oder **Bild** wählt, je nach dem, ob der Tag in Ihrem Projekt verwendet wird. Wenn der Tag ausschließlich in einem WinStudio-Bild verwendet wird, wird der Scan standardmäßig auf **Bild** zurückgesetzt. Wenn der Tag aber in anderen Arbeitsblättern verwendet wird (z.B. Skript, Berechnung, Alarm, Trend, Rezeptur, Bericht, Scheduler), wird zur Laufzeit der Scan auf **Immer** gesetzt..

  Wenn Sie nicht sicher sind, welche Option zu wählen ist, wählen Sie **Immer**. Dadurch wird sichergestellt, dass der Tag gelesen und aktualisiert wird.
- **Div**: Festlegen der Divisionskonstante, wenn eine Anpassung der Skalierung erforderlich ist. Dieser Wert ist ein Divisionsfaktor in einer Lese-Operation und ein Multiplikationsfaktor in einer Schreib-Operation.
- **Addieren**: Festlegen der Additionskonstante, wenn eine Anpassung der Skalierung erforderlich ist. Dieser Wert ist ein Additionsfaktor in einer Lese-Operation und ein Subtraktionsfaktor in einer Schreib-Operation.

Um den OPC-Client-Task zur Laufzeit des Projekts auszuführen, öffnen Sie die Task-Konfiguration (**Werkzeuge ▸ Status** im Kontextmenü des Ordners "WinStudio") und wählen Sie für "OPC-Client Runtime" die Starteinstellung "Automatisch". In diesem Dialog können Sie zum Testen den Task durch Klicken auf **Start** auch manuell starten. Nach Ausführung dieses Programms wird ein kleines Symbol im Infobereich angezeigt.

Zum Schließen des OPC-Client-Laufzeittasks führen sie einen Rechtsklick auf das Symbol im Infobereich aus und klicken auf **Beenden**.



"WinStudio" und CEView verfügen ebenfalls über einen OPC-Server-Kommunikationstask mit der Bezeichnung "Studio.Scada.OPC". Dieser Task startet automatisch, wenn ein OPC-Client (lokal oder fern) versucht, sich mit dem Server "Studio.Scada.OPC" zu verbinden. Ein OPC-Client kann mit Hilfe der OPC-Schnittstelle mit der Tag-Datenbank (Projekt-Tags, System-Tags und gemeinsame Tags) Daten austauschen.

Zusätzlich kann der OPC-Server-Task automatisch gestartet werden, wenn das Projekt ausgeführt wird. Wählen Sie in der Taskkonfiguration (Eintrag **Status** im Kontextmenü von WinStudio) den OPC-Server-Task aus, klicken Sie auf die Schaltfläche **Start** und wählen Sie **Automatisch**.

Kommunikation mit anderen Geräten

OPC Fehlerbeseitigung

Wenn Sie OPC verwenden und Probleme mit dem Kommunikationsaufbau auftreten, sollten Sie zunächst die Meldungen im LogWin überprüfen.

Wenn das Projekt auf einem Windows Embedded Zielsystem ausgeführt wird, kann das Protokoll auf zwei Arten geprüft werden:

1. Remote LogWin
2. Lokales Protokoll

Weitere Informationen über diese Protokolle unter Verwenden des LogWin-Tasks auf Seite 448.

Wenn Sie im Protokoll Fehlermeldungen finden, schlagen sie diese im Handbuch/Hilfesystem nach und befolgen Sie die angegebenen Schritte zur Problemlösung. (Drücken Sie <STRG>>+ <F> zum Suchen im Handbuch; zum Suchen im kontextabhängigen Hilfesystem verwenden Sie den Index.)

Wenn Sie bei Ihrem Händler technische Unterstützung anfordern möchten, stellen Sie diesem bitte die folgenden Informationen zur Verfügung:

1. Protokolldatei
2. Softwarehersteller und Produktname des verwendeten OPC Server/ Client
3. Falls möglich eine Kopie oder eine Testversion des OPC-Servers zu Testzwecken
4. Die Kontaktdaten für den technischen Support Ihres OPC Server/Client

Drei mögliche Fehler und deren Behebung sind im Folgenden aufgeführt…

Sicherheit

Fehlercode: **16#80070005 oder -2147024891**

Fehlerursache: Wenn der OPC-Client versucht, sich mit dem OPC-Server zu verbinden, verlangt die DCOM-Ebene in der Regel eine Authentifizierung. Der Rechner, auf dem der OPC-Server läuft, muss den auf dem OPC-Client-Rechner angemeldeten Benutzer erkennen und dieser Benutzer muss über Zugriffsrechte für den OPC-Server verfügen.

Lösung: Der erste Schritt ist das Erstellen eines Benutzers, der auf beiden Rechnern über Administratorrechte und das gleiche Passwort verfügt. Melden Sie sich auf beiden Rechnern mit diesem Benutzernamen an und versuchen Sie dann, die Verbindung aufzubauen.

Name auflösen

Fehler: **Konnte keine Verbindung mit AdviseSink herstellen, Fehler: -2147022986 (16#80070776)**

Fehlerursache: Es besteht ein Problem mit dem Auflösen des Rechnernamens.

Lösung: Dieses Problem kann behoben werden, indem die IP-Adresse des Servers anstelle des Rechnernamens angegeben wird.

Proxy für Windows CE

Fehler: **OPC-Server: IIndCP::Advise - Callback-Schnittstelle konnte nicht abgefragt werden: 16#80040155**

Fehlerursache: Bei Ihrem Windows Embedded Gerät fehlt die **OPCCOMN_PS.dll**.

Lösung: Sie sollten die .dll auf das Gerät herunterladen und registrieren. Die **.dll** sollte in Ihrer "WinStudio"-Ausgabe verfügbar sein, höchstwahrscheinlich in

[…]\WinStudio\Redist\Wince x.x\processor

Falls die **.dll** nicht im Ordner für Ihren Prozessor enthalten ist, wenden Sie sich bitte an Ihren "WinStudio"-Lieferanten.

11.4 Konfigurieren einer OPC-UA-Client-Verbindung mit einem OPC-UA-Server

OPC-UA-Treiberblätter werden zur Kommunikation mit allen Systemen verwendet, die das OPC-UA-Server-Protokoll implementieren.

Über OPC UA

Das neue OPC-UA-Protokoll (OPC-Unified-Architecture) wurde von der OPC Foundation eingeführt. Zur Einführung des neuen Protokoll gibt es von der OPC-Stiftung folgende Information:

" Die bestehenden Spezifikationen auf Basis von OPC COM waren der OPC-Gemeinschaft in den vergangenen zehn Jahren sehr nützlich, aber mit dem Fortschritt der Technik müssen sich auch unsere Standards für Interoperabilität verändern. Dies sind die Faktoren, die die Entscheidung zum Erstellen einer neuen Architektur beeinflusst haben: "

- Microsoft hat sich von COM weg bewegt und hin zu plattformübergreifenden Web Services und SOA (serviceorientierte Architektur)
- OPC-Hersteller wollen zum Offenlegen der OPC-Datenmodelle (DA, A&E, HDA...) eine einzige Servicegruppe
- OPC-Hersteller wollen OPC auf Systemen ohne Microsoft implementieren, einschließlich Embedded-Geräten
- Andere beteiligte Organisationen benötigen eine zuverlässige, effiziente Möglichkeit zum Bewegen von strukturierten Daten auf höherer Ebene

OPC UA ist somit als plattform- und sprachunabhängiges Protokoll konzipiert, das auch mit “klassischen” OPC-Systemen rückwärts-kompatibel ist. Weitere Informationen unter http://www.opcfoundation.org/UA/.

☞ Diese Funktion enthält kryptographische Software, geschrieben von Eric Young (eay@cryptsoft.com).

Konfigurieren einer neuen OPC-UA-Verbindung

Um mit einem Gerät mittels des OPC-UA-Servers zu kommunizieren muss die Verbindung zu diesem Server zunächst konfiguriert werden. Gehen Sie dazu wie folgt vor:

1. Öffnen Sie das Kontextmenü des Ordners "OPC-UA" und wählen Sie den Eintrag "OPC-UA-Verbindungen konfigurieren".

   Es erscheint ein Dialog, der alle bisher erstellten Konfigurationen anzeigt. Um eine neue Verbindung zu einem OPC-UA-Server zu erstellen, drücken Sie die Schaltfläche "Hinzufügen".

   Der Dialog "UA-Server-Konfiguration" wird angezeigt.

Kommunikation mit anderen Geräten

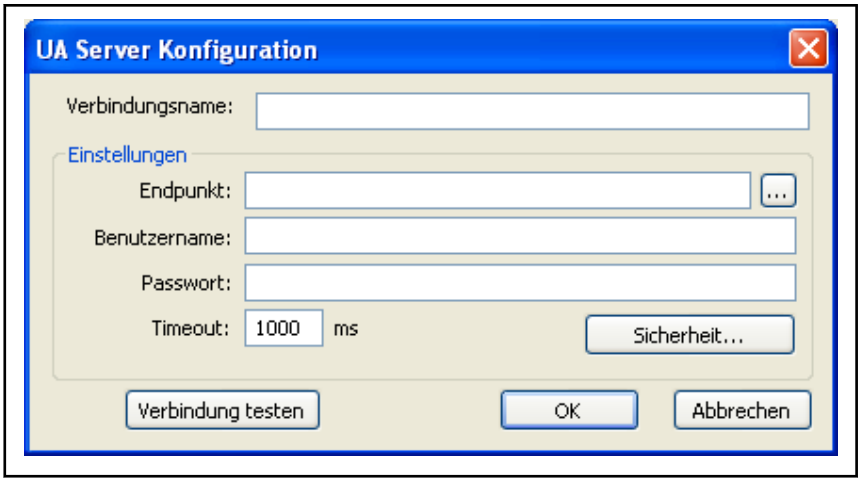


Abb.11-11: Dialog UA-Server-Konfiguration

2. Tragen Sie im Feld **Verbindungsname** einen Namen für die Verbindung ein. Dieser Name wird im Ordner OPC-UA-Verbindungen im Projektexplorer angezeigt und ist der Name, nach dem Sie beim Konfigurieren des Arbeitsblatts OPC-UA-Client suchen müssen.
3. Geben Sie im Feld **Endpunkt** die URL des OPC-UA-Servers ein, mit dem Sie sich verbinden möchten. Wenn Sie die URL nicht kennen, klicken Sie rechts neben dem Feld auf die Schaltfläche Suchen. Der Dialog "UA Discovery" wird angezeigt.

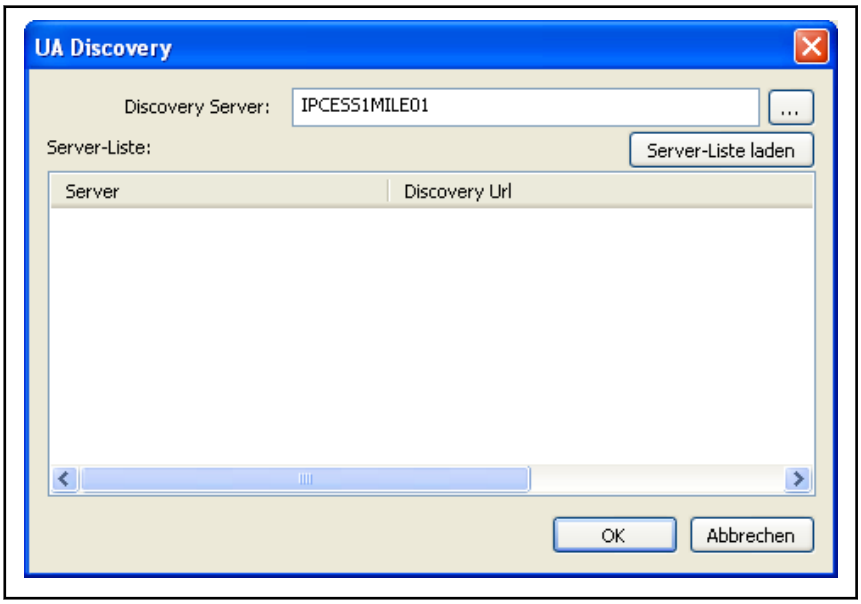


Abb.11-12: Dialog UA Discovery

Verwenden Sie diesen Dialog zum Auffinden des Discovery-Servers, der eine Liste mit OPC-UA-Servern im Netzwerk veröffentlicht, und wählen Sie dann den Server für Ihre Verbindung aus.

4. Tragen Sie Ihre Zugangsdaten für den OPC-UA-Server in die Felder **Benutzername** und **Passwort** ein.
5. Wenn der OPC-UA-Server für eine sichere Verbindung konfiguriert ist, müssen Sie zusätzliche Schritte ausführen, um das Serverzertifikat in Ihrem Projekt zu installieren und ein Client-Zertifikat zur Installation auf dem Server zu erzeugen:

5.1 Holen Sie das Server-Zertifikat und speichern Sie es im Unterordner "Konfig" Ihres Projektordners (z.B. **[…]\Eigene Dateien\WinStudio Projekte\ projekt_name\Config**).

Wie Sie das Server-Zertifikat erhalten, hängt vom jeweiligen Server ab. Bitte wenden Sie sich an den Serverhersteller.

5.2 Klicken Sie im Dialog "UA-Serverkonfiguration" auf **Sicherheit**. Der Dialog "OPC-UA-Sicherheit" wird angezeigt.

5.3 Wählen Sie das Serverzertifikat aus der Liste **Serverzertifikate**. (Hier sollten alle im Unterordner "Config" gespeicherten Zertifikate aufgeführt sein.)

5.4 Wählen Sie aus den Listen **Sicherheitsverfahren** und **Sicherheitsmodus Meldungen** die entsprechenden Optionen für die Konfiguration des OPC-UA-Servers. Bitte konsultieren Sie auch hierzu Ihren Serverhersteller.

5.5 Klicken Sie auf **Zertifikat erzeugen**. Das Client-Zertifikat wird erzeugt und Unterordner "Config" gespeichert.

5.6 Schließen Sie den Dialog "OPC-UA-Sicherheit" indem Sie auf **OK** klicken.

5.7 Installieren Sie das Client-Zertifikat auf dem OPC-UA-Server. Die genaue Vorgehensweise hängt vom jeweiligen Server ab. Bitte wenden Sie sich auch hierzu an den Serverhersteller.

Für weitere Informationen laden Sie bitte das White Paper The OPC UA Security Model for Administrators von der OPC Foundation herunter.

6. Klicken Sie auf **OK**. Die Verbindung wird im Ordner OPC-UA-Verbindungen im Projektexplorer gespeichert.

Konfigurieren eines neuen OPC-UA-Treiberblatts

Um ein neues Treiberblatt für den OPC-UA-Client zu erstellen gehen Sie folgendermaßen vor:

1. Öffnen Sie das Kontextmenü des Ordners "OPC-UA" und wählen Sie "Neues Treiberblatt".

   Ein neues Element wird im Ordner OPC-UA hinzugefügt. Öffnen Sie nun mit einem Doppelklick den Treiberblatteditor:

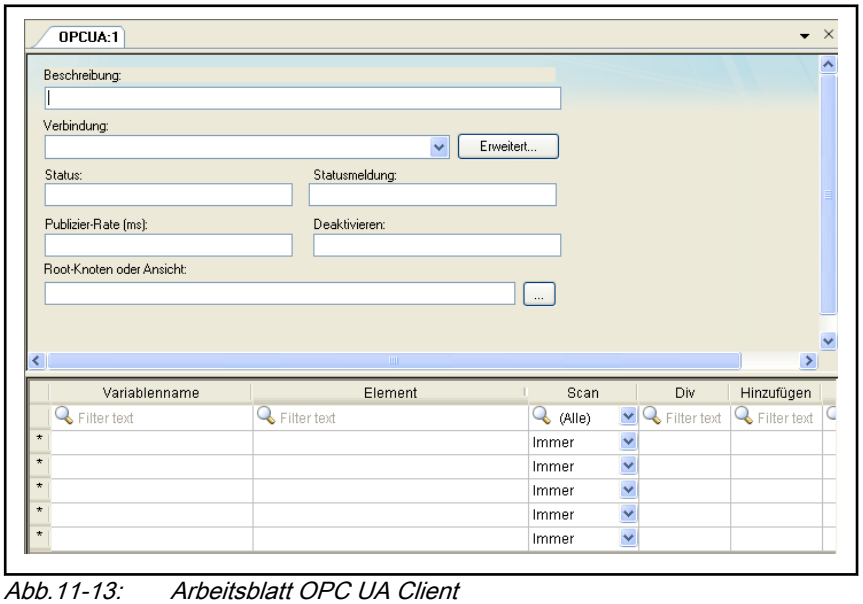


Abb. 11-13: Arbeitsblatt OPC UA Client

2. Tragen Sie im Feld **Beschreibung** eine Beschreibung für das Arbeitsblatt ein. Dies dient nur zu Dokumentationszwecken und beeinflusst nicht die Ausführung des Arbeitsblatts.
3. Wählen Sie aus der Liste **Verbindung** die zuvor konfigurierte Serververbindung.
4. Für weitere Verbindungsoptionen klicken Sie auf **Erweitert**. Der Dialog für weitere Einstellungen wird angezeigt:

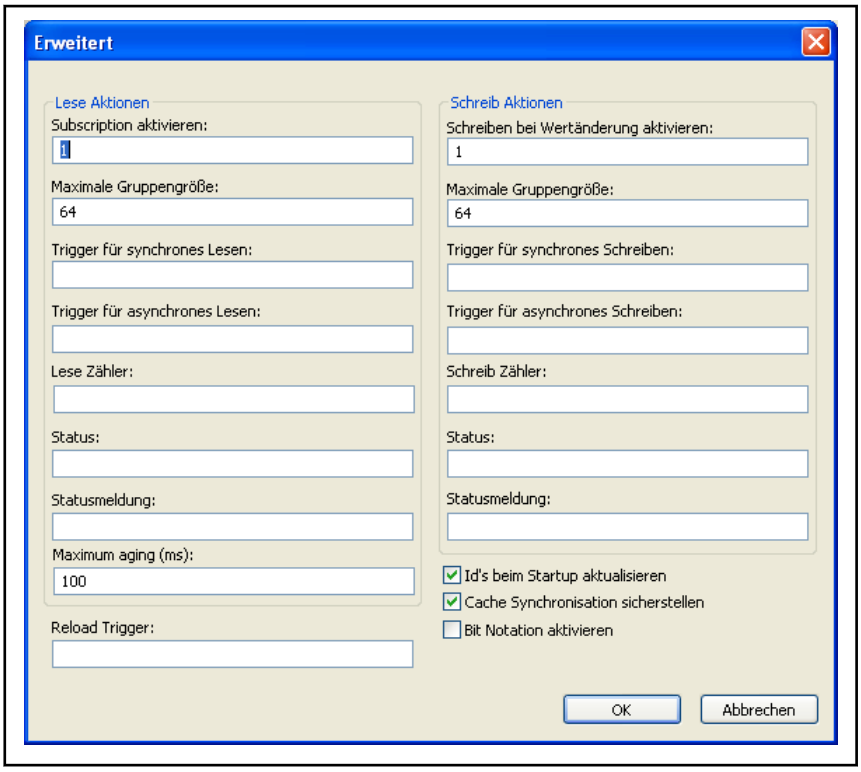


Abb.11-14: Arbeitsblatt OPC UA Client, weitere Einstellungen

Überprüfen Sie die Optionen und konfigurieren Sie nach Bedarf:

Bereich / Element		Beschreibung
Lese-Aktionen	Subskription aktivieren	Wenn dieser Wert WAHR (nicht Null) ist, fordert der Client kontinuierlich Aktualisierungen vom Server an. Diese Option ist standardmäßig aktiviert. Wenn sie deaktiviert wird, müssen Leseauslöser verwendet werden (siehe unten).
	Maximale Gruppengröße	Die maximale Anzahl an Tag-Lesevorgängen, die in einem einzelnen Lesevorgang ausgeführt werden dürfen. Wenn zum Beispiel 1000 Elemente/Zeilen im Arbeitsblatt konfiguriert sind und die **Maximale Gruppengröße** auf 100 eingestellt ist, dann werden bei jedem Scan des Arbeitsblatts 10 Lesevorgänge durchgeführt.
	Trigger für synchrones Lesen	Wenn sich der Wert dieses Tags/Ausdrucks ändert, wird das Arbeitsblatt gescannt und alle Tag-Werte werden vom Server gelesen. Das Projekt wartet vor dem Fortfahren auf das Beenden des Scans.
	Trigger für asynchrones Lesen	Wenn sich der Wert dieses Tags/Ausdrucks ändert, wird das Arbeitsblatt gescannt und alle Tag-Werte werden vom Server gelesen. Das Projekt läuft weiter, ohne auf das Beenden des Scans zu warten.
	Lese-Zähler	Der Name eines Tags (Typ Integer), der die Anzahl der seit Projektausführung durchgeführten Lesevorgänge erhält.
	Status	Der Name eines Tags (Typ Integer), der für den letzten von einem Auslöser durchgeführten Lesevorgang einen Statuscode erhält.
	Statusmeldung	Der Name eines Tags (Typ String), der die entsprechende Statusmeldung erhält.
	Maximales Alter	Das maximale Alter (in Millisekunden) von Werten, die vom Server-Cache akzeptiert werden. Ist ein Wert älter, so wird der Server gezwungen, den letzten Wert vom Zielgerät zu holen.
Schreib-Aktionen	Schreiben bei Wert-Änderung aktivieren	Wenn dieser Wert WAHR (nicht Null) ist, wird bei jeder Änderung des Projekt-Tags automatisch ein Schreibvorgang ausgeführt. Hinweis: Diese Option ist standardmäßig aktiviert. Wenn sie deaktiviert wird, müssen Schreibauslöser verwendet werden (siehe unten).
	Maximale Gruppengröße	Die maximale Anzahl an Tag-Schreibvorgängen, die in einem einzelnen Schreibvorgang ausgeführt werden dürfen. Wenn zum Beispiel 1000 Elemente/Zeilen im Arbeitsblatt konfiguriert sind und die **Maximale Gruppengröße** auf 100 eingestellt ist, dann werden bei jedem Scan des Arbeitsblatts 10 Schreibvorgänge durchgeführt.
	Trigger für synchrones Schreiben	Wenn sich der Wert dieses Tags/Ausdrucks ändert, wird das Arbeitsblatt gescannt und alle Tag-Werte werden zum Server geschrieben. Das Projekt wartet vor dem Fortfahren auf das Beenden des Scans.
	Trigger für asynchrones Schreiben	Wenn sich der Wert dieses Tags/Ausdrucks ändert, wird das Arbeitsblatt gescannt und alle Tag-Werte werden zum Server geschrieben. Das Projekt läuft weiter, ohne auf das Beenden des Scans zu warten.
	Schreib-Zähler	Der Name eines Tags (Typ Integer), der die Anzahl der seit Projektausführung durchgeführten Schreibvorgänge erhält.
	Status	Der Name eines Tags (Typ Integer), der für den letzten von einem Auslöser durchgeführten Schreibvorgang einen Statuscode erhält.
	Statusmeldung	Der Name eines Tags (Typ String), der die entsprechende Statusmeldung erhält.

Kommunikation mit anderen Geräten

Bereich / Element	Beschreibung
Trigger für erneutes Laden	Im Hauptteil dieses Arbeitsblatts konfigurierte indirekte Tags (z.B. **{MyTag}**) werden nur dann neu geladen, wenn sich der Wert dieses Tags/Ausdrucks ändert.
IDs beim Start aktualisieren	Durch Aktivieren dieser Option werden die Knoten-IDs im Arbeitsblatt bei jeder Ausführung des Projekts aus dem angegebenen Elementpfad aktualisiert. **Hinweis:** Durch dieses Aktualisieren der IDs dauerte das Starten des Projekts eventuell länger. Bei Aktivierung dieser Option sollte dann auch ein Wurzelknoten festgelegt werden (siehe unten), um zu begrenzen, wie viel von der Elementliste des Servers gescannt werden muss.

☞ Der OPC-Client von "WinStudio" verwendet normalerweise den im OPC-Protokoll festgelegten Trigger-Modus nicht. Statt dessen lässt es die Verwendung beliebiger Änderungen in einem beliebigen Tag/Ausdruck als Auslöser zu. Wenn Sie den Trigger-Modus verwenden möchten, konfigurieren Sie ein Arbeitsblatt zum Lesen der Triggerwerte und konfigurieren dann ein anderes Arbeitsblatt, das die Lesewerte als Triggerwerte festgelegt.

5. Klicken Sie zum Schließen des Dialogs "Erweitert" auf **OK** und kehren Sie zum Arbeitsblatt zurück.
6. Im Feld **Status** geben Sie den Namen eines Tags (vom Typ Integer) ein, der während der Projektlaufzeit Verbindungsstatus-Codes erhalten soll.
7. Im Feld **Status** geben Sie den Namen eines Tags (vom Typ String) ein, der die entsprechenden Statusmeldungen erhalten soll.
8. Im Feld **Publizier-Rate** geben Sie die Frequenz (in Millisekunden) ein, mit der der Client Aktualisierungen vom Server anfragt.
9. Tragen Sie im Feld **Deaktivieren** einen Tag/Ausdruck ein. Wenn der Wert WAHR (nicht Null) ist, wird das Arbeitsblatt nicht ausgeführt.
10. Legen Sie im Feld **Wurzelknoten oder Ansicht** den Serverknoten fest, der für alle Elemente im Hauptteil des Arbeitsblatts als Wurzel dient. Das Angeben eines Wurzelknotens erleichtert das Auffinden von Elementen und verbessert die Laufzeitleistung.
11. Legen Sie für jede Zeile im Hauptteil des Arbeitsblatts die folgenden Angaben fest:
    - **Tag-Name**: Der Name eines Tags. Sie können indirekte Tags verwenden (z.B. **{MyTag}**), dann muss aber auf jeden Fall die oben genannte Option **Trigger für erneutes Laden** konfiguriert werden.
    - **Item**: Das Serverelement, mit dem der Tag verbunden werden soll. Zum Durchsuchen der Elementliste des Servers führen Sie einen Rechtsklick auf dieses Feld aus und klicken dann auf das Kontextmenü **Suchen**.
    - **Scan** : Bei **Immer** wird die Zeile während der Laufzeit kontinuierlich gescannt. Bei **Bild** wird die Zeile nur gescannt, wenn ein WinStudio-Bild, das den angegebenen Tag verwendet, geöffnet ist.
    - **Div**: Zum Skalieren – der Wert wird beim Lesen vom Server durch diese Zahl geteilt und beim Schreiben zum Server mit dieser Zahl multipliziert.

- **Add**: Zum Skalieren – diese Zahl wird beim Lesen vom Server zu dem Wert addiert und beim Schreiben zum Server von dem Wert subtrahiert.
- **Knoten-ID**: Die Knoten-ID wird automatisch aus dem vollständigen Pfad des Serverelements erzeugt.

12. Speichern und schließen Sie das Arbeitsblatt.

Aktivieren des OPC-UA-Client-Tasks

Zum Aktivieren des OPC-UA-Client-Tasks für die Laufzeit:

1. Öffnen Sie die Task-Konfiguration über **WinStudio** ▸ **Werkzeuge** ▸ **Status**. Der Dialog "Ausführung Tasks" wird angezeigt.
2. Wählen Sie **OPC-UA-Client-Laufzeit** aus der Taskliste.
3. Klicken Sie auf **Start**.

   Der Dialog "Start" wird angezeigt.
4. Wählen Sie **Automatisch** und klicken Sie dann auf **OK**.
5. Schließen Sie den Dialog "Ausführung Tasks", indem Sie auf **OK** klicken.

11.5 Konfigurieren einer TCP/IP-Client-Verbindung zu einem anderen Projekt

TCP/IP-Client-Task/-Arbeitsblätter werden für die Kommunikation mit anderen "WinStudio"-Projektservern verwendet.

Die TCP/IP-Client- und -Server-Tasks ermöglichen die Synchronisierung der Datenbanken von zwei oder mehr Projekten.

Diese Tasks müssen das TCP/IP-Protokoll verwenden, um die Kommunikation zwischen Projekten bereitzustellen.

Vor der Verwendung der TCP/IP-Client- und -Server-Tasks muss sichergestellt werden, dass die Kommunikation über TCP/IP (Ethernet) korrekt konfiguriert ist und auf beiden Servern läuft.

- **Zum Konfigurieren der Server**: Auf dem Server selbst muss nichts konfiguriert werden. Es muss nur der TCP/IP-Server-Task von "WinStudio" ausgeführt werden. Dieser kann entweder beim automatisch beim Start der Laufzeit oder in der Taskkonfiguration (entweder lokal oder fern) manuell ausgeführt werden. Nach Ausführung dieses Programms wird ein kleines Symbol im Infobereich angezeigt.
- Zum Schließen des TCP/IP-Server-Tasks von "WinStudio": Führen Sie einen Rechtsklick auf das **TCP/IP-Server**-Symbol im Infobereich aus und klicken Sie auf **Beenden**.
- **Zum Konfigurieren der Clients**: Zum Festlegen der Server-IP-Adresse und der Tags, die mit dem Server auf dem Client-System gemeinsam verwendet werden sollen, müssen Sie "TCP/IP-Arbeitsblätter" verwenden. Diese können Sie im Ordner **WinStudio** ▸ **Variablen** ▸ **Kommunikation** ▸ **TCP/IP** erstellen.

Kommunikation mit anderen Geräten

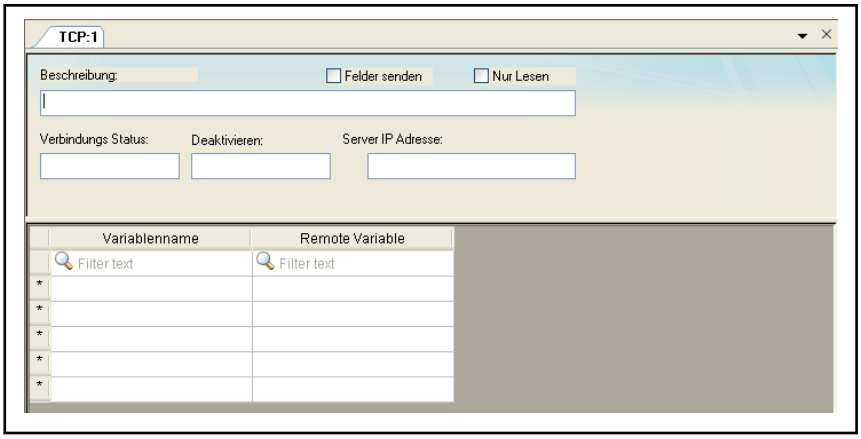


Abb.11-15: Konfiguration eines TCP/IP-Treiberblatts

Verwenden Sie zum Konfigurieren der TCP/IP-Client-Kommunikation die folgenden Parameter:

- **Beschreibung**: Geben Sie zu Dokumentationszwecken eine Beschreibung für das TCP/IP-Treiberblatt ein. Der TCP/IP-Client-Task ignoriert diese Information.
- **Verbindungsstatus**: Geben Sie einen Tag-Namen ein und der TCP/IP-Client-Konfigurationstask aktualisiert diesen Tag in Übereinstimmung mit dessen Verbindungsstatus. Ein Tag-Wert Null zeigt an, dass die Verbindung in Ordnung ist. Ein anderer Tag-Wert zeigt an, dass von der Bibliothek "Windows Socket" ein Fehlercode zurückgegeben wurde.
- **Deaktivieren**: Geben Sie in diesem Feld einen Tag-Namen ein. Wenn ein Tag einen anderen Wert als 0 hat, wird dieses TCP/IP-Arbeitsblatt deaktiviert. Bei Verwendung dieses Felds kann das TCP/IP-Client-Arbeitsblatt während der Laufzeit aktiviert/deaktiviert werden.
- **Server-IP-Adresse**: Geben Sie die IP-Adresse und den Port (optional) des Zielservers ein – z.B. "169.254.182.158:123". Der Port sollte für Client- und Serverstation gleich sein.

  Es kann auch ein String-Tag in geschweiften Klammern angegeben werden (z.B. `{tagname}`), wenn diese Adresse während der Laufzeit dynamisch verändert werden soll.
- **Felder senden**:
  - Durch "Deaktivieren" dieses Kontrollkästchen tauschen die TCP/IP-Client/Server-Tasks "nur" die Tag-Werte, deren Zeitstempel und Qualität aus.
  - Durch "Aktivieren" dieses Kontrollkästchens tauschen die TCP/IP-Client/Server-Tasks "auch" die Tag-Felder Min, Max, Best, Einheit, LoLoLimit, LoLimit, HiLimit, HiHiLimit, RateLimit, AbwSollwert, AbwpLimit und AbwenLimit aus.

☞ Es ist möglich, der TCP/IP-Kommunikation weitere Felder hinzuzufügen oder einzelne Felder zu deaktivieren. Weitere Informationen erhalten Sie vom jeweiligen Hersteller.

- **Nur Lesen**: Wenn diese Option ausgewählt ist, findet die gesamte Kommunikation nur in einer Richtung statt und es werden keine Tag-Werte an den angegebenen Server zurückgeschrieben. Dies ist hilfreich, wenn der TCP/IP Client nur zum Lesen von Daten aus anderen Projekten verwendet wird. Die Aktivierung dieser Option kann die Sicherheit und Stabilität während der Laufzeit erheblich verbessern.

- **Variablenname**: Geben Sie die Tags ein, die Sie mit dem Server gemeinsam verwenden wollen. Wenn der Tag ein Array oder eine Klasse ist (oder beides), aktiviert das Projekt standardmäßig automatisch alle Array- und Klassen-Elemente für die TCP/IP-Kommunikation. Zum Konfigurieren eines bestimmten Array- und/oder eines bestimmten Klassen-Elements geben Sie die Array-Index bzw. das Klassen-Element in eckigen Klammern ein, gefolgt vom Tag-Namen. Zum Beispiel "ebene[3].member".
- **Remote-Variable** ("optional"): Geben Sie den Namen eines Tags ein, der mit dem im Feld **Variablenname** festgelegten Tag verknüpft werden soll. Wenn Sie dieses Feld leer lassen, verwendet das Projekt den gleichen Tag-Namen wie im Client und im Server.

☞ Wenn Sie ein Array gemeinsam verwenden müssen, sollte der Tag im Server die gleiche Anzahl an Elementen enthalten wie der Tag im Client. Wenn der Tag eine Klasse ist, sollte die Klassendefinition sowohl in den Server- als auch in den Client-Programmen identisch sein. Nichtbeachtung dieser Regeln kann zu nicht voraussagbaren Ergebnissen führen.

Der TCP/IP-Client-Task kann entweder automatisch beim Start der Laufzeitumgebung oder manuell in der Task-Konfiguration (**WinStudio ▸ Werkzeuge ▸ Status**). Nach dem Start des TCP/IP-Tasks wird ein kleines Symbol im Infobereich angezeigt.

Nur der Client verwendet den Parameter "ConnectionRetryTimeout".

11.6 Konfigurieren einer DDE-Client-Verbindung mit einem DDE-Server

DDE-Treiberblätter werden zum Konfigurieren einer DDE-Client-Verbindung zu einer DDE-Server-Anwendung wie z.B. Microsoft Excel (oder ein anderes Windows-Programm, das diese Schnittstelle unterstützt) verwendet.

DDE (Dynamic Data Exchange) ist ein Protokoll, mit dem Daten dynamisch zwischen Windows-Anwendungen wie Excel ausgetauscht werden können. Bei einer DDE-Kommunikation handelt es sich um eine Interaktion zwischen Server- und Client-Programmen. "WinStudio" stellt Schnittstellen zur Verfügung, die als Client oder als Server ausgeführt werden können. Siehe DDE-Client-Laufzeit und DDE-Server in den Laufzeittasks (siehe **Werkzeuge ▸ Status** im Kontextmenü des Ordners "WinStudio".

- Zum Ausführen als Server wird der DDE- oder NetDDE-Server-Task wie in "Laufzeit-Tasks" beschrieben gestartet.
- Zum Ausführen als Client wird des DDE-Treiberblatt im Ordner **WinStudio ▸ Variablen ▸ Kommunikation ▸ DDE** konfiguriert.

Network Dynamic Data Exchange (NetDDE) ist eine Erweiterung von DDE, die rechnerübergreifend auf einem Netzwerk funktioniert.

- Um "WinStudio" als Server für eine NetDDE-Verbindung auszuführen, muss die DDE-Serveranwendung gestartet werden.
- Um "WinStudio" als Client für eine NetDDE-Verbindung auszuführen, werden die gleichen DDE-Treiberblätter mit der korrekten Konfiguration für das Adressieren eines "WinStudio"-Projekts verwendet.

☞ Beim Ausführen von NetDDE akzeptiert "WinStudio" nur die SCHREIB-Trigger. Zum Lesen von Daten muss auf dem Serverrechner ein Schreibbefehl konfiguriert werden.

Zum Erstellen eines neuen "DDE"-Arbeitsblatts wählen Sie "Neues Treiberblatt " im Kontextmenü des Ordners "DDE". Anschließend können Sie eine Doppelklick auf das neu hinzugefügte Element ausführen, um den Treiberblatteditor zu öffnen.

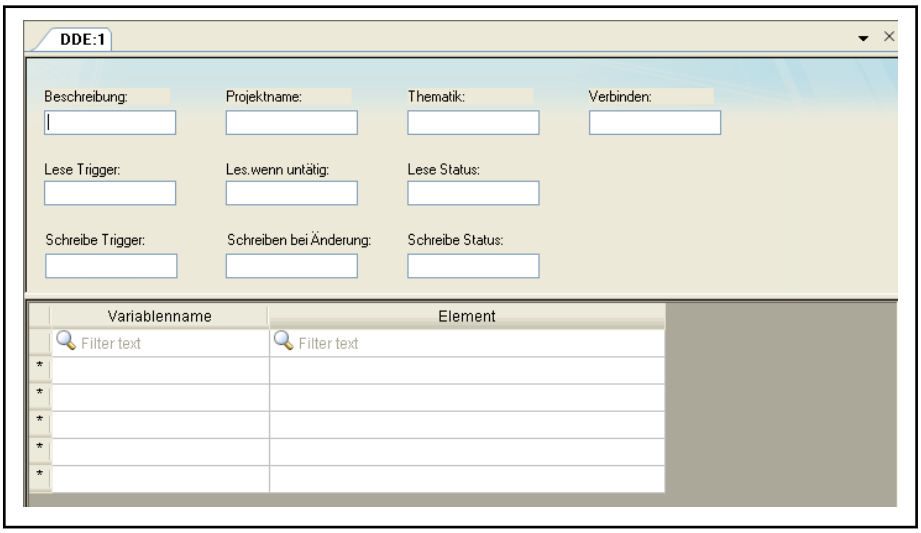


Abb.11-16: DDE-Arbeitsblatt

Das Arbeitsblatt "DDE" gliedert sich in zwei Bereiche:

- "Titelbereich" (oberer Bereich), enthält Informationen für die gesamte Gruppe und definiert die Tags zum Auslösen den Lese- und Schreib-Vorgangs sowie zum Empfangen des Verbindungsstatus.
- Der "Hauptteil" (unterer Bereich) dient zum Definieren von Tags im Projekt und Elementen in Bezug auf die DDE-Serveranwendung.

Jede DDE-Schnittstelle basiert auf der Adressierung einer Applikation mit Hilfe der folgenden drei Strukturen:

- Projektname
- Topic
- Element

Diese Informationen müssen vor der Konfiguration des DDE-Treiberblatts in der DDE-Serveranwendung ermittelt werden.

Die Parameter im Titelbereich des DDE-Client-Arbeitsblatts werden wie folgt verwendet:

- **Beschreibung**: Geben Sie zu Dokumentationszwecken eine Beschreibung für das DDE-Arbeitsblatt ein.
- **Projektname**: Geben Sie den Applikationsnamen für den DDE-Server ein.
- **Thematik**: Legen Sie ein Topic in der Serveranwendung fest.
- **Verbinden**: Geben Sie einen Tag zur Steuerung der Verbindung des "WinStudio"-DDE-Clients und der DDE-Server-Anwendung ein. Wenn der Tag auf 1 gesetzt wird, fordert er eine Verbindung mit dem Server an. Wenn die Verbindung nicht möglich ist oder nicht zustande kommt, setzt "WinStudio" den Wert auf 0 zurück. Wenn die Verbindung in Ordnung ist, bleibt der Wert auf 1.
- **Lese-Trigger**: Geben Sie einen Tag als Lesebefehl für die Tabelle ein. Wenn dieser Tag seinen Wert ändert, ruft "WinStudio" die Variablenwerte vom DDE-Server ab. Dieser Parameter kann nur mit lokalem DDE verwendet werden und nicht mit NetDDE-Servern.

- **Lesen wenn untätig**: Geben Sie einen Tag-Wert größer Null ein, um Lesen vom Gerät freizugeben.
- **Lese-Status**: Enthält den Status des Lesebefehls.
- **Schreib-Trigger**: Geben Sie einen Tag ein, der den Schreibbefehl auslöst.
- **Schreiben bei Tag-Änderung**: Geben Sie einen Tag-Wert größer Null ein, um den Kommunikationstreiber für die kontinuierliche Überprüfung auf Veränderungen von Tag-Werten in diesem Arbeitsblatt freizugeben. Wenn der Treiber eine Änderung feststellt, schreibt er den veränderten Tag zusammen mit der Tag-Adresse zum Gerät.
- **Schreib-Status**: Enthält den Status des Schreibbefehls.

Die Parameter im Hauptteil des DDE Client werden wie folgt verwendet:

- **Variablenname**: Geben Sie einen Tag zum Lesen oder Schreiben der "WinStudio"-Datenbank von der DDE-Serveranwendung ein.
- **Element**: Geben Sie den ITEM-Bestandteil der DDE-Adresse auf dem Server ein. Informationen zur genauen Syntax für **APP**, **TOPIC** und **ITEM** finden Sie in der Dokumentation für Ihre Serversoftware.

  Die Felder **Thema** und **Element** können mit Tags in der Adresse konfiguriert werden, gemäß der folgenden Syntax:

  `text{tag}`

  "WinStudio" wertet den Wert von **{tag}** in einen String aus und verwendet diesen in der Adresse. Zum Beispiel:

  - `Topic: topic_{tag_topic_name}_beispiel`
  - `Item: {tag_item_name}` oder

    `A{tag_nummer}`

Das Konfigurieren einer NetDDE-Verbindung ist ähnlich der Konfiguration einer DDE-Verbindung, mit Ausnahme von "Name" und "Topic" im Hauptteil der Anwendung. Bevor Sie mit den Tests beginnen, sollten Sie überprüfen, dass der DDE-Server auf der für den Datenaustausch vorgesehenen Station freigegeben ist.

☞ Für das Verbinden mit anderen Servern als "WinStudio" entnehmen Sie die Informationen zur korrekten Syntax für **APP**, **TOPIC** und **ITEM** bitte der Serverdokumentation.

Verwenden Sie die Titelbereichparameter im NetDDE-Client-Arbeitsblatt wie folgt zum Definieren von Tags für Lese- und Schreibbeginn und Tags, die den Verbindungsstatus erhalten:

- **Projektname**: Geben Sie **rechner_name\NDDE$** ein, wobei **rechner_name** ein gültiger Name eines Netzwerkrechners sein muss.
- **Thema**: Geben Sie den Topic-Namen **UNISOFT$** ein, um mit einer anderen "WinStudio"-Station zu verbinden.

Verwenden Sie die Hauptteilparameter des DDE-Client-Arbeitsblatts, um jeden Tag wie folgt mit jedem ITEM-Teil der DDE-Serveradresse zu verbinden:

- **Variablenname**: Geben Sie den Tag-Namen der lokalen "WinStudio"-Datenbank ein, der sich auf einen Remote-Tag-Namen bezieht.
- **Element**: Geben Sie den Remote-Tag-Namen ein, der sich auf den lokalen Tag-Namen bezieht.

Kommunikation mit anderen Geräten

☞ Standardmäßig unterstützt das DDE-Client-Modul DDE-Server, die String-Daten im UNICODE-Format handhaben. Wenn der DDE-Server String-Daten im ASCII-ANSI-Format handhabt, müssen die folgenden Einstellungen manuell in der Datei **projekt_name.app** konfiguriert werden (zum Bearbeiten dieser Datei kann Notepad verwendet werden):

[Optionen]

DDEANSI=1

12 Projektsicherheit

12.1 Projektsicherheit, Allgemeines

WinStudio enthält ein Projektsicherheitssystem zur Verwaltung des Zugriffs auf das Projekt durch Benutzer und Benutzergruppen, sowohl während der Entwicklung als auch während der Laufzeit.

☞ Das Sicherheitssystem von WInStudio ist nur verfügbar, wenn in den Eigenschaften des Visualisierungsgeräts der **Applikationstyp** die Option "WinStudio-Anwendung" ausgewählt ist. Bei anderen Applikationstypen erfolgt der Schutz von Funktionen während der Laufzeit über die IndraWorks-Funktion "Zugriffsschutz".

12.2 Sicherheitsmodi

Zusätzlich zur lokalen Verwaltung von Benutzern und Benutzergruppen als Ganzes in einem Einzelprojekt können außerdem vordefinierte Benutzer und Gruppen aus anderen "WinStudio"-Projekten oder von einem LDAP-konformen Domain-Server übernommen werden.

"WinStudio" unterstützt vier Sicherheitsmodi:

Nur lokal

Dies ist der Standardmodus für die meisten Projekte: Benutzer und Gruppen werden in der Entwicklungsumgebung erstellt und sind nur für das Projekt gültig, für das sie erstellt wurden.

Verteilt – Server

Dieser Modus ist ähnlich wie **Nur lokal**, aber hier wird die Konfiguration des Projektsicherheitssystems auch für andere "WinStudio"-Projekte im gleichen Netzwerk verfügbar gemacht (wobei diese auf **Verteilt – Client** eingestellt sind).

Außerdem kann bei Verlust der Sicherheitssystemkonfiguration des Projekts diese Konfiguration von einem der Client-Projekte zurückimportiert werden.

Verteilt – Client

Wenn dieser Modus gewählt ist, erhält das Projekt seine gesamte Sicherheitssystemkonfiguration von einem anderen "WinStudio"-Projekt im gleichen Netzwerk (das auf **Verteilt - Server** eingestellt ist).

Das Projekt speicherte diese Konfiguration und kann weiterhin ausgeführt werden, auch wenn die Kommunikation mit dem Serverprojekt unterbrochen wird.

Domäne (LDAP)

Das Lightweight Directory Access Protocol (LDAP) ist ein anerkannter Standard für die Verwaltung von Benutzern und Gruppen über viele verschiedene Anwendungen in einem Netzwerk hinweg.

Wenn dieser Modus gewählt wird, erhält das Projekt seine Benutzer und Gruppen von einem LDAP-konformen Domain-Server wie zum Beispiel Microsoft Active Directory für Windows oder OpenLDAP für Linux. Allerdings werden nur Benutzernamen, Passwörter und Gruppenzuordnungen von der Domäne übernommen, spezielle Berechtigungen müssen immer noch innerhalb des Projekts konfiguriert werden.

12.3 Sicherheitszugriffsebenen

Fast allen Elementen eines Projekts – Bildobjekte, Objektanimationen, Projektbilder, Task-Arbeitsblätter – kann eine Sicherheitszugriffsebene zugewiesen werden. Diese Zugriffsebene (Access level) bestimmt, welche Benutzergruppen das Element während der Entwicklung bearbeiten bzw. während der Laufzeit verwenden können.

Es gibt 255 mögliche Zugriffsebenen, was eine sehr fein abgestufte Sicherheitszuweisung ermöglicht. Jede Benutzergruppe ist mit Bereichen von Ebenen sowohl für Entwicklung als auch für Laufzeit konfiguriert und die Bereiche der Gruppen können sich überschneiden.

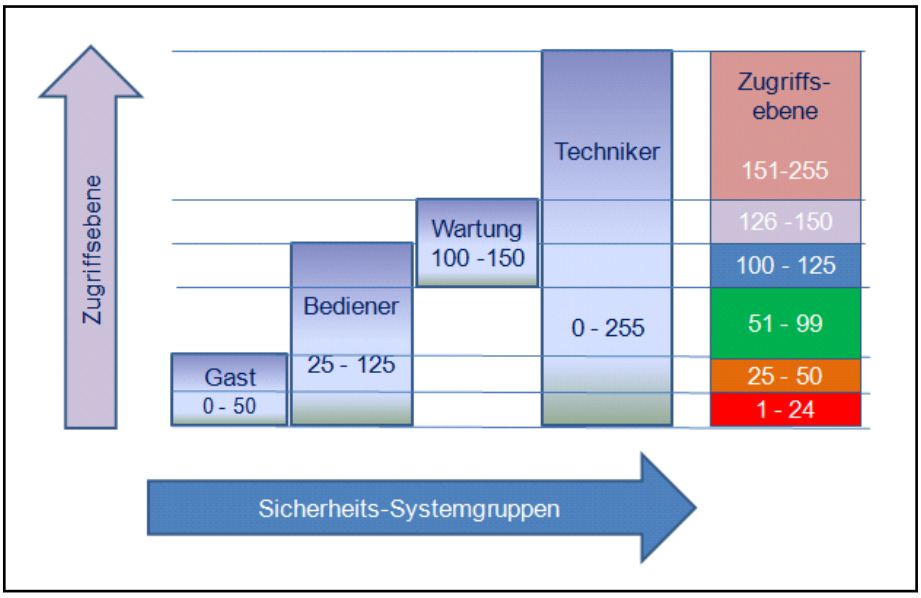


Abb.12-1: Beispiel für Sicherheitszugriffsebenen

Das bedeutet, dass die Zugriffsebene für ein Element innerhalb des für die Gruppe eines Benutzers angegebenen Bereichs liegen muss, damit dieser Benutzer das Element bearbeiten bzw. verwenden kann.

Zum Beispiel: **BenutzerA** aus **GruppeA** verfügt über einen Sicherheitszugriffsbereich von "1-10" und **BenutzerB** aus **GruppeB** verfügen über einen Sicherheitszugriffsbereich von "5-15". Des weiteren enthält das Projekt folgende Elemente:

- Element Nr. 1 hat Zugriffsebene = 1
- Element Nr. 2 hat Zugriffsebene = 7
- Element Nr. 3 hat Zugriffsebene = 12
- Element Nr. 4 hat Zugriffsebene = 20

Hierbei ergeben sich folgende für die Elemente folgende Zugriffsberechtigungen:

- Nur **BenutzerA** hat Zugriff auf Element Nr. 1
- Beide Benutzer haben Zugriff auf Element Nr. 2
- Nur **BenutzerB** hat Zugriff auf Element Nr. 3
- Keiner der Benutzer hat Zugriff auf Element Nr. 4



Die Standardzugriffebene für alle Elemente ist 0 und auf dieser Ebene können alle Benutzer alle Elemente verwenden.

Festlegen von Zugriffsebenen für verschiedene Arten von Elementen

Für WinStudio-Bilder kann die Zugriffsebene im Dialog Bildattribute auf Seite 170 festgelegt werden. Sicherheitseinstellung (Zugriffsebene) im Dialog "Bildattribute"

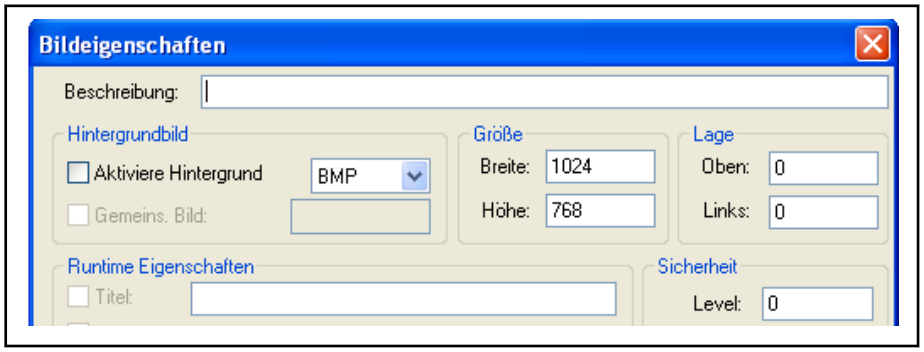


Abb. 12-2: Sicherheitseinstellung (Zugriffsebene) im Dialog Bildattribute

Für Bildobjekte und Objektanimationen kann die Zugriffsebene im jeweiligen Dialog Objekteigenschaften auf Seite 178 festgelegt werden. (Wenn die Option nicht im Hauptdialog verfügbar ist, sollte sie in einem der Unterdialoge verfügbar sein.) Sicherheitseinstellung (Zugriffsebene) im Dialog "Objekteigenschaften"

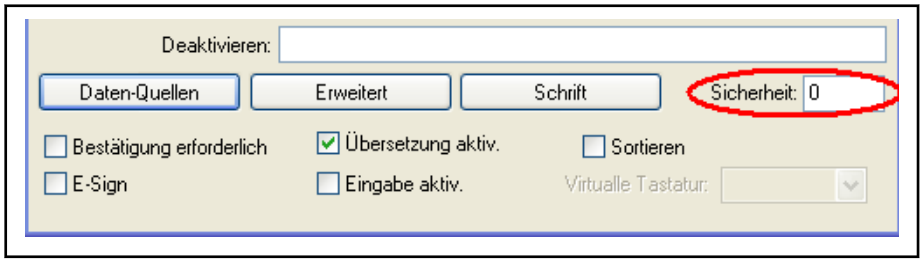


Abb. 12-3: Sicherheitseinstellung (Zugriffsebene) im Dialog Objekteigenschaften

Für Task-Arbeitsblätter gehen Sie wie folgt vor:

1. Öffnen Sie das Arbeitsblatt zum Bearbeiten.
2. Klicken Sie auf eine beliebige Stelle im Hauptteil des Arbeitsblatts. Der Eintrag **WinStudio-<Arbeitsblatt>** ▸ **Sicherheitsstufe ändern** im Hauptmenü von IndraWorks wird aktiviert.
3. Klicken Sie auf **Sicherheitsstufe ändern**. Der Dialog "Sicherheit" wird angezeigt.

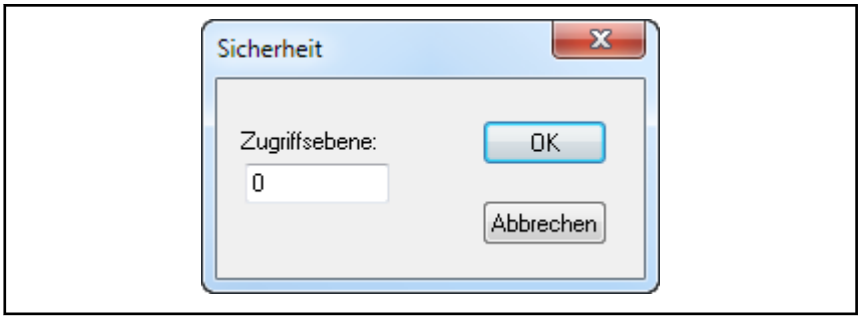


Abb. 12-4: Dialog Sicherheit

4. Geben Sie im Feld **Zugriffsebene** eine Zugriffsebene für das Bearbeiten des Arbeitsblatts ein.
5. Klicken Sie auf **OK**.

12.4 Verwenden des Konfigurationsassistenten für das Sicherheitssystem

Der Konfigurationsassistent für das Sicherheitssystem führt Sie durch die Schritte zum Konfigurieren des Sicherheitssystems des Projekts. Um den Konfigurationsassistenten zu starten, wählen Sie den Eintrag **Werkzeuge ▸ Sicherheitseinstellungen** im Kontextmenü des Ordners **WinStudio** aus.

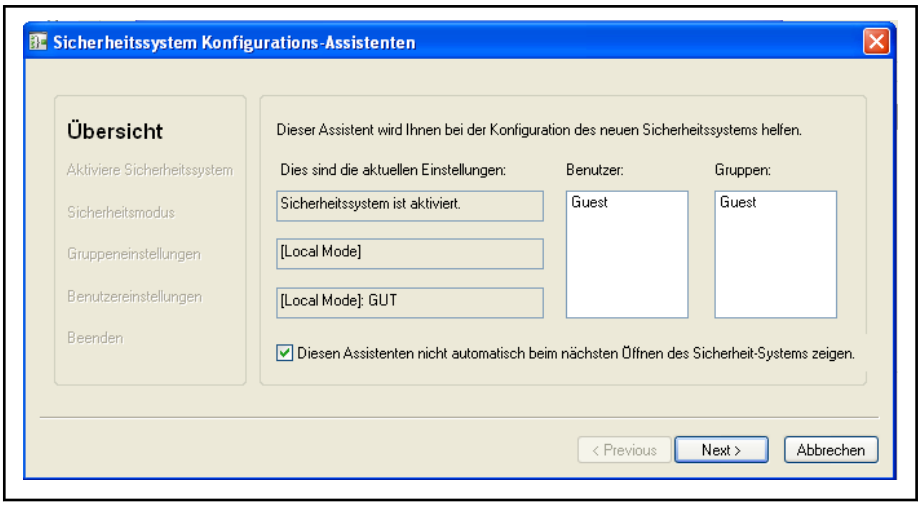


Abb. 12-5: Konfigurationsassistent für das Sicherheitssystem

1. Die erste Seite des Assistenten wird angezeigt. Diese Seite zeigt immer die aktuelle Konfiguration des Sicherheitssystems.
2. Klicken Sie auf **Weiter**. Die zweite Seite des Assistenten wird angezeigt.

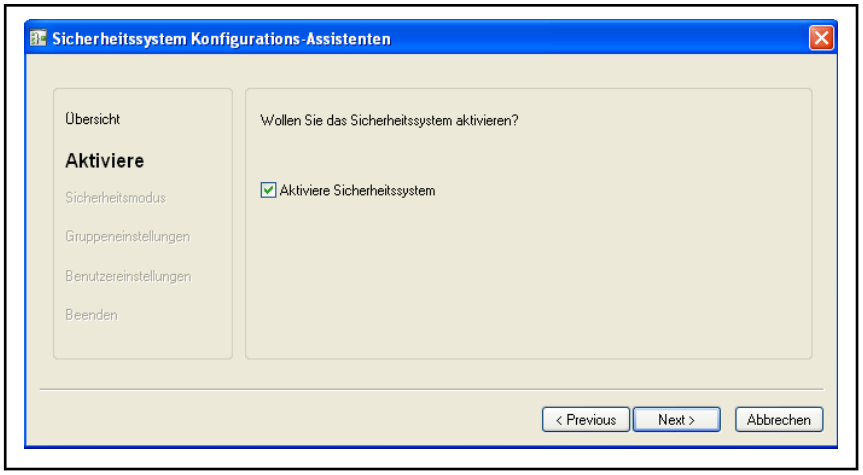


Abb. 12-6: Konfigurationsassistent für das Sicherheitssystem (2)

Wählen Sie **Sicherheitssystem aktivieren**.

3. Klicken Sie auf **Weiter**. Die dritte Seite des Assistenten wird angezeigt.

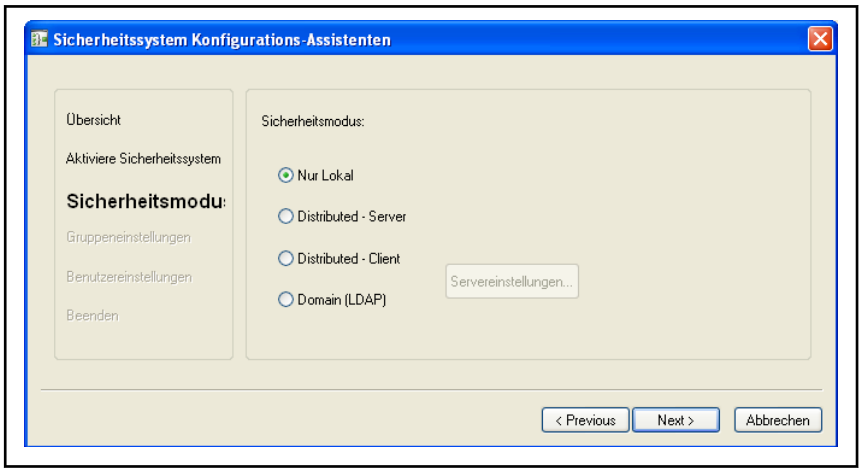


Abb. 12-7: Konfigurationsassistent für das Sicherheitssystem (3)

Wählen Sie den Sicherheitsmodus auf Seite 407 aus.

Für **Verteilt – Client** und **Domäne (LDAP)** klicken Sie auf **Servereinstellungen** und konfigurieren dann die benötigten Einstellungen auf Seite 412.

4. Klicken Sie auf **Weiter**. Die vierte Seite des Assistenten wird angezeigt.

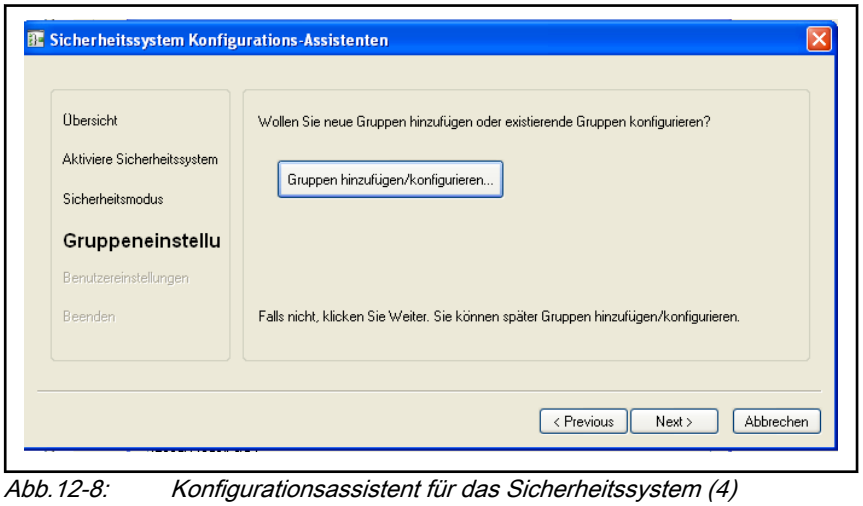


Abb. 12-8: Konfigurationsassistent für das Sicherheitssystem (4)

Wenn Gruppen erstellt oder konfiguriert werden müssen, klicken Sie auf **Gruppen erstellen/konfigurieren**. Der Dialog Gruppenkonto auf Seite 423 wird angezeigt. Wenn Sie diesen Dialog beenden, kehren Sie automatisch zum Assistenten zurück.

5. Klicken Sie auf **Weiter**. Die fünfte Seite des Assistenten wird angezeigt.

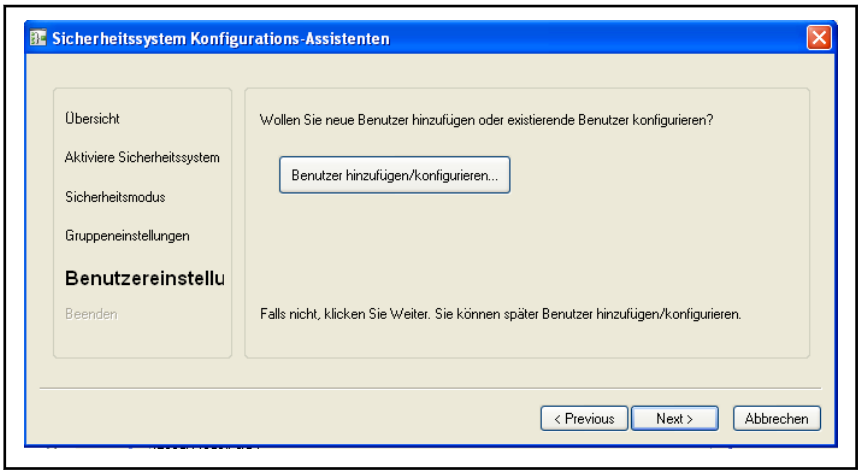


Abb. 12-9: Konfigurationsassistent für das Sicherheitssystem (5)

Wenn Benutzer erstellt oder konfiguriert werden müssen, klicken Sie auf **Benutzer erstellen/konfigurieren**. Der Dialog Benutzerkonto auf Seite 428 wird angezeigt. Wenn Sie diesen Dialog beenden, kehren Sie automatisch zum Assistenten zurück.

6. Klicken Sie auf **Weiter**. Die sechste Seite des Assistenten wird angezeigt.

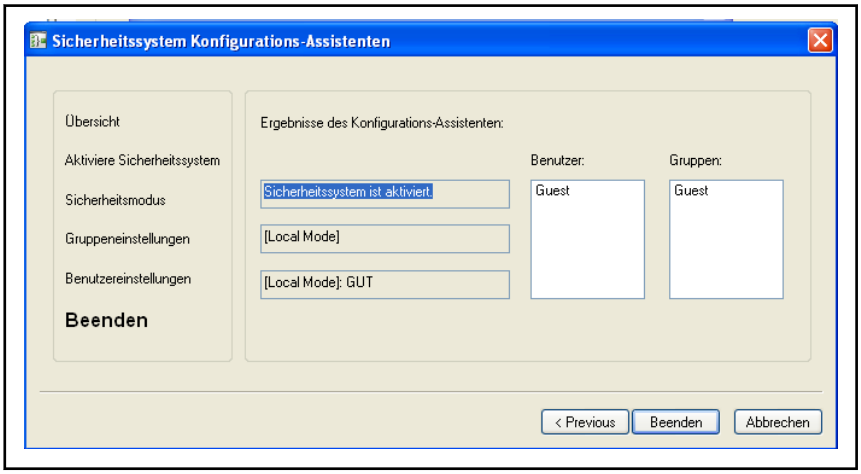


Abb. 12-10: Konfigurationsassistent für das Sicherheitssystem (6)

7. Überprüfen Sie Ihre Konfiguration und klicken Sie zum Schließen des Assistenten auf **Beenden**.

12.5 Konfigurieren der Servereinstellungen für Sicherheitsmodi

12.5.1 Konfigurieren der Servereinstellungen für Sicherheitsmodi, Allgemeines

Wenn der Sicherheitsmodus für Ihr Projekt auf Verteilt – Client oder Domäne (LDAP) eingestellt ist, müssen außerdem die Servereinstellungen konfiguriert werden.

Durch Klicken auf **Servereinstellungen** im Konfigurationsassistenten für das Sicherheitssystem auf Seite 410 oder den Hauptdialog Sicherheitssystem auf Seite 430 wird der entsprechende Dialog "Servereinstellungen" angezeigt.

Konfigurieren der Servereinstellungen für Sicherheitsmodus: Verteilt – Client

Zum Konfigurieren der Servereinstellungen: Dialog Servereinstellungen für Sicherheitsmodus:

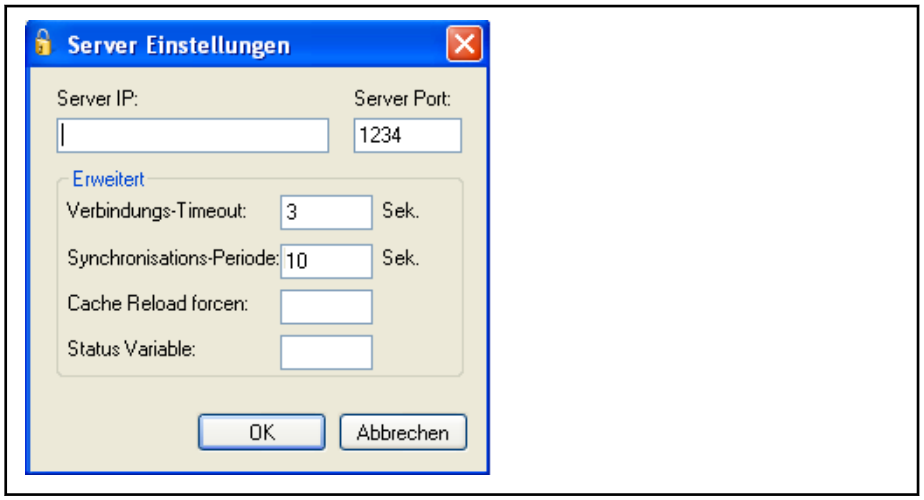


Abb.12-11: Verteilt – Client

1. Geben Sie in den Feldern **Server IP** und **Server Port** die IP-Adresse eines auf **Verteilt – Server** eingestellten Laufzeitprojekts ein.
2. Geben Sie im Feld **Zeitüberschreitung Verbindung** den Zeitraum (in Sekunden) ein, nach dessen Ablauf der Client einen erneuten Verbindungsversuch mit dem Server unternimmt (Ein üblicher Wert für die Zeitüberschreitung beträgt 3 Sekunden.)
3. Geben Sie im Feld **Synchronisationszyklus** die Frequenz (in Sekunden) ein, mit der der Client seine Konfiguration für das Sicherheitssystem mit der des Server synchronisiert. (Ein üblicher Wert für den Synchronisationszyklus beträgt 10 Sekunden.)
4. Geben Sie im Feld **Cache-Aktualisierung erzwingen** den Namen einer Anwendungsvariablen ein (Typ Integer oder Boolean). Wenn der Tag-Wert WAHR (nicht Null) ist und der Zeitstempel des angegebenen Servers älter als der des Clients ist, wird das lokale Sicherheitssystem mit veralteten Serverinformationen aktualisiert.
5. Geben Sie im Feld **Statusvariable** den Namen einer Anwendungsvariablen (Typ Integer) ein, die einen Wert zur Angabe des Status der Serververbindung erhält. Während das Projekt läuft, sind die folgenden Werte möglich: Weitere Informationen unter GetSecuritySystemStatus auf Seite 613.

Wert	Beschreibung
0	Kein Cache
1	Aktualisierter Cache
2	Veralteter lokaler Cache
3	Veralteter Server-Cache
4	Vom Server getrennt

6. Klicken Sie auf **OK**.

Projektsicherheit

Konfigurieren der Servereinstellungen für Sicherheitsmodus: Domäne (LDAP)

Zum Konfigurieren der Servereinstellungen: Dialog Servereinstellungen für Sicherheitsmodus:

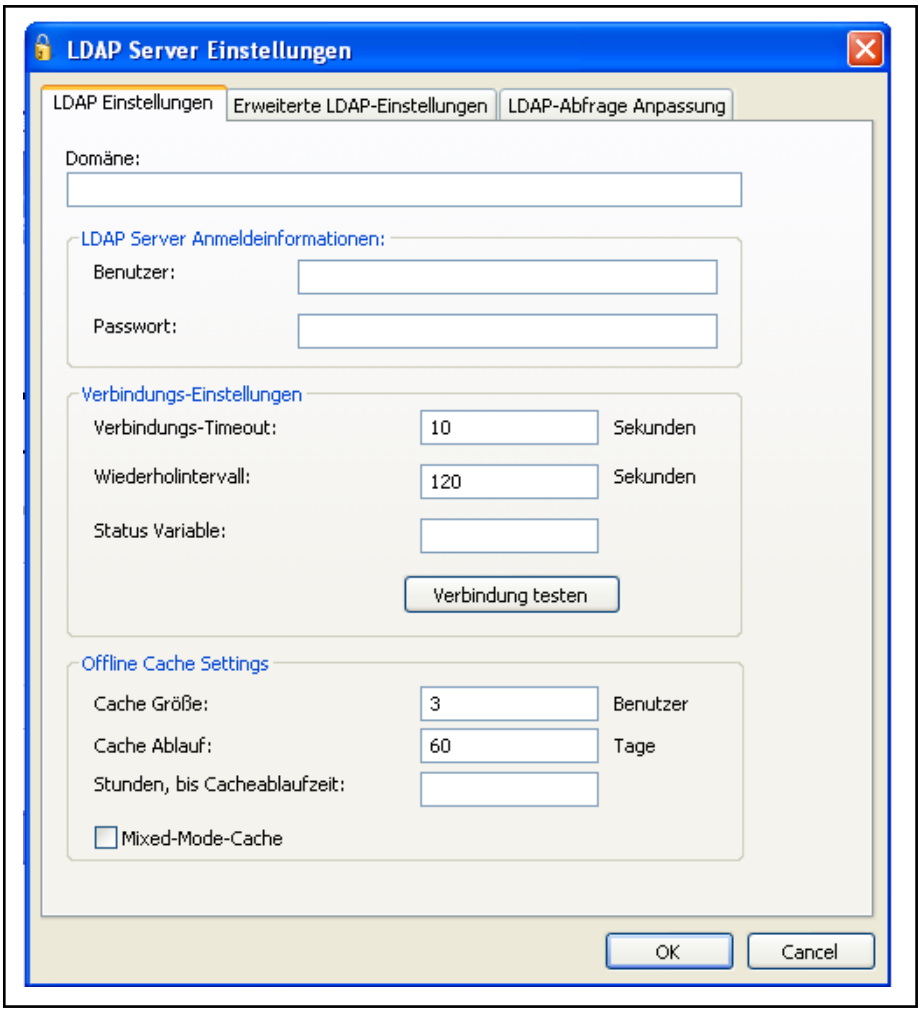


Abb. 12-12: Domäne (LDAP)

1. Geben Sie im Feld **Domäne** den Domänenname des LDAP-Servers ein.
2. Tragen Sie Ihre Zugangsdaten für den LDAP-Server in die Felder **Benutzername** und **Passwort** ein. Sie müssen über ausreichend Berechtigungen zum Abrufen von Gruppen- und Benutzerlisten verfügen. Bitte wenden Sie sich hierzu an Ihren LDAP-Administrator.
3. Geben Sie im Feld **Verbindungs-Timeout** den Zeitraum (in Sekunden) ein, nach dessen Ablauf der Client einen erneuten Verbindungsversuch mit dem LDAP-Server unternimmt (Ein üblicher Wert für die Zeitüberschreitung beträgt 4-5 Sekunden.)
4. Geben Sie im Feld **Wiederholungsintervall** die Frequenz (in Sekunden) ein, mit der der Client einen Verbindungsversuch zum LDAP-Server unternimmt, wenn die Verbindung nicht aufgebaut werden konnte.
5. Geben Sie im Feld **Statusvariable** den Namen einer Anwendungsvariablen (Typ Integer) ein, die einen Wert zur Angabe des Status der Serververbindung erhält. Während das Projekt läuft, sind die folgenden Werte möglich:

Wert	Beschreibung
0	Zeitüberschreitung Verbindung
1	Zeitüberschreitung Bindung
2	Zeitüberschreitung Abfrage
3	Getrennt
4	Verbunden
5	Abfrage ergab keine Benutzer oder Gruppen
6	Ungültige/r Benutzer oder Gruppe

Weitere Informationen unter GetSecuritySystemStatus auf Seite 613.

6. Klicken Sie auf **Verbindung testen**, um zu bestätigen, dass das Projekt sich mit der angegebenen Domäne verbinden kann. Wenn dies nicht möglich ist, überprüfen und korrigieren Sie Ihre Einstellungen.
7. Klicken Sie auf **OK**.

☞ Das Sicherheitssystem des Projekts hält die zuletzt verwendeten Benutzer im Cache vor, falls das Projekt seine Verbindungen mit dem LDAP-Server verlieren sollte. Diese Benutzer können sich weiterhin am Projekt anmelden. Die Größe des Cache kann durch Einstellen der **Cache-Größe** auf "0" auf unendlich eingestellt werden und durch Einstellung von **Ablauf Cache** auf "0" wird der Cache niemals ungültig.

Zum Überwachen des Cache während der Projektlaufzeit geben Sie den Namen einer Anwendungsvariablen im Feld **Cache-Gültigkeit** ein. Dieser Tag erhält einen Wert, der der Anzahl der Stunden bis zum Ablauf des Cache entspricht.

Wenn es Benutzern möglich sein soll, sich entweder mit lokalen oder mit server-definierten Konten anzumelden, wählen Sie **Gemischtmodus Cache**. Lokale Benutzer und Gruppen können nur erstellt werden, wenn der **Sicherheitsmodus** auf **Nur lokal** eingestellt ist. Nach ihrer Erstellung können sie im Cache vorgehalten werden, wenn der **Sicherheitsmodus** auf **Domäne (LDAP)** eingestellt ist.

Auf der Registerkarte "Erweiterte LDAP-Einstellungen" des Dialogs stehen zusätzliche Einstellungen zur Verfügung, die nur von erfahrenen LDAP-Administratoren konfiguriert werden sollten:

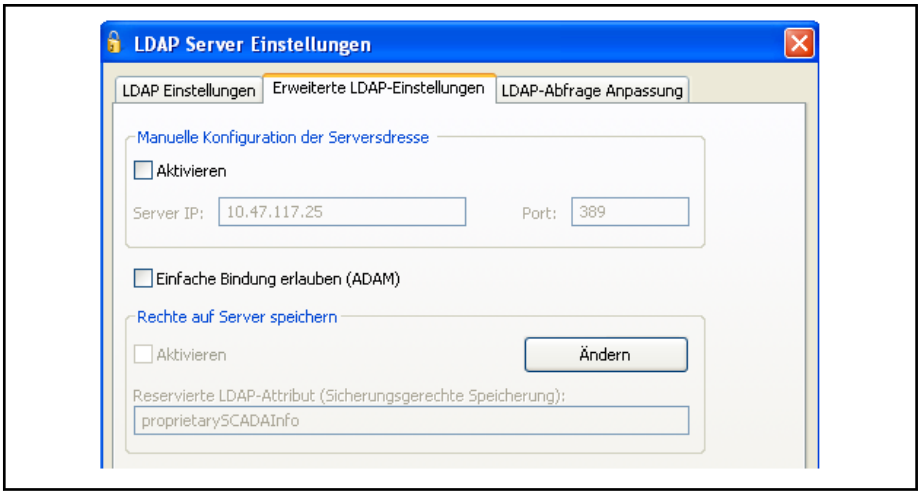


Abb. 12-13: 'Erweiterte LDAP-Einstellungen

Serveradresse manuelle Konfiguration

Wenn der LDAP-Server aus bestimmten Gründen nicht über seinen Domänen-Namen erreicht werden kann, kann die IP-Adresse des Servers manuell konfiguriert werden: Wählen Sie **Aktivieren** und geben Sie dann die IP-Adresse und Port-Nummer des LDAP-Servers ein.

☞ Der Standard-Port für LDAP ist 389. Überprüfen Sie aber die Port-Nummer mit dem Server-Administrator.

Einfache Bindung erlauben (ADAM)

LDAP erfordert normalerweise eine sichere Bindung für die Authentifizierung, allerdings bietet Active Directory Application Mode (ADAM) im Windows Server 2003 keine vollständige Unterstützung für sichere Bindung. Wählen Sie diese Option, um einfache Bindung mit einem ADAM-Server zu ermöglichen.

☞ Einfache Bindung bedeutet, dass Benutzerdaten im Klartext gesendet werden, die Verbindung sollte also durch andere Mittel wie zum Beispiel VPN, TLS/SSL oder Proxies gesichert werden.

Berechtigungen auf Server speichern

Standardmäßig werden die Sicherheitsberechtigungen von "WinStudio" innerhalb Ihres Projekts gespeichert. Die Berechtigungen können aber zurück zum LDAP-Server gespeichert werden, um sie anderen Projekten, die den gleichen LDAP-Server verwenden, zur Verfügung zu stellen oder für einfache Redundanz.

Damit diese Option funktioniert, muss zunächst das LDAP-Schema des Servers um zusätzliche Informationen über das Sicherheitssystem des Projekts erweitert werden. Siehe Erweitern des LDAP-Schemas zum Speichern von Sicherheitsberechtigungen auf Seite 417.

Wenn dies erledigt ist, klicken Sie auf **Ändern**, um Ihre LDAP-Serverzugangsdaten einzugeben, und wählen dann **Aktivieren**.

Auf der Registerkarte "Anpassung LDAP-Abfrage" des Dialogs kann das Format der LDAP-Serverabfragen weiter angepasst werden:

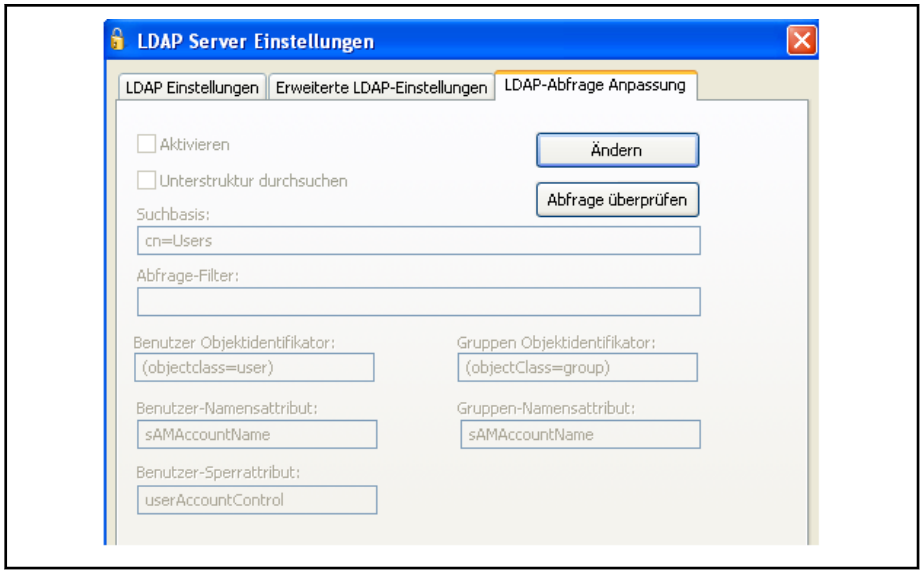


Abb. 12-14: Anpassung LDAP-Abfrage

Standardmäßig stellt der LDAP-Server eine Liste "aller" registrierten Benutzer und Gruppen zur Verfügung. In einer großen oder komplexen Netzwerkumgebung kann dies zu einer langen Liste führen, die für die Konfiguration des Sicherheitssystems Ihres Projekts sehr unhandlich wäre. Zum Einschränken der Liste der Benutzer und Gruppen kann die LDAP-Abfrage so angepasst werden, dass alle Benutzer bzw. Gruppen eliminiert werden, die niemals auf Ihr Projekt Zugriff haben sollen: Klicken Sie auf **Ändern** und geben Sie Ihre LDAP-Serverzugangsdaten ein, wählen Sie **Aktivieren** und konfigurieren Sie dann die Einstellungen **Suchbasis** und **Abfragefilter**. Die korrekte Syntax finden Sie in Ihrer LDAP-Serverdokumentation.

Einige nicht standardisierte LDAP-Implementierungen – z.B. Linux-basierte LDAP-Server und Active Directory Application Mode (ADAM) im Windows Server 2003 – verwenden außerdem andere Bezeichner und Attribute für Dateneinheiten. Diese können in diesem Dialog angepasst werden, aber auch dies sollte von einem erfahrenen LDAP-Administrator durchgeführt werden.

Beispiel für alternative Attribute in ADAM

LDAP-Server	Attribut Benutzername	Attribut Gruppenname	Attribut Benutzersperre
Active Directory	sAMAccountName	sAMAccountName	userAccountControl
Active Directory Application Mode (ADAM)	Name	Name	userAccountControl

12.5.2 Erweitern des LDAP-Schemas zum Speichern von Sicherheitsberechtigungen

Um Projekt-Sicherheitsberechtigungen von "WinStudio" zurück zu einem Domänen(LDAP)-Server zu speichern, muss das LDAP-Schema des Servers für die zusätzlichen Informationen erweitert werden.

Der Server muss bereits konfiguriert sein und in Ihrem Netzwerk laufen und Sie müssen über ausreichend Berechtigungen zum Ändern der Serverkonfiguration verfügen.

In dieser Prozedur werden neue Attribute mit der Bezeichnung "proprietarySCADAInfo" zur Aufnahme der Projekt-Sicherheitsberechtigungen von "WinStudio" erstellt. Anschließend werden die Attribute den Klassen "Person" und

"Gruppe" in der Serverkonfiguration hinzugefügt. Dieser Klassen entsprechen den Benutzern und Gruppen im Sicherheitssystem des Projekts.

Bitte beachten Sie, dass diese Vorgehensweise nur zeigt, wie das Schema im Microsoft Active Directory unter Windows Server 2003 erweitert wird. Für andere LDAP-Server und Betriebssysteme gilt eine andere Vorgehensweise, aber die grundlegenden Schritte sollten prinzipiell gleich sein. Bitte ziehen Sie Ihre LDAP-Serverdokumentation zu Rate.

HINWEIS	**Das Erweitern des LDAP-Schemas eines Servers kann nicht rückgängig gemacht werden.**

1. Registrierung der Schemaverwaltungs-DLL.
   1.1 Klicken Sie auf **Start** ▸ **Alle Programme** ▸ **Zubehör** ▸ **Eingabeaufforderung**.

   Ein Eingabeaufforderungsfenster wird angezeigt.

   1.2 Geben Sie in der Eingabeaufforderung **cd %SystemRoot%\System32** ein und drücken Sie die Eingabetaste.

   Das Arbeitsverzeichnis wird gewechselt.

   1.3 Geben Sie **regsvr32 schmmgmt.dll** ein und drücken Sie die Eingabetaste.

   Wenn die DLL erfolgreich registriert wurde, wird eine Bestätigungsmeldung angezeigt.

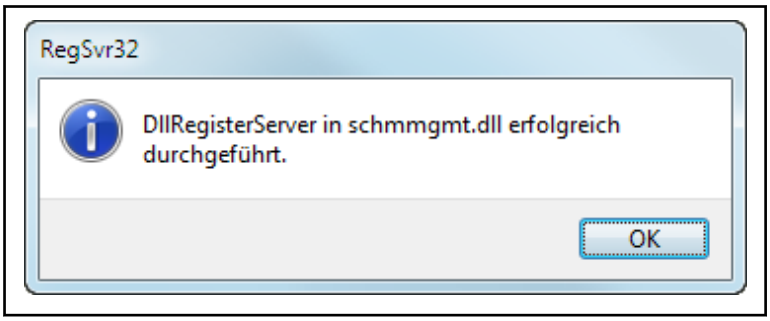


Abb.12-15: Bestätigungsmeldung

   1.4 Schließen Sie die Meldung durch Klicken auf OK.

   1.5 Schließen Sie das Eingabeaufforderungsfenster.

2. Hinzufügen des Active Directory Schema Snap-In zum Konsolenstamm.
   2.1 Klicken Sie auf **Start** ▸ **Alle Programme** ▸ **Zubehör** ▸ **Ausführen**. Der Dialog Ausführen wird angezeigt.

   2.2 Geben Sie im Feld **Öffnen** "mmc" ein und klicken Sie dann auf **OK**.

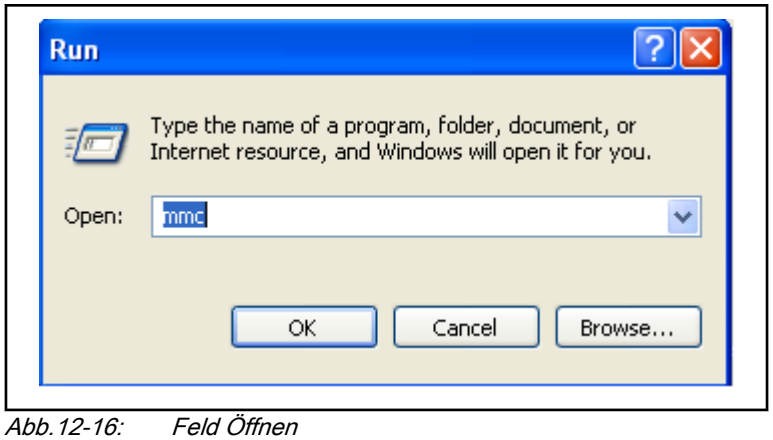


Abb.12-16: Feld Öffnen

(Wenn die Benutzerkontensteuerung (UAC) aktiviert ist, werden Sie um Bestätigung gebeten, dass Microsoft Management Console Änderungen vornehmen darf. Klicken Sie auf Ja.) Ein Konsolenfenster wird angezeigt.

2.3 Im Konsolenfenster klicken Sie auf **Datei ▸ Snap-In hinzufügen/entfernen**.

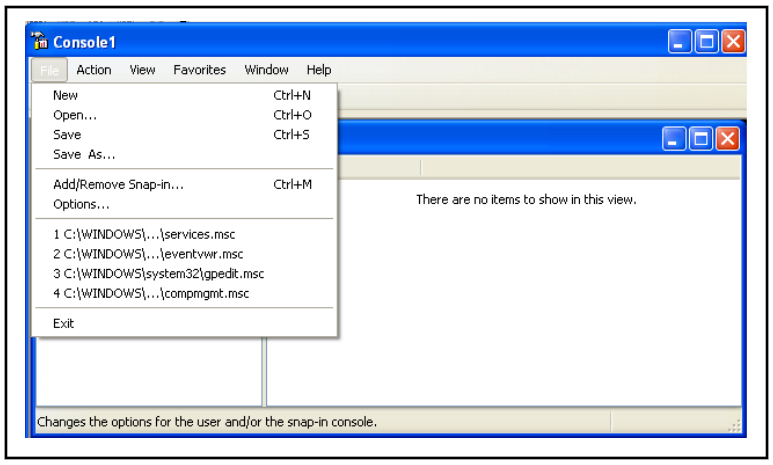


Abb.12-17: Menü: Snap-In hinzufügen/entfernen

Der Dialog Snap-In hinzufügen/entfernen wird angezeigt.

2.4 Wählen Sie aus der Liste **Snap-Ins** den Eintrag **Konsolenstamm** und klicken Sie auf **Hinzufügen**. Der Dialog "Eigenständiges Snap-In hinzufügen" wird angezeigt.

2.5 Wählen Sie aus der Liste der verfügbaren Snap-Ins den Eintrag **Active Directory Schema** und klicken Sie dann auf **Hinzufügen**.

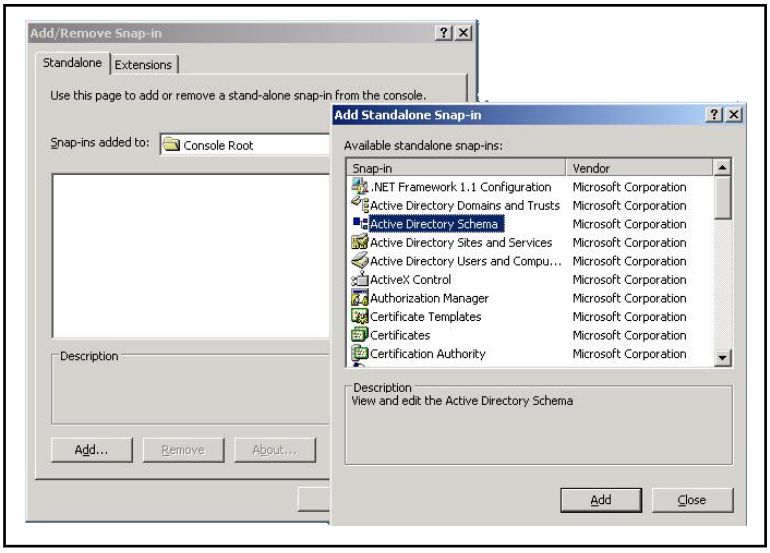


Abb.12-18: Dialog Eigenständiges Snap-In hinzufügen

Das Snap-In wird zum Konsolenstamm hinzugefügt.

2.6 Schließen Sie den Dialog Snap-In hinzufügen/entfernen durch Klicken auf **OK**.

3. Erstellen Sie das proprietäre SCADAInfo-Attribut im Active Directory Schema Snap-In. Das proprietäre SCADAInfo-Attribut wird zur Liste hinzugefügt.

3.1 Erweitern Sie in der Baumansicht des **Konsolenstamms** das **Active Directory Schema**.

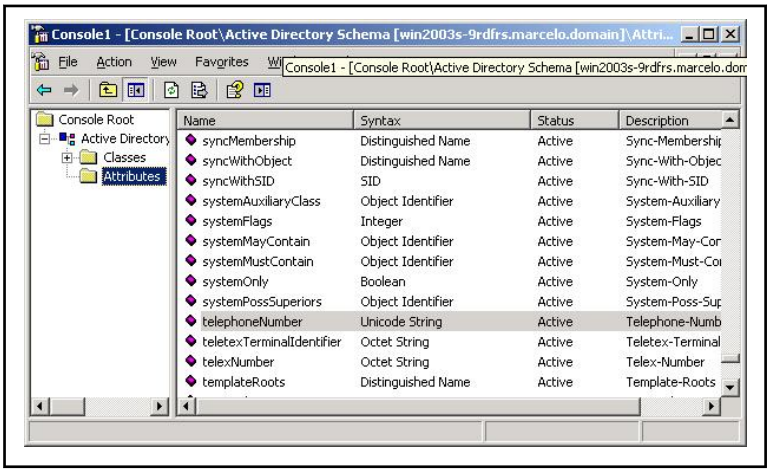


Abb.12-19: Active Directory Schema

3.2 Führen Sie einen Rechtsklick auf Active Directory **Schema** ▸ **Attribute** aus klicken Sie dann im Kontextmenü auf **Attribut erstellen**. Es wird eine Meldung angezeigt, dass Ihr Schema dauerhaft verändert wird.

3.3 Klicken Sie auf Weiter. Ein Dialog Neues Attribut erstellen wird angezeigt.

3.4 Füllen Sie die Felder im Dialog wie folgt aus.

- **Vollständiger Name**: `proprietarySCADAInfo`
- **LDAP Anzeigenname**: `proprietarySCADAInfo`
- **Unique X500 Object ID**: `0.7.7777.77777777.777.7.7`

Es sollte eine eindeutige Objekt-ID verwendet werden.

Beschreibung: `proprietarySCADAInfo`

Syntax: `Octet String`

Minimum: `0`

Maximum: `10240`

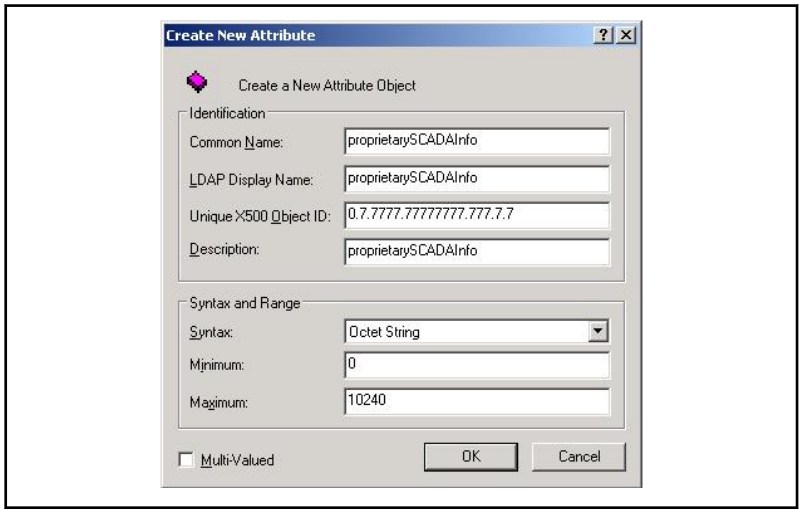


Abb. 12-20: Dialog: Neues Attribut erstellen

3.5 Schließen Sie den Dialog durch Klicken auf **OK**.

Ein Attribut proprietarySCADAInfo wird zur Liste hinzugefügt.

4. Fügen Sie das Attribut proprietarySCADAInfo zu den Klassen **Person** und **Gruppe** hinzu.

4.1 Wählen Sie in der Baumansicht des **Konsolenstamms** den Eintrag Active Directory **Schema** ▸ **Klassen**.

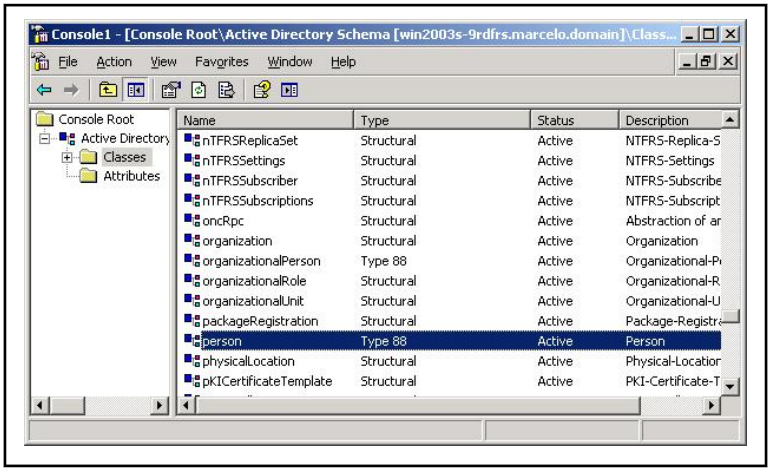


Abb. 12-21: Dialog Active Directory Schema > Klassen

4.2 Führen Sie in der Liste der Klassen einen Rechtsklick auf **Person** aus und klicken Sie dann im Kontextmenü auf **Eigenschaften**.

Der Dialog Eigenschaften wird angezeigt.

4.3 Klicken Sie im Dialog auf die Registerkarte **Attribute**.

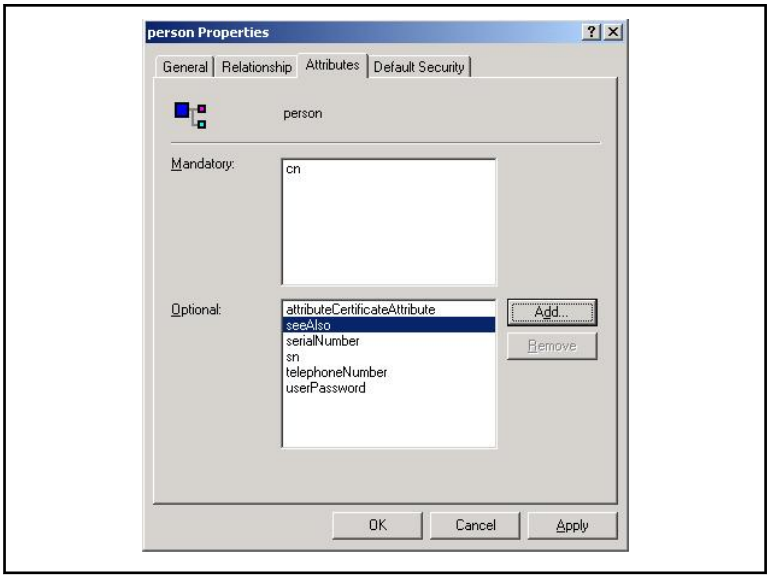


Abb.12-22: Dialog Personeneigenschaften, Attribute

4.4 Klicken Sie auf **Hinzufügen**. Der Dialog Schemaobjekt auswählen wird angezeigt.

4.5 Wählen Sie aus der Liste der Schemaobjekte **proprietarySCADAInfo** und klicken Sie auf **OK**.

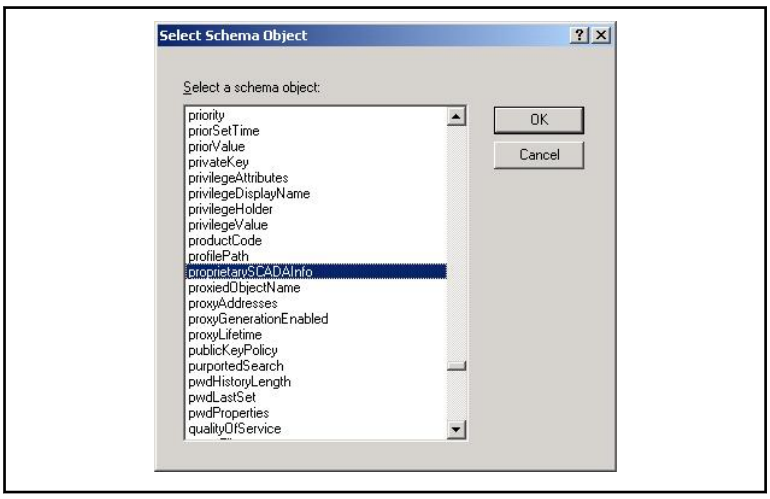


Abb.12-23: Dialog Schemaobjekt auswählen

Das Attribut wird den Klasseneigenschaften hinzugefügt.

4.6 Schließen Sie den Dialog Eigenschaften durch Klicken auf **OK**.

4.7 Wiederholen Sie die Schritte b bis f für die Klasse **Gruppe**.

5. Führen Sie in der Baumansicht des **Konsolenstamms** einen Rechtsklick auf **Active Directory Schema** aus und klicken Sie dann im Kontextmenü auf **Aktualisieren**.
6. Klicken Sie auf **Datei** ▸ **Schließen**, um das Konsolenfenster zu schließen.
7. Führen Sie einen Neustart des Servers durch.

12.6 Erstellen und Konfigurieren von Gruppen

Der Dialog "Gruppenkonto" dient dem Erstellen und Löschen von Benutzergruppen sowie dem Konfigurieren der Zugangsberechtigungen für eine ausgewählte Gruppe.

Zugriff auf den Dialog

Unter der Voraussetzung, dass das Sicherheitssystem des Projekts bereits aktiviert ist (d.h. der Konfigurationsassistent für das Sicherheitssystem wurde mindestens einmal verwendet), können Sie folgendermaßen auf diesen Dialog zugreifen:

- Öffnen Sie den Dialog Sicherheitssystem auf Seite 430 und klicken Sie auf **Gruppen** oder
- wählen Sie im Kontextmenü von WinStudio den Eintrag **Werkzeuge ▸ Gruppe einfügen** und klicken Sie auf **Gruppen**

.

Der Dialog im Detail

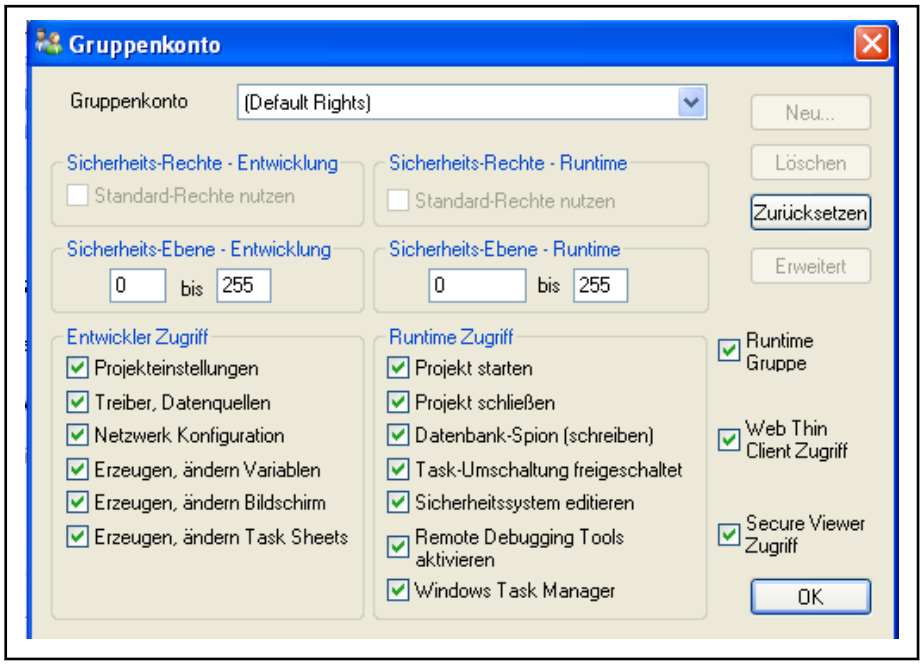


Abb. 12-24: Dialog Gruppenkonto

Bereich / Element	Beschreibung
Gruppenkonto	Die Benutzergruppe, die aktuell konfiguriert wird. **Hinweis:** Es gibt zwei Standardgruppen für alle Projekte: **Guest** und **(Standardberechtigungen)**. **Hinweis:** Wenn der Sicherheitsmodus auf **Domäne (LDAP)** eingestellt ist, ist zu beachten, dass die im Microsoft Active Directory eingebauten Gruppen in dieser Gruppenliste nicht erscheinen und dem Projekt nicht hinzugefügt werden können.
Neu	Erstellt eine neue Gruppe. Geben Sie im Dialog "Neues Gruppenkonto" den Namen der neuen Gruppe ein und klicken Sie dann auf **OK**.
Löschen	Löscht die aktuell ausgewählte Gruppe.

Projektsicherheit

Bereich / Element		Beschreibung
Zurücksetzen		Setzt die Berechtigungen der aktuell ausgewählten Gruppe auf die Einstellungen der Gruppe **(Standardberechtigungen)** zurück. Hierdurch wird die Gruppe nicht auf den Standard gesperrt, es können weitere Änderungen vorgenommen werden. Zum Sperren der Gruppe siehe **Standardberechtigungen verwenden** weiter unten.
Weitere Einstellungen		Öffnet den Dialog "Weitere Einstellungen Gruppenkonto" (siehe unten).
Sicherheitsberechtigungen – Entwicklung	**Standardberechtigungen verwenden**	Sperrt die Berechtigungen der aktuell ausgewählten Gruppe auf die für die Gruppe **(Standardberechtigungen)** konfigurierten Einstellungen. Werden Änderungen an der Gruppe **(Standardberechtigungen)** vorgenommen, werden diese auch auf diese Gruppe angewendet.
Sicherheitsebene – Entwicklung	**Bereich**	Der Bereich der Zugriffsstufen auf Seite 408, auf den diese Gruppe in der Entwicklungsanwendung Zugriff hat.
Entwicklungsberechtigungen	**Projekteinstellungen**	Mitglieder der Gruppe können die Projekteinstellungen auf Seite 67 ändern.
	Treiber, Datenquellen	Mitglieder der Gruppe können Gerätetreiber und externe Datenquellen auf Seite 379 erstellen und ändern.
	Netzwerkkonfiguration	Mitglieder der Gruppe können TCP/IP-Client-Arbeitsblätter auf Seite 401 erstellen und ändern.
	Tags erstellen, ändern	Mitglieder der Gruppe können Projekt-Tags auf Seite 116 erstellen und ändern.
	Bilder erstellen, ändern	Mitglieder der Gruppe können Projektbilder auf Seite 169 erstellen und ändern.
	Task-Arbeitsblätter erstellen, ändern	Mitglieder der Gruppe können Task-Arbeitsblätter auf Seite 357 erstellen und ändern.
Sicherheitsberechtigungen – Laufzeit	**Standardberechtigungen verwenden**	Sperrt die Laufzeitberechtigungen der aktuell ausgewählten Gruppe auf die für die Gruppe **(Standardberechtigungen)** konfigurierten Einstellungen. Werden Änderungen an der Gruppe **(Standardberechtigungen)** vorgenommen, werden diese auch auf diese Gruppe angewendet.
Sicherheitsstufe – Laufzeit	**Bereich**	Der Bereich der Zugriffsstufen auf Seite 408, auf die diese Gruppe im Laufzeitprojekt Zugriff hat.

Bereich / Element		Beschreibung
Laufzeitberechtigungen	Projekt ausführen	Mitglieder der Gruppe können das Projekt ausführen.
	Projekt schließen	Mitglieder der Gruppe können das Projekt anhalten.
	Datenbankspion (schreiben)	Mitglieder der Gruppe können mithilfe des Fensters Datenbankspion auf Seite 93 Werte in die Projektdatenbank schreiben. **Hinweis:** Dies gilt nur für lokal ausgeführte Projekte. Für Remote-Projekte siehe **Werkzeuge zur Remote-Fehlerbeseitigung freigeben**.
	Task-Umschaltung freigegeben	Mitglieder der Gruppe können vom Laufzeitprojekt zu einem anderen Windows-Task wechseln. (Diese Option gilt nicht für auf Windows Embedded Zielsystemen ausgeführte Projekte; der Benutzer kann immer aus dem Laufzeitprojekt heraus wechseln.)
	Sicherheitssystem bearbeiten	Mitglieder der Gruppe können während der Laufzeit Änderungen am Sicherheitssystem des Projekts vornehmen. **Hinweis:** Diese Option darf für Ihre eigene Gruppe nicht deaktiviert werden, andernfalls sind Sie gegebenenfalls nicht in der Lage, Ihrer eigenen Änderungen rückgängig zu machen.
	Werkzeuge zur Remote-Fehlerbeseitigung freigeben	Mitglieder der Gruppe können den Remote-Datenbank Spion auf Seite 447 und Remote LogWin auf Seite 448 verwenden.
	Windows Task-Manager	Mitglieder der Gruppe können den Windows Task-Manager öffnen. (Diese Option gilt nicht für auf Windows Embedded Zielsystemen ausgeführte Projekte; der Benutzer kann immer den Windows Task-Manager öffnen.) **Hinweis:** Durch Deaktivieren dieser Option wird der Task-Manager während der Laufzeit deaktiviert, wofür Administratorrechte erforderlich sind. Dafür muss das Projekt mit höheren Berechtigungen ausgeführt werden.
Laufzeitgruppe		Ein durch Aufrufen der Funktion CreateUser auf Seite 611 während der Laufzeit erstellter Benutzer kann dieser Gruppe zugewiesen werden.
Web-Thin-Client-Zugriff		Mitglieder der Gruppe können mithilfe eines Web Thin Client auf das Laufzeitprojekt zugreifen.
Secure Viewer-Zugriff		Mitglieder der Gruppe können mithilfe eines Secure Viewer auf das Laufzeitprojekt zugreifen.

Projektsicherheit

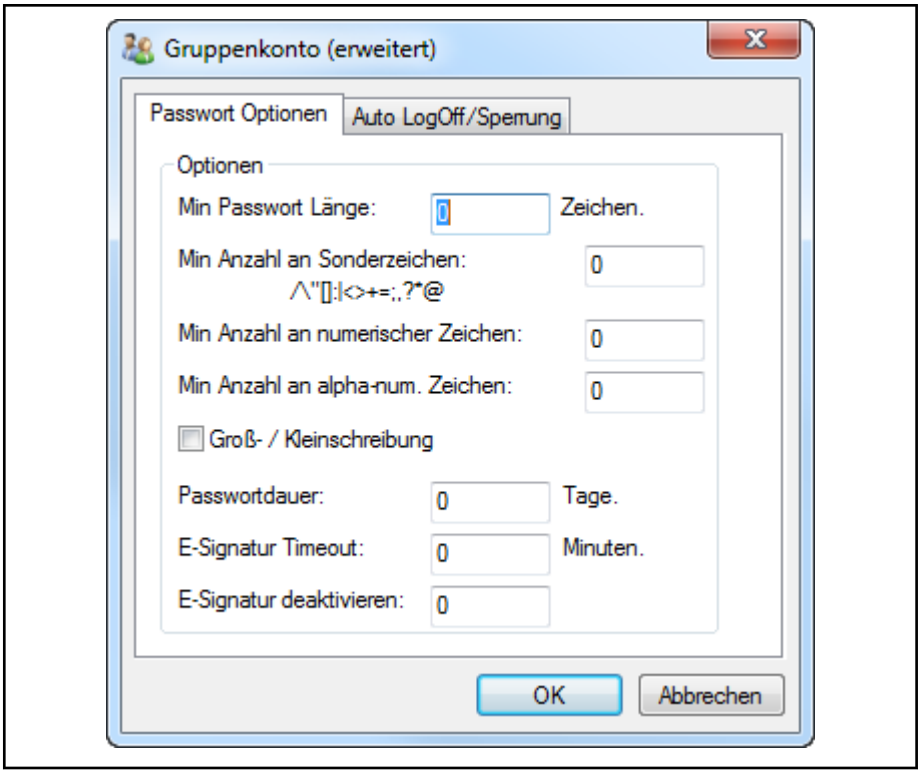


Abb.12-25: Dialog Weitere Einstellungen – Passwortoptionen

Bereich / Element	Beschreibung
Min. Passwortlänge **Min. Anzahl Sonderzeichen** **Min. Anzahl Ziffern** **Min. Anzahl Buchstaben**	Um Benutzerpasswörter komplexer und dadurch sicherer zu machen, kann eingestellt werden, dass diese eine bestimmte Anzahl von Buchstaben (A-Z, a-z), Ziffern (0-9) und Sonderzeichen (Satzzeichen) enthalten müssen. Wenn der Benutzer zum Ändern seines Passworts aufgefordert wird – zum Beispiel wenn das alte Passwort abgelaufen ist (siehe **Passwortalterung** unten) – wird das neue Passwort nur akzeptiert, wenn es diesen Bedingungen entspricht.
Groß-/Kleinschreibung	Wenn diese Option ausgewählt wird, muss bei Passwörtern die Groß-/Kleinschreibung beachtet werden – d.h. mit Groß- und Kleinbuchstaben erstellte Passwörter "müssen" vom Benutzer in genau dieser Form eingegeben werden. **Hinweis:** Bei mit "Rexroth WinStudio v6.5" bis "v7.2+SP1" erstellten Projekten gilt ist alle Passwörter Groß-/Kleinschreibung erforderlich.
Passwortdauer	Gültigkeit (in Tagen) des Passworts für alle Benutzer in dieser Gruppe. Nach Ablauf der angegebenen Anzahl von Tagen zwingt das Projekt den Benutzer zur Änderung seines Passworts: Wenn der Benutzer versucht, sich anzumelden, wird automatisch der Dialog "Passwort ändern" angezeigt und der Benutzer kann die Anmeldung erst nach Erstellen eines neuen Passworts abschließen. Standardmäßig muss der Benutzer ein neues Passwort wählen, das sich vom alten Passwort unterscheidet. Um dies so zu ändern, dass der Benutzer das gleiche Passwort wiederverwenden kann, wird die Projektdatei (**projekt_name.app**) mit der folgenden Einstellung manuell geändert: **[Sicherheit]** **ChangePasswordMode=1** Wenn **Passwortalterung** auf "0" eingestellt ist, laufen Passwörter niemals ab.

Bereich / Element	Beschreibung
E-Signatur Timeout	Zeitüberschreitungsdauer (in Minuten) der Eingabeaufforderung für die E-Signatur für alle Benutzer in dieser Gruppe. Der Benutzer muss Benutzername und Passwort vor Ablauf der festgelegten Zeitüberschreitungsdauer eingeben, um Projektfunktionen verwenden zu können, die eine E-Signatur erfordern.
E-Signatur deaktivieren	Wenn der Wert in diesem Feld WAHR (nicht Null) ist, können Benutzer in dieser Gruppe keine Projektfunktionen verwenden, die eine E-Signatur erfordern. In diesem Feld kann ein Projekt-Tag konfiguriert werden, so dass die E-Signatur während der Laufzeit dynamisch aktiviert/deaktiviert wird.

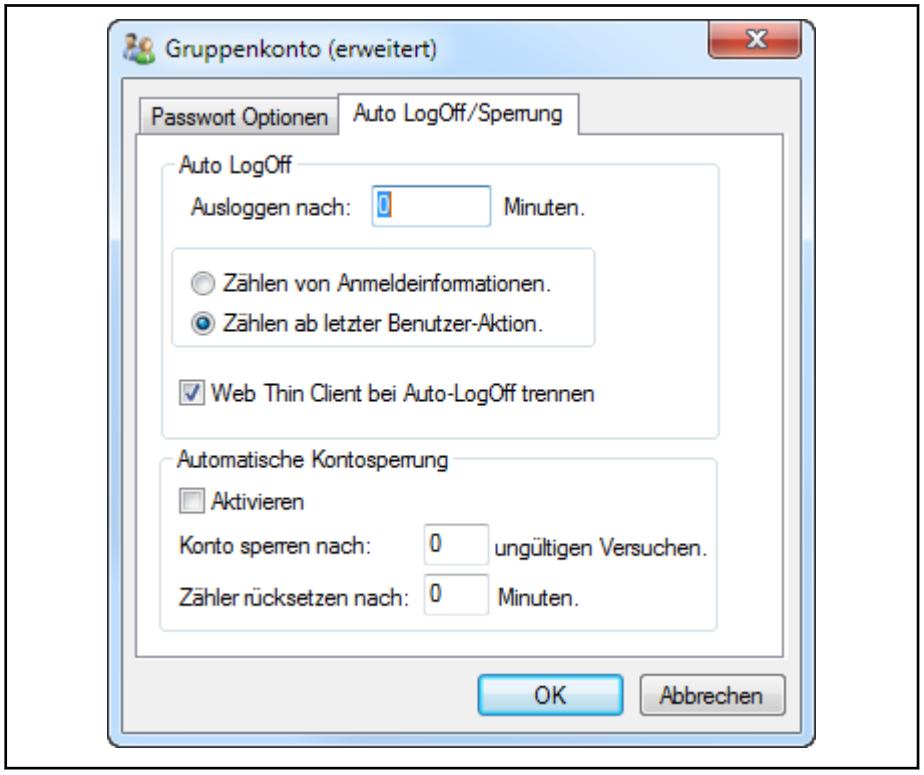


Abb. 12-26: Dialog Weitere Einstellungen – Auto abmelden/sperren

Bereich / Element		Beschreibung
Automatische Abmeldung	**Abmelden nach**	Anzahl der Minuten, nach der der aktuelle Benutzer automatisch abgemeldet werden muss. Wenn dieses Feld leer bleibt (oder den Wert 0 enthält), wird der aktuelle Benutzer niemals automatisch abgemeldet.
	Zählung ab Anmeldung	Wenn diese Option ausgewählt ist, wird der aktuelle Benutzer nach Ablauf der im Feld **Abmelden nach** konfigurierten Zeitdauer ab Anmeldung des aktuellen Benutzers am System automatisch abgemeldet.
	Zählung ab letzter Benutzeraktion	Wenn diese Option ausgewählt ist, wird der aktuelle Benutzer nach Ablauf der im Feld **Abmelden nach** konfigurierten Zeitdauer ab der letzten vom aktuellen Benutzer durchgeführten Aktion (Maus-, Touchscreen- oder Tastaturaktion) automatisch abgemeldet.
	Web Thin Client bei Autom. Abmeldung trennen	Wenn der Benutzer über einen Web Thin Client angemeldet ist, wird bei Abmeldung des Benutzers der Client automatisch vom Datenserver getrennt.

Projektsicherheit

Bereich / Element		Beschreibung
Automatische Kontosperrung	**Aktivieren**	Freigabe der nachfolgend beschriebenen automatischen Sperrfunktionen.
	Konto sperren nach	Maximale Anzahl von Anmeldeversuchen für einen Benutzer. Wenn der Benutzer innerhalb des im Feld **Zähler zurücksetzen nach** festgelegten Zeitraums die festgelegte Höchstzahl an Versuchen überschreitet (ein ungültiges Passwort eingibt), wird der Benutzer automatisch vom Projekt gesperrt.
	Zähler zurücksetzen nach	Definiert, wie lange (in Minuten) das Projekt nach einem ungültigen Anmeldeversuch wartet, bis der Zähler für Anmeldeversuche zurückgesetzt wird.

☞ Gehört ein Benutzer zu mehr als einer Gruppe (siehe Erstellen und Konfigurieren von Benutzern auf Seite 428), können die Gruppeneinstellungen miteinander kollidieren. Die Korrektur der Einstellungen hängt von der Art der Einstellungen ab:

- Die Einstellungen im oben beschriebenen Dialog "Gruppenkonto" sind "tolerant", d.h. für den Benutzer gilt die toleranteste Einstellung aus allen Benutzergruppen. Wenn zum Beispiel eine der Gruppen zum Erstellen und Ändern von Tags berechtigt ist, dann kann der Benutzer Tags erstellen und ändern.
- Die Einstellungen im oben beschriebenen Dialog "Weitere Einstellungen Gruppenkonto" (bei Registerkarten) sind "restriktiv", d.h. für den Benutzer gilt die restriktivste Einstellung aus allen Benutzergruppen. Wenn zum Beispiel die Mindestpasswortlänge für eine Gruppe 8 Zeichen und für eine andere Gruppe 12 Zeichen beträgt, dann ist die Mindestpasswortlänge für den Benutzer 12. (Für Automatische Abmeldung gilt gesondert **Zählung ab Anmeldung** hat Vorrang vor **Zählung ab letzter Benutzeraktion**.)

12.7 Erstellen und Konfigurieren von Benutzern

Zum Erstellen und Verwalten von Konten für Projektbenutzer klicken Sie im Dialog "Sicherheitssystem" auf die Schaltfläche **Benutzer**.

Der Dialog "Benutzerkonto" wird angezeigt.

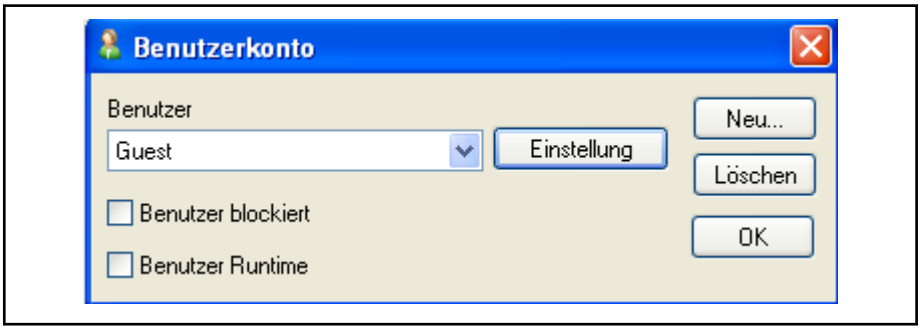


Abb.12-27: Dialog Benutzerkonto

Wenn sich nach Initialisierung des Projekts kein Benutzer anmeldet (oder wenn sich der aktuelle Benutzer abmeldet), meldet das Projekt automatisch den Standardbenutzer (**Guest**) an. Zusätzlich zum Standardbenutzer **Guest** gibt es eine Gruppe **Guest**, die über Standardberechtigungen zum Aktivieren aller Tasks verfügt. Wir empfehlen, die Berechtigungen der Gruppe **Guest** zu überprüfen und zu bearbeiten, um ein Mindestanzahl an Berechtigungen für den Startvorgang festzulegen.

Zum Erstellen eines neuen Benutzers klicken Sie auf **Neu** zum Öffnen des Dialogs "Neues Benutzerkonto".

Zum Löschen eines Benutzers klicken Sie auf die Combobox-Schaltfläche **Benutzer**, wählen den Benutzernamen aus der Liste und klicken dann auf **Löschen**.

Zu Konfigurieren eines Benutzers gehen Sie wie folgt vor:

1. Klicken Sie auf die Combobox-Schaltfläche **Benutzer** und wählen Sie einen Benutzer aus der Liste.
2. Bei Bedarf klicken Sie auf das Kontrollkästchen **Benutzer gesperrt**, um den ausgewählten Benutzer zu entsperren.

☞ Die Option **Laufzeitbenutzer** gibt an, dass ein Benutzer mithilfe der Funktion CreateUser auf Seite 611 während der Laufzeit erstellt wurde.

Sie kann für einen mit der hier beschriebenen Vorgehensweise erstellten Benutzer nicht verwendet werden.

---

3. Klicken Sie auf die Schaltfläche **Einstellungen**, um den Dialog "Benutzereinstellungen" zu öffnen:

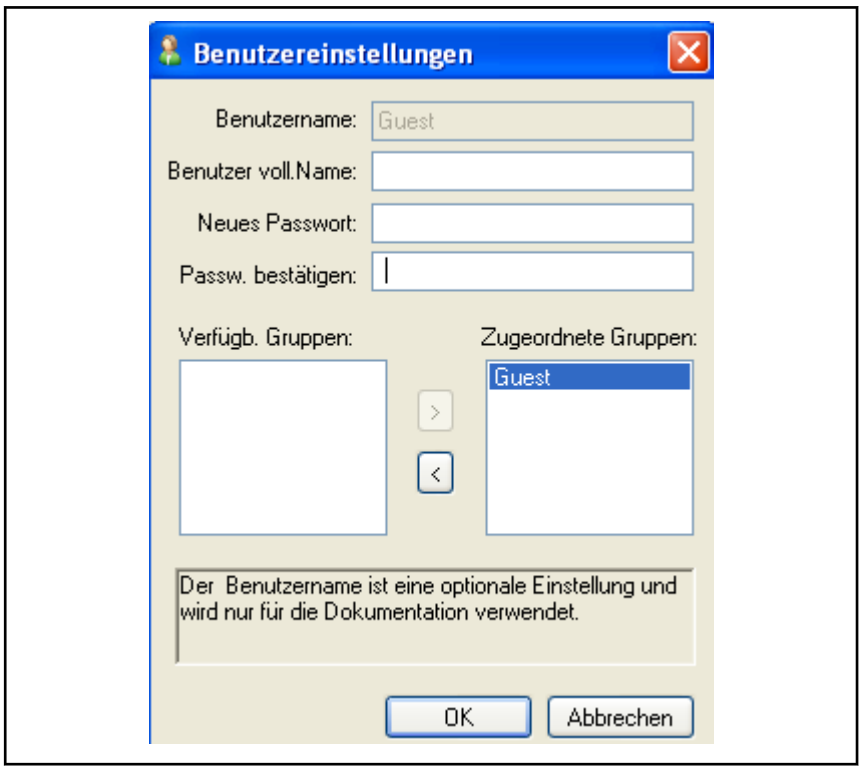


Abb. 12-28: Dialog Benutzereinstellungen

4. Die Parameter in diesem Dialog werden wie folgt verwendet:
   - Textfeld **Vollständiger Benutzername** ("optional"): Geben Sie den vollständigen Benutzername ein.
   - Textfeld **Neues Passwort**: Geben Sie das Benutzerpasswort ein.
   - Textfeld **Passwort bestätigen**: Geben Sie das Benutzerpasswort erneut ein.

5. Wählen Sie in der Liste **Verfügbare Gruppen** die Gruppe(n), zu der der/denen Benutzer zugewiesen werden soll und klicken Sie dann auf **>**, um diese Gruppe(n) in die Liste **Zugewiesene Gruppen** zu verschieben.
6. Nach Fertigstellung klicken Sie auf **OK**, um die Änderungen anzuwenden und den Dialog "Einstellungen" zu speichern.

12.8 Dialog Sicherheitssystem

Der Hauptdialog "Sicherheitssystem" dient der Verwaltung des Projekt-Sicherheitssystems nach der erstmaligen Konfiguration.

Zugriff auf den Dialog

Unter der Voraussetzung, dass das Sicherheitssystem des Projekts bereits aktiviert ist (d.h. der Konfigurationsassistent für das Sicherheitssystem wurde mindestens einmal verwendet), können Sie folgendermaßen auf diesem Dialog zugreifen:

Wählen Sie im Kontextmenü von WinStudio den Eintrag **Werkzeuge ▸ Sicherheitseinstellungen**.

Wenn eine diese Aktionen "vor" Aktivierung des Sicherheitssystems ausgefüllt wird, öffnet sich automatisch der Konfigurationsassistent Sicherheitssystem auf Seite 410.

Wenn das Sicherheitssystem bereits konfiguriert und ein Hauptpasswort gesetzt wurde, wird das Passwort abgefragt.

Der Dialog im Detail

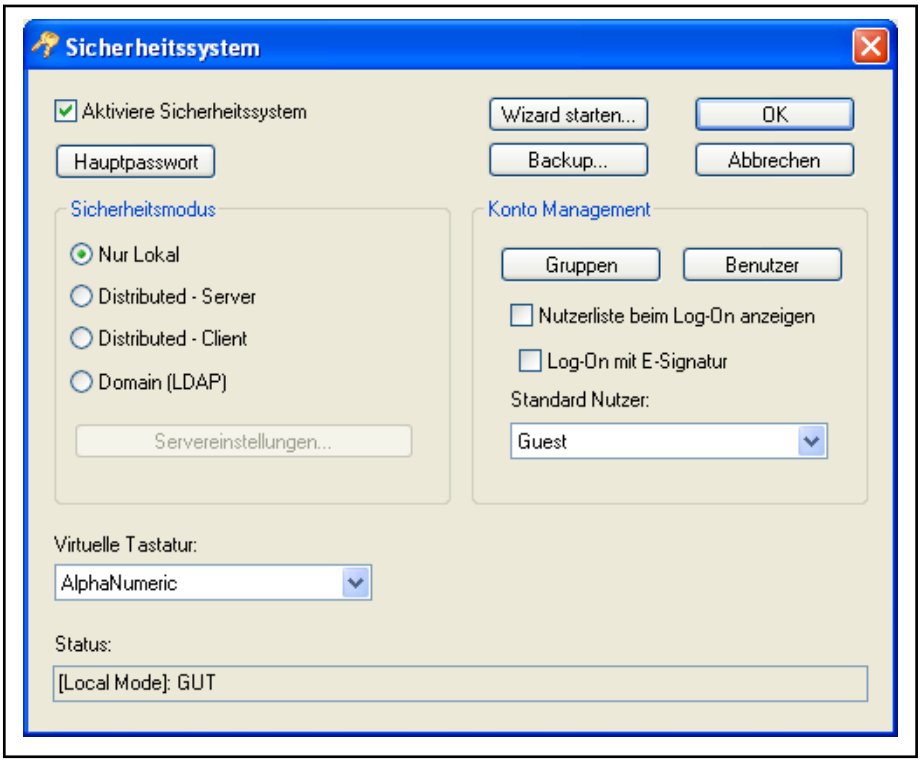


Abb. 12-29: Dialog Sicherheitssystem

Bereich / Element	Beschreibung
Sicherheitssystem aktivieren	Zeigt an, ob das Sicherheitssystem des Projekts derzeit aktiviert ist. Falls ja, gelten die festgelegten Zugriffsberechtigungen für Benutzer und Gruppen.
Hauptpasswort	Öffnet einen Dialog zur Festlegung eines Hauptverwaltungspasswort für das gesamte Projekt.

Bereich / Element		Beschreibung
Sicherheitsmodus	Modus	Der derzeitige Sicherheitsmodus auf Seite 407 des Projekts.
	Servereinstellungen	Öffnet den Dialog Servereinstellungen auf Seite 412 zur Konfiguration der Servereinstellungen für **Verteilt – Client** oder **Domäne (LDAP)**.
Wizard starten		Öffnet den Konfigurationsassistenten Sicherheitssystem auf Seite 410.
Backup		Öffnet den Dialog Import/Export auf Seite 431 zum Importieren oder Exportieren der Konfiguration des Sicherheitssystems.
Kontenverwaltung	**Gruppen**	Öffnet den Dialog Gruppenkonto auf Seite 423 zum Erstellen und Konfigurieren von Gruppen.
	Benutzer	Öffnet den Dialog Benutzerkonto auf Seite 428 zum Erstellen und Konfigurieren von Benutzern.
	Benutzerliste bei Anmeldung anzeigen	Zeigt eine Liste der verfügbaren Benutzer an (im Dialog Anmelden auf Seite 432), wenn ein Benutzer zum Anmelden aufgefordert wird. Der Benutzer kann aus dieser Liste wählen, anstatt seinen Benutzernamen einzugeben.
	Anmelden bei E-Signatur	Zwingt einen Benutzer zur Anmeldung mit dessen eigenem Benutzerkonto, wenn für ein Ereignis die E-Signatur angefordert wird. Wird dies nicht ausgewählt, bleibt das aktuelle Benutzerkonto angemeldet, ungeachtet dessen, wer die E-Signatur für das Ereignis bereitstellt.
	Standardbenutzer	Dieser Benutzer wird automatisch angemeldet, wenn kein anderer Benutzer angemeldet ist, z.B. wenn die Sitzung des vorherigen Benutzers abgelaufen ist oder dieser sich manuell abmeldet. **Hinweis:** Die Berechtigungen dieses Benutzers sollten stark eingeschränkt werden, um Ihr Projekt keinen übermäßigen Risiken auszusetzen.
Virtuelle Tastatur		Die Art der virtuellen Tastatur, die für die Anmeldeaufforderung für einen Benutzer am Client angezeigt wird.

12.9 Datensicherung der Sicherheitssystemkonfiguration

Eine Datensicherung der Sicherheitssystemkonfiguration Ihres Projekts kann durch Exportieren in eine Datei durchgeführt werden. Es kann auch eine Konfiguration aus einer Datei oder aus einem anderen Laufzeitprojekt importiert werden.

Exportieren der Konfiguration in eine Datei

Exportieren der Sicherheitssystemkonfiguration:

1. Klicken Sie im Hauptdialog Sicherheitssystem auf Seite 430 auf **Backup**. Der Dialog "Import/Export" wird angezeigt.
2. Klicken Sie auf **Exportieren in Datei**. Ein Standard-Dialog "Speichern unter" wird angezeigt.
3. Geben Sie einen Dateinamen und Pfad für die Datei an und klicken Sie dann auf **OK**.

Die exportierte Datei wird mit dem im Dialog "Sicherheitssystem" konfigurierten Hauptpasswort verschlüsselt.



Die Konfiguration kann auch während der Laufzeit durch Aufrufen der Funktion ExportSecuritySystem auf Seite 612 exportiert werden.

Importieren der Konfiguration aus einer Datei

Wenn der Sicherheitsmodus auf Seite 407 Ihres Projekts auf **Nur lokal** eingestellt ist, kann eine Konfiguration aus einer zuvor exportierten Datei importiert werden.

Importieren der Sicherheitssystemkonfiguration:

1. Klicken Sie im Hauptdialog Sicherheitssystem auf Seite 430 auf **Datensicherung**. Der Dialog "Import/Export" wird angezeigt.
2. Klicken Sie auf **Importieren aus Datei**: Ein Standard-Dialog "Öffnen" wird angezeigt.
3. Suchen Sie die zu importierende Konfigurationsdatei (***.dat**) und klicken Sie auf **OK**. Sie werden zur Eingabe des Hauptpassworts für die Konfiguration aufgefordert.
4. Geben Sie das Passwort ein und klicken Sie dann auf **OK**. Der Dialog "Importieren aus Datei" wird angezeigt.
5. Wählen Sie eine Importmethode:
   - **Nur Einstellungen ohne Konflikte importieren**: Die importierten Einstellungen mit den aktuellen Projekteinstellungen zusammenführen. Bei Konflikten werden die aktuellen Einstellungen beibehalten.
   - **Alle Einstellungen importieren und Konflikte ersetzen**: Die importierten Einstellungen mit den Projekteinstellungen zusammenführen. Bei Konflikten werden die importierten Einstellungen verwendet.
   - **Aktuelle Einstellungen ersetzen**: Die aktuellen Projekteinstellungen vollständig durch die aus der Datei importierten Einstellungen ersetzen.
6. Klicken Sie auf **OK**.



Die Konfiguration kann auch während der Laufzeit durch Aufrufen der Funktion ImportSecuritySystem auf Seite 617 importiert werden.

Importieren der Konfiguration aus einem anderen Projekt

Wenn der Sicherheitsmodus auf Seite 407 Ihres Projekts auf **Verteilt – Server** eingestellt ist, kann eine Konfiguration aus einem anderen Projekt importiert werden, wenn:

- der Sicherheitsmodus des anderen Projekts auf **Verteilt – Client** eingestellt ist und dessen Servereinstellungen zur Verwendung Ihres Projekts als Server konfiguriert sind; und
- das andere Projekt aktuell im gleichen Netzwerk läuft.

Importieren der Sicherheitssystemkonfiguration:

1. Klicken Sie im Hauptdialog Sicherheitssystem auf Seite 430 auf **Backup**. Der Dialog "Import/Export" wird angezeigt.
2. Klicken Sie auf **Importieren von Client-Station**: Der Dialog **Sicherheit von Client-Station importieren** wird angezeigt. Der Dialog zeigt eine Liste mit Laufzeitprojekten an, die Ihr Projekt als Sicherheitssystemserver verwenden. Jede Projekt-/Client-Liste enthält einen Zeitstempel, der den Zeitpunkt der letzten Zwischenspeicherung der Sicherheitssystemkonfiguration angibt.
3. Wählen Sie eine Client-Station aus und klicken Sie dann auf **Importieren von Client**. Sie werden zur Eingabe des Hauptpassworts für die Konfiguration aufgefordert.
4. Geben Sie das Passwort ein und klicken Sie dann auf **OK**.

12.10 An- / Abmelden

Wenn das Sicherheitssystem des Projekts aktiviert ist und die Berechtigungen des Standardbenutzers "Guest" eingeschränkt sind, müssen Sie sich an-

melden, um die Entwicklungsanwendung und/ oder das Laufzeitprojekt vollständig verwenden zu können.



Das Sicherheitssystem des Projekts muss vor Verwendung dieser Funktion aktiviert sein.

Zum Anmelden an der Entwicklungsanwendung wählen Sie im Kontextmenü von WinStudio den Eintrag **Werkzeuge ▸ Benutzer an-/abmelden**.

Um einen Benutzer zur Anmeldung am Laufzeitprojekt aufzufordern, gehen Sie wie folgt vor:

- Rufen Sie an einer Stelle, an der ein Ausdruck konfiguriert werden kann, die Funktion LogOn auf Seite 632 auf – z.B. durch Zeichnen eines Schaltflächenobjekts auf Seite 191 in einem Bild und Anwendung der Befehlsanimation auf Seite 243, so dass durch Drücken der Schaltfläche einer Anmeldeaufforderung angezeigt wird oder
- wählen Sie die Option **Anmelden bei E-Signatur** (im Hauptdialog Sicherheitssystem auf Seite 430), durch die der Benutzer bei jeder Aktion, die eine E-Signatur erfordert, zum Anmelden aufgefordert wird.

Der Dialog Anmelden wird entweder in der Entwicklungsanwendung oder im Laufzeitprojekt angezeigt:

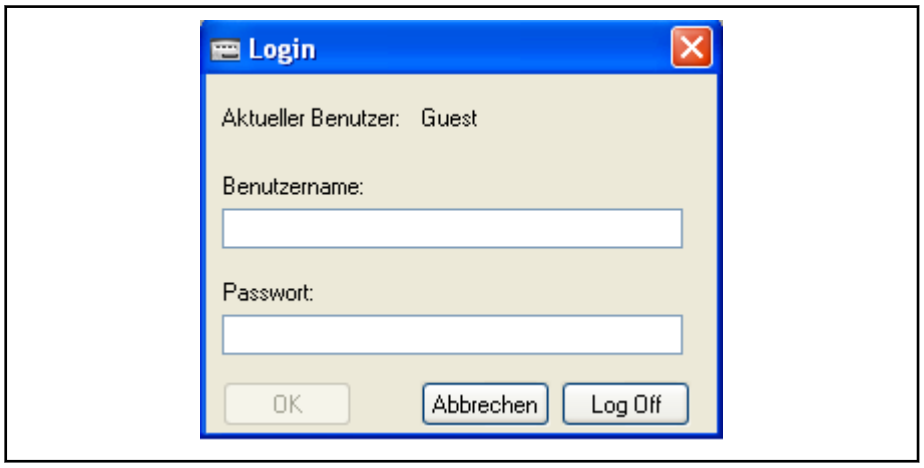


Abb. 12-30: Dialog Anmelden

Verwenden Sie diesen Dialog wie folgt:

- Um sich mit Ihrem eigenen Konto anzumelden, geben Sie Ihren Benutzernamen und Ihr Passwort in die entsprechenden Felder ein und klicken dann auf **OK**.
- Um sich als der Standardbenutzer „Guest“ anzumelden, geben Sie im Feld **Benutzername** "Guest" ein und klicken Sie dann auf **OK**.



„Guest“ hat standardmäßig kein Passwort, das Feld **Passwort** kann also leer bleiben. Wenn das Passwort allerdings geändert wurde oder Sie Ihre Sicherheitseinstellungen von einem Server (Verteilt oder Domäne) erhalten, müssen Sie ein Passwort für „Guest“ eingeben.

- Zum Abmelden klicken Sie einfach auf **Abmelden**.

  Der Standardbenutzer (normalerweise „Guest“, dies kann aber im Hauptdialog Sicherheitssystem auf Seite 430 geändert werden) wird automatisch anstelle Ihres Benutzers angemeldet.

HINWEIS

Wenn der Sicherheitsmodus auf Seite 407 auf Domäne (LDAP) eingestellt ist und ein auf dem LDAP-Server erstellter Benutzer beim ersten Anmelden an der Domäne sein Passwort ändern muss, so muss er dies durchführen, bevor er sich am "WinStudio"-Projekt anmelden kann.

12.11 Sperren / Entsperren eines Benutzers

Ein einzelner Benutzer im Sicherheitssystem des Projekts kann vollständig für den Zugriff auf das Projekt gesperrt werden und ein gesperrter Benutzer kann später entsperrt werden.

Ein Benutzer kann auf folgende Arten gesperrt werden:

- durch manuelles Auswählen der Option **Benutzer geblockt** im Dialog Benutzerkonto auf Seite 428;
- durch Aufrufen der Funktion BlockUser auf Seite 610 während der Laufzeit oder
- automatisch, wenn der Benutzer zu oft ein falsches Passwort eingibt. (Die Anzahl der zulässigen Versuche wird im Dialog Gruppenkonto auf Seite 423 konfiguriert.)

Zur Überprüfung, ob ein Benutzer gesperrt ist, rufen Sie während der Laufzeit die Funktion GetUserState auf Seite 617 auf.

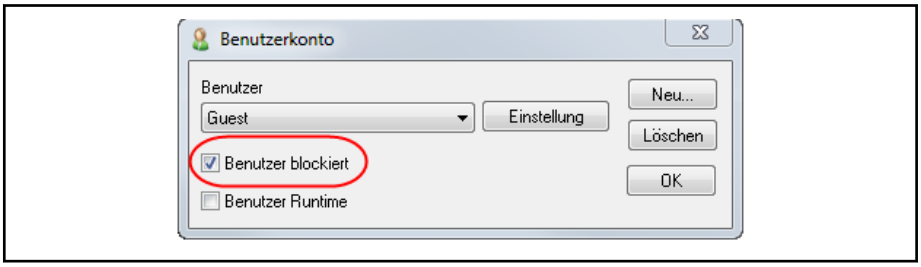


Abb.12-31: Benutzer entsperrt

Zum Entsperren eines Benutzers wählen Sie eine der folgenden Vorgehensweisen:

- Deaktivieren Sie die Option **Benutzer gesperrt** im "Dialog Benutzerkonto" oder
- Rufen Sie während der Laufzeit die Funktion UnblockUser auf Seite 620 auf.

12.12 Passwortschutz für Bilder, Symbole und Arbeitsblätter

Bilder, Symbole und Arbeitsblätter im Projektexplorer können mit einem Passwort geschützt werden. Sie können jeder Datei ein eigenes Passwort zuweisen oder allen Dateien im Projekt ein einziges Passwort zuweisen.

Fast alle Projektdateien werden automatisch verschlüsselt, um unbefugte Auswertung durch fremde Hilfsprogramme zu verhindern. (Bilddateien werden nicht verschlüsselt, weil die Entschlüsselung während der Laufzeit die Leistung einschränken würde.)

Sie können aber Ihre Dateien zusätzlich mit einem Passwortschutz versehen, um unbefugtes Ändern oder Weiterverwenden durch andere "WinStudio"-Projektentwickler zu verhindern.

Bei diesen Passwörtern muss immer auf Groß-/Kleinschreibung geachtet werden.

Zuweisen eines Passworts für eine Einzeldatei

Zuweisen eines Passworts für eine einzelne Projektdatei:

1. Suchen Sie im Projektexplorer die gewünschte Datei und führen Sie darauf einen Rechtsklick aus. Klicken Sie dann im Kontextmenü auf **Passwortschutz**. Der Dialog "Passwortschutz bearbeiten" wird angezeigt.
2. Geben Sie das neue Passwort ein und geben sie es erneut zur Bestätigung ein.
3. Schließen Sie den Dialog durch Klicken auf **OK**.

Die Datei ist jetzt geschützt. Beim nächsten Öffnen werden Sie zur Eingabe des Passwort aufgefordert.

Entfernen des Passworts von einer Einzeldatei

Entfernen eines Passworts von einer einzelnen Projektdatei:

1. Suchen Sie im Projektexplorer die gewünschte Datei und führen Sie zum Öffnen einen Doppelklick darauf aus. Sie werden zur Eingabe des Passwort aufgefordert.
2. Während die Datei zum Bearbeiten geöffnet ist, führen Sie im Projektexplorer einen Rechtsklick auf die Datei aus und klicken dann im Kontextmenü auf **Passwortschutz**. Der Dialog "Passwortschutz bearbeiten" wird angezeigt.
3. Lassen Sie die Felder **Neues Passwort** und **Passwort bestätigen** leer.
4. Schließen Sie den Dialog durch Klicken auf **OK**.

Die Datei ist nicht mehr geschützt. Die Datei kann ohne Aufforderung zur Eingabe des Passworts geöffnet werden.

Zuweisen eines Passworts für alle Dateien

Zuweisen eines einzelnen Passworts für alle Dateien in Ihrem Projekt:

1. Wählen Sie im Kontextmenü von WinStudio den Eintrag **Anwendung verifizieren**. Der Dialog "Anwendung verifizieren" wird angezeigt.
2. Klicken Sie auf **Passwort für alle Dateien festlegen**. Der Dialog "Passwortschutz bearbeiten" wird angezeigt.
3. Geben Sie das aktuelle Passwort für Ihr Projekt ein, falls vorhanden.
4. Geben Sie das neue Passwort ein und geben sie es erneut zur Bestätigung ein.
5. Klicken Sie auf **OK**. Der Überprüfungsvorgang wird fortgesetzt.
6. Schließen Sie den Dialog "Projekt überprüfen" durch Klicken auf **OK**.

Nun sind alle Dateien in Ihrem Projekt geschützt Beim nächsten Öffnen einer Datei werden Sie zur Eingabe des Passwort aufgefordert.

Entfernen des Passworts von allen Dateien

Zuweisen eines einzelnen Passworts für alle Dateien in Ihrem Projekt:

1. Wählen Sie im Kontextmenü von WinStudio den Eintrag **Anwendung verifizieren**. Der Dialog "Anwendung verifizieren" wird angezeigt.
2. Klicken Sie auf **Passwort für alle Dateien festlegen**. Der Dialog "Passwortschutz bearbeiten" wird angezeigt.
3. Geben Sie das aktuelle Passwort für Ihr Projekt ein.
4. Lassen Sie die Felder **Neues Passwort** und **Passwort bestätigen** leer.
5. Klicken Sie auf **OK**. Der Überprüfungsvorgang wird fortgesetzt.
6. Schließen Sie den Dialog "Anwendung verifizieren" durch Klicken auf **OK**.

Ihre Projektdateien sind nicht mehr geschützt.

13 Übersetzung von Anwendertexten

13.1 Übersetzung von Anwendertexten, Allgemeines

Sie können alle Texte, die Sie in WinStudio-Bildern, Bildobjekten, Alarmmeldungen, usw verwenden in andere Sprachen übersetzen und zur Laufzeit zwischen den verschiedenen Sprachen umschalten. Um in Ihrem Projekt eine Sprachumschaltung zu ermöglichen, sind folgende Arbeitsschritte notwendig:

- Hinzufügen von Projektsprachen zu Ihrem IndraWorks-Projekt
- Übersetzen Ihrer Anwendungstexte in die hinzugefügten Sprachen
- Aktivierung einer Sprache zur Laufzeit

Die einzelnen Schritte werden in den folgenden Kapiteln beschrieben.

13.2 Hinzufügen einer Projektsprache

Die Übersetzungstabelle dient der Verwaltung der Sprachen, in die Sie Ihr Projekt übersetzen möchten. Die Liste der auswählbaren Sprachen entspricht den Sprachen, die in Ihrem IndraWorks-Projekt verfügbar sind.

Um eine Sprache hinzuzufügen, gehen Sie folgendermaßen vor:

1. Wählen Sie im Hauptmenü von IndraWorks den Eintrag **Projekt ▸ Sprache ▸ Projektsprachen verwalten**

   Es öffnet sich ein Dialog, in dem Sie mit der Schaltfläche "Hinzufügen" die gewünschten Sprachen hinzufügen können.

2. Öffnen Sie im IndraWorks Projektexplorer das Arbeitsblatt "Lokalisierung" im Ordner "WinStudio\Tasks". Das Arbeitsblatt "Lokalisierung" zeigt im Tabellenteil in der Spalte Quelle alle übersetzbaren Zeichenketten aus Ihrem Projekt an.

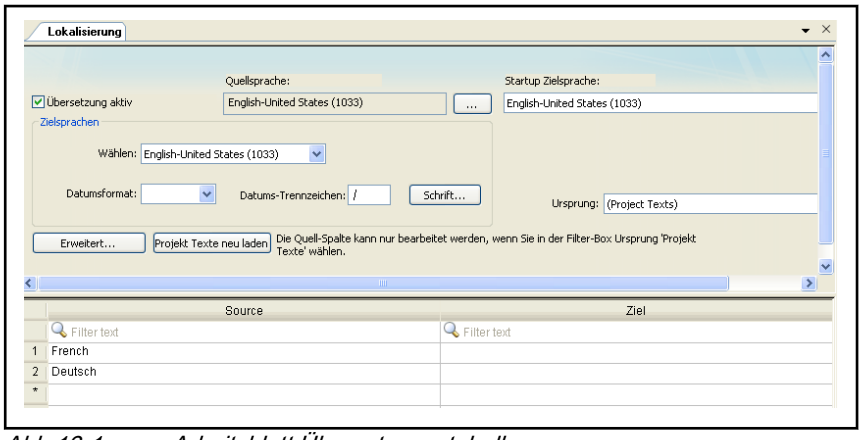


Abb. 13-1: Arbeitsblatt Übersetzungstabelle

3. Wählen Sie im Bereich "Zielsprachen" die Sprache aus, in die Sie Ihre Anwendungstexte übersetzen möchten. Die Liste der auswählbaren Sprachen entspricht den Projektsprachen im IndraWorks-Projekt.
4. Vergewissern Sie sich, dass die Option **Übersetzung aktivieren** ausgewählt ist.
5. Konfigurieren Sie **Datumsformat** und **Datumstrennzeichen** wie gewünscht für die Zielsprache. So ist zum Beispiel das **Datumsformat** für "English-United States" (d.h. amerikanisches Englisch) normalerweise "MTJ" und das **Datumstrennzeichen** ist normalerweise "/", was das Datumsformat "MM/TT/JJJJ" ergibt.

13.3 Übersetzen Ihrer Anwendungstexte in weitere Sprachen

Nachdem Sie Ihrem Projekt die gewünschten Projektsprachen hinzugefügt haben, können Sie diese in der WinStudio-Anwendung verwenden und die Anwendungstexte in die jeweiligen Sprachen übersetzen.

Zum Übersetzen der Anwendungstexte gehen Sie wie folgt vor:

1. Öffnen Sie das Arbeitsblatts "Lokalisierung" im Ordner "WinStudio \Tasks". Das Arbeitsblatt "Lokalisierung" zeigt im Tabellenteil in der Spalte Quelle alle übersetzbaren Zeichenketten aus Ihrem Projekt an.
2. Wählen Sie im Bereich "Zielsprachen" die Sprache aus, in die Sie Ihre Anwendungstexte übersetzen möchten. Die Liste der auswählbaren Sprachen entspricht den Projektsprachen im IndraWorks-Projekt.
3. Tragen Sie in der Spalte "Ziel" die Übersetzungen der Originaltexte für die ausgewählte Zielsprache ein. Um nur die Texte aus einem bestimmten Kontext Ihrer WinStudio-Anwendung zu bearbeiten, wählen Sie aus der Auswahlliste "Ursprung" den entsprechenden Typ aus, z.B. AlarmControl.
4. Verwenden Sie die Felder **Filtertext** zum Durchsuchen des Arbeitsblatts nach einer Zeichenkette; bei Eingabe einiger Zeichen wird die Liste dynamisch gefiltert und zeigt nur die passenden Zeichenketten an.
5. Speichern und schließen Sie das Arbeitsblatt.

Änderungen in der Übersetzungstabelle werden erst bei Aufruf der Funktion Einstellen der Projektsprache während der Laufzeit auf Seite 439 oder bei Neustart des Laufzeitprojekts wirksam.



Die Übersetzungstabelle wird als tab-getrennte Textdatei in Ihrem Projektordner gespeichert, unter

[…]\projekt_name\Web\Translation.trn.

Diese Datei kann bei Bedarf mit Microsoft Excel geöffnet und direkt bearbeitet werden.

13.4 Aktivierung einer Sprache zur Laufzeit

13.4.1 Aktivierung einer Sprache zur Laufzeit, Allgemein

Bei der Aktivierung einer Projektsprache zur Laufzeit muss grundsätzlich der Applikationstyp berücksichtigt werden:

- **HMI- oder MTX-Applikation**

  Falls Sie eine HMI- oder MTX-Applikation projektieren, Ihre Anwendung also innerhalb von Indraworks Operation ausgeführt wird, müssen Sie für die Aktivierung einer Sprache keine weiteren Konfigurationen vornehmen. Die Applikation ist für die Sprachumschaltung bereits vorbereitet und Sie können in IndraWorks Operation die aktive Sprache entweder über das Hauptmenü unter **Extras** ▸ **Optionen** oder eine F-/M-Taste ändern.

  Die ausgewählte Sprache gilt hier auch für alle WinStudio-Bilder.

- **WinStudio-Applikation**

  Bei einer WinStudio-Applikation müssen Sie folgende Schritte ausführen, um zur Laufzeit eine Sprache zu aktivieren:

  1. Wählen Sie eine Startsprache aus, die beim Start Ihrer Anwendung aktiviert wird.

2. Fügen Sie Steuerelemente an geeigneter Stelle in Ihre Anwendung ein, um die aktive Sprache zur Laufzeit zu ändern.

Die beiden Schritte werden in den folgenden Kapiteln erläutert.

13.4.2 Einstellen der Projektsprache bei Programmstart

Wenn mehrere Sprachen für Ihr Projekt konfiguriert sind, müssen Sie festgelegen, mit welcher Sprache Ihr Projekt zur Laufzeit starten soll.

Dieser Vorgang setzt voraus, dass Sie bereits mindestens eine Zielsprache zur Übersetzungstabelle hinzugefügt haben.

Weitere Informationen unter Sprache hinzufügen auf Seite 437.

1. Öffnen Sie die Übersetzungstabelle, indem Sie im Projektexplorer einen Doppelklick auf das Element "Lokalisierung" im Ordner "WinStudio \Tasks" ausführen.

   Das Arbeitsblatt Übersetzungstabelle wird zum Bearbeiten geöffnet.
2. Wählen Sie aus der Liste **Zielsprache für Start** die Sprache, mit der das Projekt starten soll. Die Liste der verfügbaren Sprachen beinhaltet die Quellsprache, in der Sie das Projekt entwickelt haben, und alle hinzugefügten Zielsprachen.
3. Speichern und schließen Sie das Arbeitsblatt.

13.4.3 Einstellen der Projektsprache während der Laufzeit

Für Aktivierung einer Projektsprache bietet WinStudio das Kommando SetLanguage auf Seite 685 an. Mit Hilfe dieser Funktion kann die Projektsprache während der Laufzeit überall dort eingestellt werden, wo ein Ausdruck konfiguriert werden kann.

Die Verwendung dieser Funktion setzt voraus, dass Sie bereits mindestens eine Zielsprache zur Übersetzungstabelle hinzugefügt haben. Weitere Informationen unter Sprache hinzufügen auf Seite 437.

Die Funktion "SetLanguage" erfordert als Parameter einen numerischen Code für die Zielsprache. Die Codes werden in der Liste der Sprachen in Klammern angezeigt – z.B. "1033" für "English-United States".

Die folgenden, sehr einfachen Beispiele erläutern das Zeichnen von zwei Schaltflächenobjekten zum Umschalten der Projektsprache zwischen Englisch und Französisch.

1. Klicken Sie im Ordner "WinStudio" auf ein WinStudio-Bild, um dieses zum Bearbeiten zu öffnen.
2. Klicken Sie im Werkzeugfenster "WinStudio Screen Tools" auf **Schaltfläche**.
3. Zeichnen Sie in das WinStudio-Bild ein Schaltflächenobjekt.
4. Führen Sie einen Doppelklick auf das Schaltflächenobjekt aus. Der Dialog "Objekteigenschaften" wird angezeigt.
5. Geben Sie im Feld **Beschriftung** "Englisch" ein.
6. Klicken Sie auf **Befehl**. Die Eigenschaften für die Befehlsanimation werden im Dialog angezeigt.
7. Geben Sie in der ersten Zeile der Registerkarte **Beim Drücken** im Feld **Ausdruck** `SetLanguage(1033)` ein.

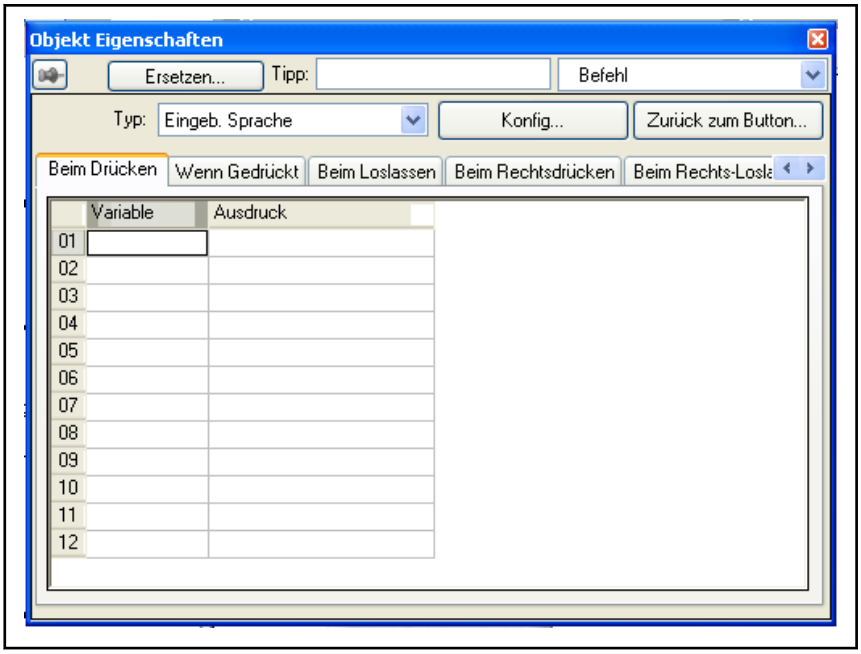


Abb. 13-2: Einstellen der Projektsprache (1)

8. Schließen Sie den Dialog "Objekteigenschaften".
9. Duplizieren Sie das Schaltflächenobjekt mit Kopieren und Einfügen oder mit Strg+Klicken.
10. Wiederholen Sie die Schritte 4 bis 8 und ersetzen Sie die Beschriftung durch "Französisch" und den Ausdruck durch `SetLanguage(1036)`.

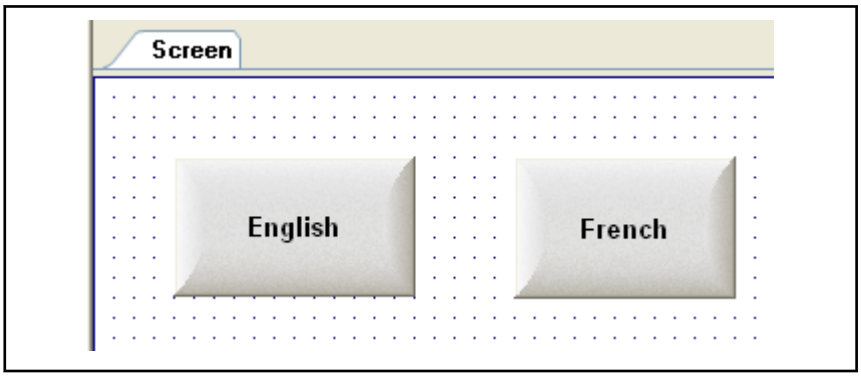


Abb. 13-3: Einstellen der Projektsprache (2)

11. Speichern und schließen Sie das WinStudio-Bild.

Während der Projektlaufzeit wird durch Klicken auf die jeweilige Schaltfläche die Sprache des gesamten Projekts in dieser Sprache geändert, unter Verwendung der übersetzten Texte aus der Übersetzungstabelle.

13.5 Übersetzung ausgewählter Bildobjekte deaktivieren

Standardmäßig ist die Übersetzung für alle Bildobjekte mit übersetzbarem Text aktiviert. Die Übersetzung kann aber für ausgewählte Objekte deaktiviert werden, wenn deren Originaltext erhalten bleiben soll.

1. Mit einem Doppelklick auf das Bildobjekt öffnet sich der Dialog "Objekteigenschaften".
2. Suchen Sie im Dialog die Option **Übersetzung aktivieren**. Wenn die Option in den grundlegenden Eigenschaften des Objekts nicht verfügbar

ist, klicken Sie auf **Erweiterte Eigenschaften** zum Öffnen der erweiterten Eigenschaften.

3. Deaktivieren Sie die Option **Übersetzung aktivieren**.
4. Schließen Sie den Dialog "Objekteigenschaften".

Wenn diese Option für ein Objekt deaktiviert ist, wird dessen Text während der Projektlaufzeit nicht mehr übersetzt.

Der Text wird trotzdem zur Übersetzungstabelle hinzugefügt und mit allen anderen Projekttexten mit der automatischen Übersetzungsfunktion verarbeitet, die entsprechende Übersetzung wird aber während der Laufzeit nicht auf das Objekt angewendet.

14 Test und Fehlerbeseitigung für Projekte

14.1 Ausgabe (LogWin)

Das Fenster "WinStudio Log" können Sie verwenden, um Fehleranalysemeldungen anzuzeigen, die während der Projektlaufzeit erzeugt werden. Das Fenster zeigt OPC-, DDE- und TCP/IP-Transaktionen, Modulaktivierungen, Trace-Meldungen usw. an.

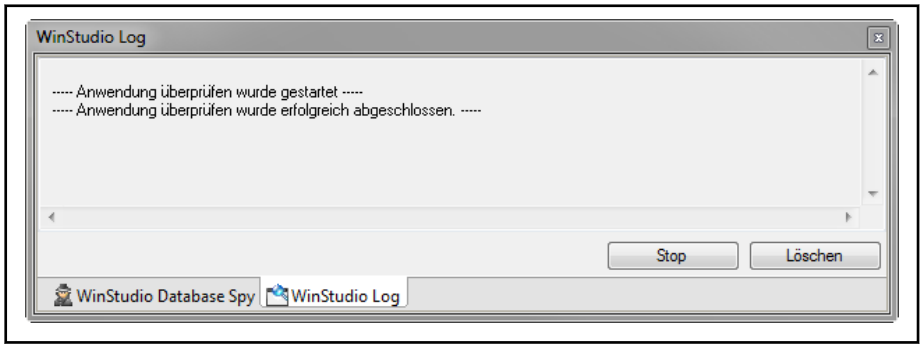


Abb. 14-1: Beispiel-Fenster Ausgabe

Das Fenster "WinStudio Log" ist andockfähig, d.h. es kann an eine andere Stelle in der Entwicklungsumgebung verschoben werden. Klicken Sie dazu auf die Titelleiste und ziehen Sie das Fenster an eine neue Position. Lassen Sie die Maustaste los, um das Fenster an seinem neuen Platz zu befestigen bzw. "anzudocken".

Konfigurieren des Protokolls zur Anzeige zusätzlicher Informationen

Standardmäßig zeigt das Protokoll nur Fehleranalyse- und Fehlermeldungen an – also Meldungen, aus denen hervorgeht, dass Ihr Projekt nicht korrekt ausgeführt wird. Würde das Protokoll **alle** von "WinStudio" erzeugten Meldungen anzeigen, wäre die Ausgabe schnell unübersichtlich.

Um die im Protokoll angezeigten Meldungen zu konfigurieren gehen Sie folgendermaßen vor:

1. Führen Sie an einer beliebigen Stelle im Fenster "WinStudio Log" einen Rechtsklick aus und klicken Sie dann im Kontextmenü auf **Einstellungen**. Der Dialog "Logging-Eigenschaften" wird angezeigt:
2. In der Registerkarte "Log-Optionen" des Dialogs wählen Sie die Meldungsarten aus, die im Protokoll angezeigt werden sollen.

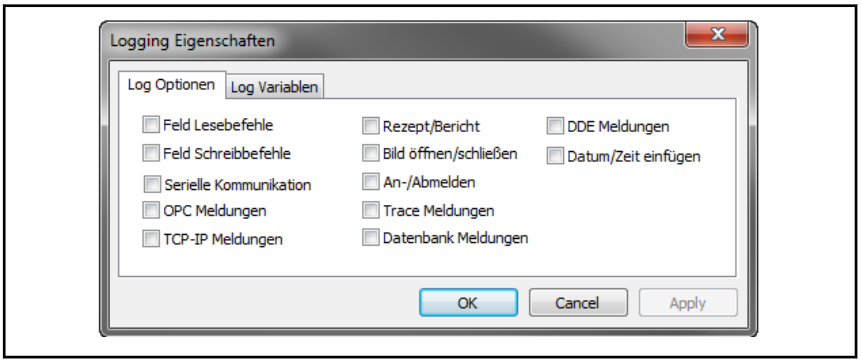


Abb. 14-2: Protokoll-Einstellungen – Registerkarte Optionen

Test und Fehlerbeseitigung für Projekte

Optionen	Beschreibung
Feld-Lesebefehle	Zeigt alle Feld-Lesebefehle, die an angeschlossene Geräte geschickt werden.
Feld-Schreibbefehle	Zeigt alle Feld-Schreibbefehle, die an angeschlossene Geräte geschickt werden.
Serielle Kommunikation	Zeigt von konfigurierten Gerätetreibern auf Seite 379 erzeugte Meldungen an.
OPC-Meldungen	Zeigt Meldungen der OPC-Kommunikation auf Seite 390 an.
TCP/IP-Meldungen	Zeigt Meldungen der TCP/IP-Kommunikation auf Seite 401 an.
Rezept/Bericht	Zeigt Meldungen der Tasks für Rezepturen auf Seite 361 und Berichte auf Seite 363 an.
Öffnen/Schließen anzeigen	Zeigt detaillierte Informationen an, wenn ein Bild auf Seite 170 geöffnet oder geschlossen wird: **Festplatten-Ladezeit:** Die Zeit, um die Bilddatei von der Festplatte in den Speicher zu laden. **Öffnungszeit:** Zeit zum Laden des WinStudio-Bild, einschließlich der im Bild verwendeten Initialisierungs-Tags und dem Ausführen von ggf. vorhandenen „OnOpen“-Skripten oder -Funktionen. **Gesamt-Ladezeit:** Gesamtzeit zum Laden des Bilds (beinhaltet die obige Festplatten- und die Öffnungsladezeit). **"Erstes-Zeichnen"-Zeit:** Zeit zum erstmaligen Zeichnen der Bildobjekte. **"Erstes-Onwhile"-Zeit:** Zeit für das erstmalige Ausführen von ggf. vorhandenen „OnWhile“-Scripten oder -Funktionen. **Gesamt-Öffnungszeit:** Zeit insgesamt zum Öffnen des Bildes (beinhaltet die obige Erstes-Zeichnen- und Erstes-OnWhile-Zeit). **Schließzeit:** Zeit zum Schließen des Bildes, einschließlich der im Bild verwendeten Abschluss-Tags und dem Ausführen von ggf. vorhandenen „OnClose“-Scripten oder -Funktionen. **Gesamt-Schließzeit:** Gesamtzeit zum Schließen des Bildes, einschließlich der Zeit zum Schreiben der Bilddatei auf die Festplatte. Diese Informationen lassen sich zum Analysieren der Laufzeitleistung auf schwach ausgestatteten Endsystemen verwenden. Wenn ein bestimmter Schritt beim Öffnen oder Schließen ungewöhnlich lange braucht, kann er gefunden und neu konzipiert werden.
Anmelden/Abmelden	Anzeigen einer Meldung, wenn sich ein Benutzer an- oder abmeldet. (Weitere Informationen erhalten Sie in Sicherheit auf Seite 407.)
Trace-Meldungen	Zeigt Meldungen der Funktion **Trace()** an. Diese Funktion wird zum Erzeugen kundenspezifischer Meldungen aus Ihrem Projekt verwendet, um die Ausführung der Anwendung nachzuverfolgen.
Datenbankmeldungen	Zeigt Meldungen der ODBC- auf Seite 364 und ADO.NET- auf Seite 371 Datenbankschnittstellen an.
DDE-Meldungen	Zeigt Meldungen der DDE-Kommunikation auf Seite 403 an.
Datum/Uhrzeit einfügen	Jede Meldung mit einem Zeitstempel versehen.

Abb. 14-1: Protokoll-Einstellungen – Registerkarte Optionen

3. Klicken Sie in der Registerkarte "Log Variablen" des Dialogs auf **Hinzufügen**, um nach Projekt-Tags auf Seite 116 zu suchen.

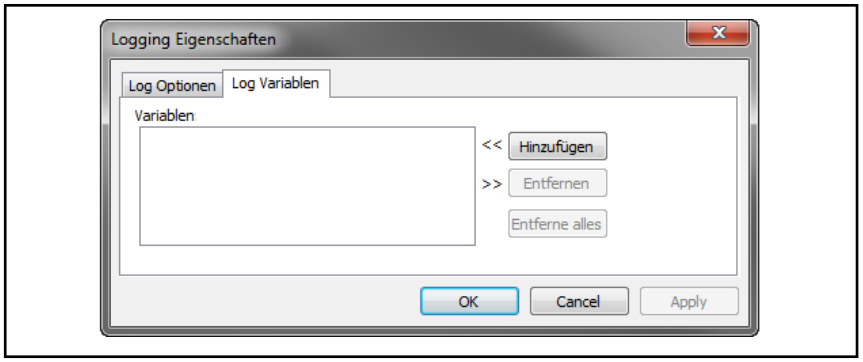


Abb. 14-3: Protokoll-Einstellungen – Registerkarte Tags

Der Protokollbereich des LogWin-Fensters zeigt eine Meldung an, wenn sich der Wert eines ausgewählten Tags ändert.

4. Klicken Sie auf **OK**, um Ihre Einstellungen zu speichern und den Dialog "Logging-Eigenschaften" zu schließen.

14.2 Datenbankspion

Das Fenster "Datenbankspion" ist ein Fehleranalysewerkzeug für die Anzeige und Änderung von Tag-Werten sowie für Ausführung und Test von Funktionen und mathematischen Ausdrücken.



Abb. 14-4: Beispiel für ein Datenbankspion-Fenster

Das Fenster zeigt die folgenden Informationen an:

- **Name**: Angeben einer Anwendungsvariablen, einer internen Variablen oder eines Ausdrucks, der überwacht werden soll.
- **Wert**: Anzeige des Werts, der vom Tag/Ausdruck zurückgegeben wird.
- **Qualität**: Anzeige der Qualität (GUT oder SCHLECHT) des Werts, der vom Tag/Ausdruck zurückgegeben wird.
- **Stetig**: Wählen Sie diese Option, wenn das Projekt den Tag/Ausdruck fortlaufend auswerten soll.

Um eine Variable zu überwachen, tragen Sie in der Spalte deren Namen ein oder klicken Sie in ein leeres Feld in der Spalte "Name" und öffnen Sie das Kontextmenü. Über die Funktion "Variable oder Ausdruck einfügen" können Sie alternativ den Browserdialog öffnen und eine Variable auswählen.



Der Datenbankspion ist andockfähig, d.h. er kann an eine andere Stelle in der Entwicklungsumgebung verschoben werden.

Klicken Sie auf die Titelleiste, und ziehen Sie das Fenster an eine neue Position.

Lassen Sie die Maustaste los, um das Fenster an seinem neuen Platz zu befestigen bzw. anzudocken.

14.3 Verwenden des LogWin-Moduls

Dieses Modul liefert eine kontinuierliche Aufzeichnung von Aktivitäten und Tags zur Fehlerbeseitigung über längere Zeiträume. Es erstellt eine Datei, in der die Ergebnisse der Datensammlung gespeichert werden können. Die Größe dieser Datei nimmt zu, bis Sie den Protokollierungsvorgang (Datensammlung) stoppen.

Verwenden Sie das LogWin-Modul (lokal und fern) zum Aufzeichnen von DDE- auf Seite 403, OPC- auf Seite 390 und TCP/IP- auf Seite 401 Transaktionen, Aktivieren von Modulen, Aufzeichnen von Tags usw.

Aktivieren des LogWin-Moduls als Bestandteil Ihres Projekts:

1. Wählen Sie im Kontextmenü des Ordners "WinStudio " den Eintrag **Werkzeuge** ▸ **Status**. Der Dialog "Ausführung Tasks" wird angezeigt:

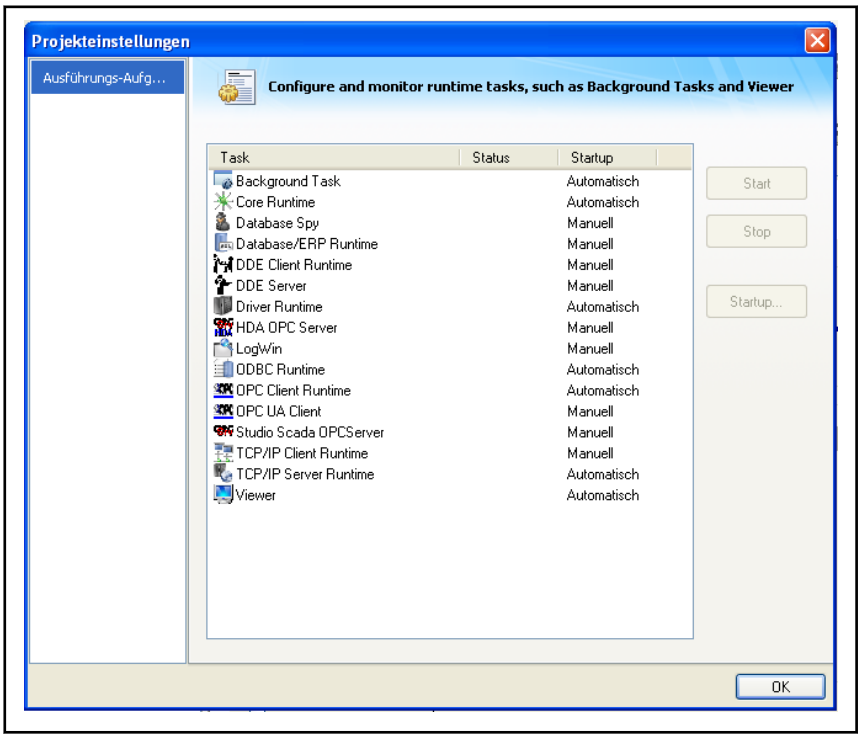


Abb. 14-5: Dialog Ausführung Tasks

2. Wählen Sie LogWin aus der Liste **Task**.
3. Nach Auswahl des LogWin-Tasks klicken Sie auf **Starten**. Der Dialog "Start" wird angezeigt.

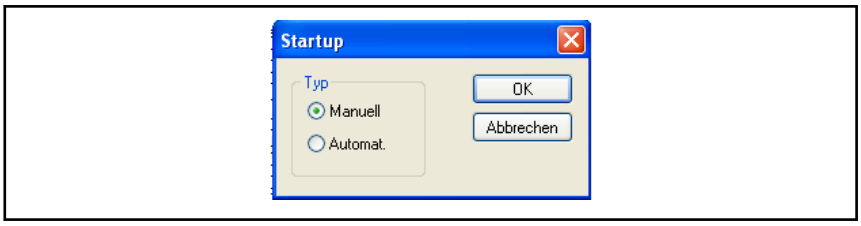


Abb. 14-6: Dialog Starten

4. Klicken Sie auf das Optionsfeld **Automatisch**.
5. Klicken Sie auf **OK**, um Ihre Auswahl zu speichern und den Dialog "Starten" zu schließen.
6. Schließen Sie den Dialog "Ausführung Tasks", indem Sie auf **OK** klicken.

- Um das LogWin-Modul direkt auf Ihrem Windows Embedded Gerät in zu starten, wählen Sie aus der Menüleiste von CEView den Eintrag **Werkzeuge ▸ LogWin**.
- Zum Aktivieren der Protokollierung in einemThin Client auf Seite 449 verwenden Sie die Funktion **Log** in den Projekteinstellungen "Web". Den Dialog für die Web-Einstellungen finden Sie im Kontextmenü von WinStudio unter **Werkzeuge ▸ Web-Einstellungen**.

14.4 Verwenden von Remote-Werkzeugen

14.4.1 Remote-Datenbankspion

Das Werkzeug **Remote-Datenbankspion** kann zum Überwachen des Datenbankspions auf Seite 93 eines auf einem Remote-Rechner ausgeführten "WinStudio"-Projekts verwendet werden. Der Ausführungstask "Datenbankspion" muss für das Projekt aktiviert sein und der Remote-Rechner muss sich in der Laufzeit auf Seite 65 befinden.

Verwendung des Werkzeugs "Remote-Datenbankspion":

1. Im Kontextmenü von WinStudio klicken Sie auf **Werkzeuge ▸ Remote-Datenbankspion**. Der Dialog "Remote-Rechner" wird angezeigt.
2. Geben Sie wie unten gezeigt die IP-Adresse des Remote-Rechners ein.

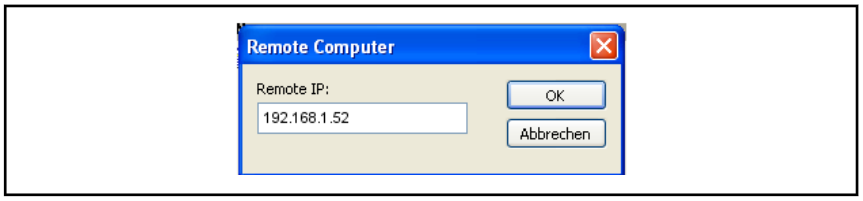


Abb. 14-7: Eingabe einer Remote IP

Die IP-Adresse **192.168.1.52** ist nur ein Beispiel. Bitte überprüfen Sie die IP-Adresse das Rechners, mit dem sie eine Verbindung herstellen wollen.

3. Klicken Sie auf **OK**, um eine Verbindung zur angegebenen Adresse herzustellen. Bei korrekter Verbindung wird das Fenster "Remote-Datenbankspion" angezeigt.

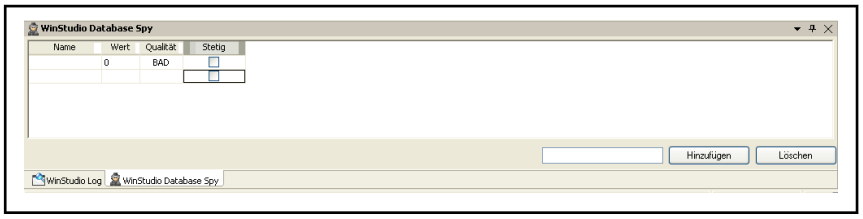


Abb. 14-8: Remote-Datenbankspion

Tags können nicht über die Remote-Verwaltung hinzugefügt oder entfernt werden. Die Tag-Liste des Datenbankspions muss auf dem Remote-Rechner selbst konfiguriert werden.

Nach Beendigung klicken Sie auf **Schließen**, um die Verbindung mit dem Remote-Rechner zu trennen.

Test und Fehlerbeseitigung für Projekte

14.4.2 Remote LogWin

Das Werkzeug **Remote-LogWin** kann zum Überwachen des Ausgabeprotokolls (LogWin), Seite 93 eines auf einem Remote-Rechner ausgeführten "WinStudio"-Projekts verwendet werden. Der Ausführungstask "LogWin" muss für das Projekt aktiviert sein und der Remote-Rechner muss sich in der Laufzeit, Seite 65 befinden.

Verwendung des Werkzeugs Remote-LogWin:

1. Im Kontextmenü von WinStudio klicken Sie auf **Werkzeuge** ▸ **Remote-LogWin**. Der Dialog "Remote-Rechner" wird angezeigt.
2. Geben Sie wie unten gezeigt die IP-Adresse des Remote-Rechners ein.

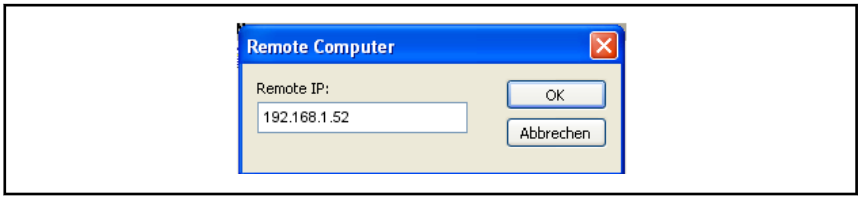


Abb. 14-9: Eingabe einer Remote IP

Die IP-Adresse **192.168.1.52** ist nur ein Beispiel. Bitte überprüfen Sie die IP-Adresse das Rechners, mit dem sie eine Verbindung herstellen wollen.

3. Klicken Sie auf **OK**, um eine Verbindung zur angegebenen Adresse herzustellen. Bei korrekter Verbindung wird das Fenster "Remote LogWin" angezeigt.

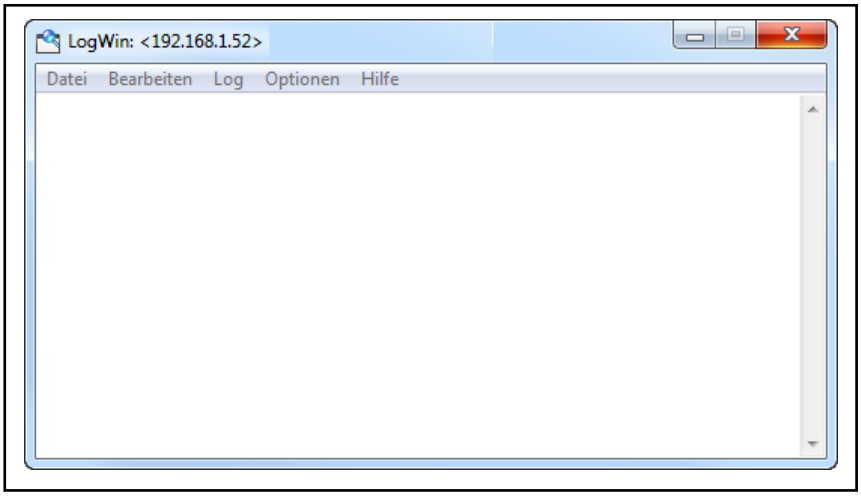


Abb. 14-10: Remote LogWin

Nach Beendigung schließen Sie das Fenster, um die Verbindung zu trennen.

15 Bereitstellung als Web-Anwendung

15.1 Einführung in Thin Clients

"WinStudio" baut auf einer Client/Server-Architektur auf und kann sowohl Thin Clients als auch Thick Clients unterstützen. Die Auswahl der Client-Architektur hängt von Ihren Systemanforderungen ab.

Thick Client

Ein Thick (Fat) Client ist ein Computer, der die meisten, wenn nicht sogar alle, Rechenarbeiten ausführt.

Der Thick Client verfügt über ausreichend Rechenleistung, Speicher, Grafik usw. zum Ausführen des Projekts. Der Thick Client kann nach Bedarf Daten mit dem Server austauschen (z.B. Datenarchivierung, Programmsynchronisierung). Ein Server kann auch ein Thick Client für einen anderen Server sein.

Thin Client

Ein Thin (Lean) Client ist ein Computer, der für Rechenarbeiten hauptsächlich vom zentralen Server abhängig ist. Der Thin Client benötigt einen Prozessor mit minimalem Speicher, einen Browser (Internet Explorer) und eine Netzwerkverbindung (z.B. mit dem Internet oder Intranet). Wir verwenden den Begriff „Web Client“ für eine Thin-Client-Lösung. Mit "WinStudio" können Sie den Browser auswählen, der am besten für Ihre Laufzeitumgebung geeignet ist.

- Microsoft Internet Explorer. Diese Art von Browser ist am besten für Desktop-Umgebungen und Umgebungen mit Remote-Zugriff geeignet.
- Secure Viewer von "Bosch Rexroth", ein Viewer speziell für "WinStudio"-Projekte. Mit Secure Viewer kann der Benutzer nicht auf andere Websites wechseln und er kann sich nur mit einem festgelegten "WinStudio"-Server verbinden. Secure Viewer ist für Produktionsumgebungen bestimmt, wo das Besuchen anderer Websites nicht erforderlich oder nicht erwünscht ist.

In diesem Kapitel wird die Verwendung von Thin Clients mit "WinStudio" betrachtet. Thick Clients werden in einem späteren Kapitel behandelt.

Thin Clients in "WinStudio"

Mit "WinStudio" können Bilder erstellt werden, die mit einem normalen Webbrowser auf einer Remote-Station visualisiert werden können (z.B. Internet Explorer). Die Station, auf der der Benutzer die grafische Schnittstelle (Bilder) auf dem Web Browser visualisieren kann, wird als Thin Client bezeichnet. Der Thin Client (Browser) ist der Host.

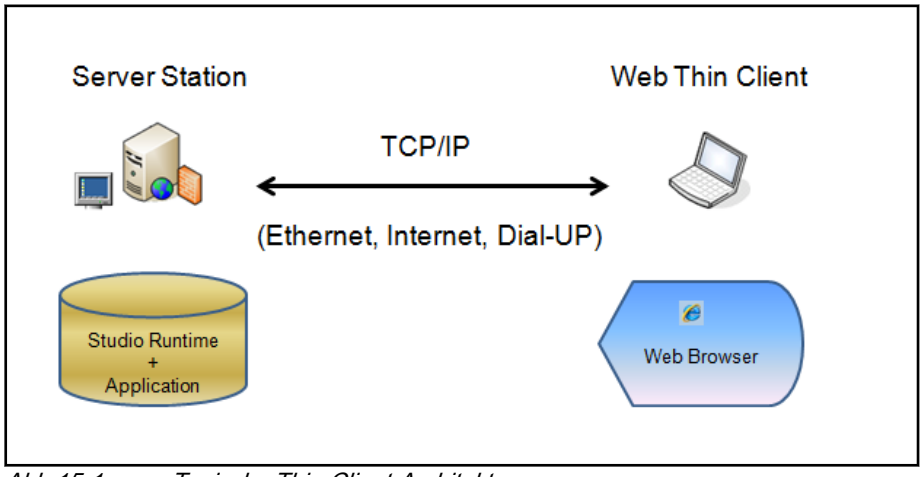


Abb.15-1: Typische Thin-Client-Architektur

Bereitstellung als Web-Anwendung

"WinStudio" wird nur auf der Serverstation installiert. Außerdem wird das Projekt (Bilddateien, Tag-Datenbank, Konfigurations-Arbeitsblätter etc.) nur auf dem Server gespeichert. Die Entwicklungsanwendung oder das Laufzeitprojekt müssen nicht auf dem Thin Client installiert werden. Diese Lösung bietet einen hohen Grad an Flexibilität, weil jeder physisch mit der Serverstation verbundene Rechner (über einen TCP/IP-Link) ohne Installation der Entwicklungsanwendung oder der Laufzeitprojekts auf dem Thin Client auf das Bild und die Online-/Historien-Daten vom Server zugreifen kann.

Alle Hintergrund-Tasks (Berechnung, Scheduler usw.) und Kommunikation-Tasks (z.B. Treiber, OPC, DDE TCP/IP) werden nur auf der Serverstation ausgeführt. Der Thin Client kann die auf dem Server konfigurierte grafische Schnittstelle (Bilder mit Objekten und Animationen) laden und die Online-Werte der im Server konfigurierten Tags sowie Historiendaten (Alarm-, Ereignis- und Trendhistoriendaten) anzeigen.

☞ Da VBScript und die "WinStudio" Skriptsprache sowohl mit einem Bild als auch mit einer Befehlsanimation für ein Objekt verbunden werden können, können diese Skripte auf einem Thin Client ausgeführt werden.

Alle Rechner oder Geräte (z.B. ein PDA mit Windows Mobile 5.0) mit Microsoft Internet Explorer 6.0 (oder höher) oder unserem Programm Secure Viewer können als Thin Client für Ihr Projekt verwendet werden. Außerdem stellt "WinStudio" ein hoch entwickeltes **Sicherheitssystem** zur Verfügung, um unbefugten Zugriff auf das Projekt zu verhindern.

Vom Thin Client aus können Sie Daten vom Server visualisieren sowie Sollwerte verändern, Alarme bestätigen und/oder Befehle an den Server schicken. Bei der Konfiguration des Projekts können optional alle Befehle vom Thin Client zum Server aktiviert oder deaktiviert werden. Auch wenn der Server alle Befehle vom Thin Client deaktiviert hat, kann der Thin Client immer noch Daten vom Server lesen, aber keine Daten an den Server schicken.

Thin Client – Wettbewerbsvorteile

"WinStudio" baut auf einer Client/Server-Architektur auf, die **echte** Thin Clients unterstützt. Diese Fähigkeit ist in "WinStudio" integriert und ist keine Zusatzfunktion. Das bedeutet:

- Der "WinStudio"-Server unterstützt eine große Zahl von gleichzeitig ablaufenden Thin Clients (bis zu 128 IE-basierte und 128 Secure-Viewer-basierte). Jeder Thin Client kann die gleichen oder andere Bilder betrachten wie ein anderer Thin Client.
- Der "WinStudio"-Server weiß, welche Anzeige jeder Thin Client betrachtet und „schiebt“ automatisch alle aktualisierten Tag-Werte zum Thin Client. So bleibt die Anzeige des Thin Client aktualisiert und das Bild muss nicht manuell aktualisiert werden.
- Der "WinStudio"-Server kann für jeden Thin Client Sprachwechsel während der Laufzeit unterstützen. Das bedeutet, dass ein Thin Client ein Bild in Englisch betrachtet, während ein anderer Thin Client das gleiche Bild in Spanisch anzeigt.

Viele Konkurrenzprodukte stellen entweder eine statische Anzeige auf einem Thin Client (d.h. zum Erhalts der neuesten Daten muss manuell aktualisiert werden), eine Terminal-Server-Lösung (der Server muss mehrere Instanzen des Projekts aufbauen, um jeden Thin Client zu unterstützen) oder eine Thin-Client-Lösung ähnlich "WinStudio" mit teuren Zusatzsoftwareprodukten zur Verfügung.

Andere Vorteile des Thin Client sind:

- optionaler Secure Viewer, der keine Navigation außerhalb des "WinStudio"-Projekts erlaubt. der Secure Viewer basiert nicht auf dem Microsoft Internet Explorer.
- Möglichkeit zur Ausführung von VBScript und der eingebauten Skriptsprache auf dem Thin Client
- Möglichkeit zum Aufbau von Thin-Client-Projekten mit Windows-Embedded-Geräten als Server.
- Fähigkeiten zur Unterstützung von redundanten Webservern und Datenservern, mit automatischer Umschaltung.

Lizenzierung für Thin Client

Die maximale Anzahl an Thin-Client-Stationen, die gleichzeitig mit dem Server verbunden sein können, hängt von den Einstellungen der auf dem Server installierten Lizenz ab. Der Benutzer muss keine Lizenzen auf dem Thin Client installieren. Weitere Informationen unter Lizenzeinstellungen auf Seite 47.

15.2 Erstellen eines einfachen Thin-Client-Programms

Wie im folgenden beschrieben, gibt es verschiedene Wege zum Aufbau eines Thin-Client-Programms. Mit der folgenden Vorgehensweise können Sie ein einfaches, ungesichertes Thin-Client-Programm entwickeln.

Vorgehensweise A: Thin Client verwendet NTWebServer, Local Loopback

Ein einfacher Weg zum Entwickeln und Testen eines ersten Thin-Client-Programms ist, den schlanken Webserver **NTWebServer** auf Ihrem Entwicklungsrechner zu verwenden, das Laufzeitprojekt zu starten und den Microsoft Internet Explorer auf dem gleichen Rechner zur Anzeige der Webseiten zu verwenden.

1. Schritt: Konfigurieren der "WinStudio"-Einstellungen

1. Öffnen Sie den Konfigurationsdialog für Web-Einstellungen, indem Sie im Kontextmenü von WInStudio den Eintrag **Werkzeuge ▸ Web-Einstellungen** wählen.

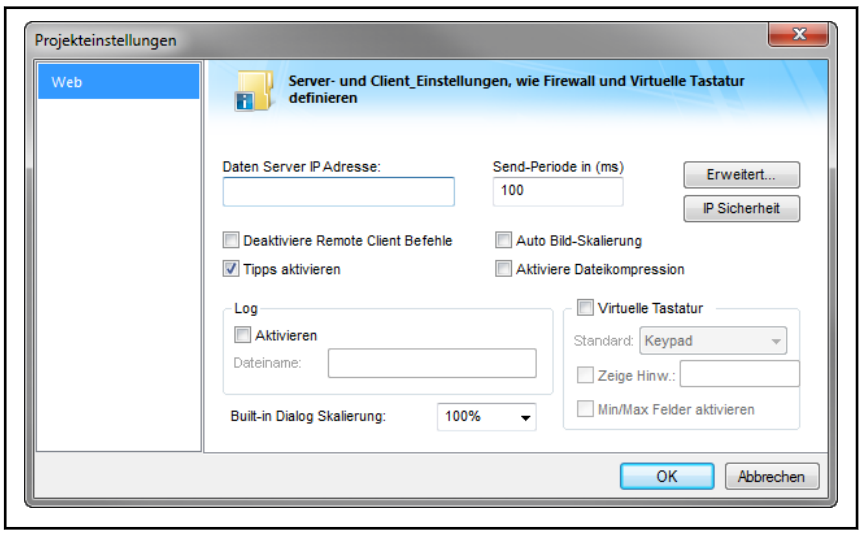


Abb. 15-2: WinStudio-Einstellungen, Thin Client

- Geben Sie im Feld IP-Adresse Datenserver **127.0.0.1** ein.
- Vergewissern Sie sich, dass die Kontrollkästchen **Automatische Bildskalierung**, **Dateikomprimierung aktivieren** und **ToolTips aktivieren** ausgewählt sind.

- Klicken Sie auf die Schaltfläche **IP-Sicherheit**. Vergewissern Sie sich, dass das Kontrollkästchen **Aktivieren** nicht aktiviert ist. Klicken Sie auf **OK**.
- Klicken Sie auf die Schaltfläche **Erweitert**. Vergewissern Sie sich, dass das Kontrollkästchen **Web Tunneling Gateway** nicht aktiviert ist.

  Klicken Sie auf **OK**.

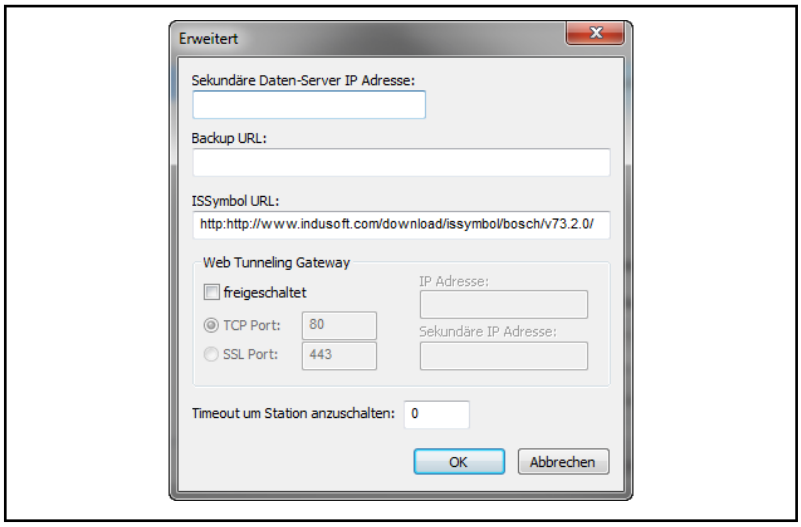


Abb. 15-3: WinStudio-Einstellungen, Thin Client, Weitere Einstellungen

2. Öffnen Sie die Task-Konfiguration (Eintrag **Werkzeuge ▸ Status** im Kontextmenü von WinStudio) und stellen Sie sicher, dass der Task **TCP/IP Server** auf Automatisch eingestellt ist.

2. Schritt: Datenbank konfigurieren

1. Öffnen Sie im Ordner "Variablen" den Editor **Anwendungsvariablen**.
2. Vergewissern Sie sich, dass die Anwendungsvariablen korrekt auf Server bzw. Lokal eingestellt sind. Wenn die Tags dem Web Client zur Verfügung gestellt werden sollen, wählen Sie die Einstellung **Server**, andernfalls **Lokal**.

3. Schritt: Entwickeln Ihrer Bilder und Erstellen von HTML-Bildern

1. Entwickeln Sie Ihr WinStudio-Bild. Je nach Bildgröße Ihres Thin Clients empfiehlt sich gegebenenfalls die Entwicklung einer separaten Reihe von Bildern mit einer anderen Auflösung. Automatische Bildskalierung wird unterstützt (in einem vorhergehenden Schritt aktiviert), aber die Funktion hat natürliche Grenzen.
2. Speichern und schließen Sie alle Projektbilder.
3. Vergewissern Sie sich, dass Sie einen Startbildschirm (im Register **Viewer** des WinStudio-Eigenschaftendialogs) definiert haben.
4. Sie können durch Auswahl der Optionen **Speichern als HTML** oder **Bildgruppe als HTML speichern** im Anwendungsmenü einzelne Bilder als HTML speichern oder durch Auswählen der Option **Alle als HTML speichern** alle Bilder und Bildgruppen als HTML speichern.
5. Wenn Sie nur ein Bild aktualisieren und vor allem, wenn Sie Veränderungen an der Konfiguration der Web-Einstellungen vornehmen, sollten Sie die Funktion "Anwendung verifizieren" anwenden (im Kontextmenü des Ordners "WinStudio").

6. Die HTML-Seiten (Webseiten) werden im Unterordner "Web" Ihres Projektordners gespeichert.

4. Schritt: NTWebServer installieren

1. NTWebServer ist ein schlanker Webserver von "Bosch Rexroth" für die Windows-Umgebungen NT/XP/2000/Server 2003/Vista. **NTWebServer.exe** befindet sich in **[…]\WinStudio\ Bin**
2. Kopieren Sie **NTWebServer.exe** aus dem Ordner **\Bin** und fügen Sie ihn im Unterordner "Web" Ihres Projektordners ein.
3. Führen Sie aus dem Unterordner "Web" einen Doppelklick auf den NTWebServer aus, um diesen zu starten.
4. Es sollte sich ein neues Fenster öffnen. Am unteren Rand sollte eine Meldung erscheinen, dass der NTWebServer empfangsbereit ist. Wenn eine Meldung erscheint, dass der NTWebServer kein Socket öffnen konnte, wurde dies wahrscheinlich durch das Ausführen von Microsoft IIS (Web Server) im Hintergrund verursacht. Wenn dies der Fall ist, muss IIS angehalten und dann der NTWebServer neu gestartet werden.



Ein Webserver arbeitet oder „hört“ normalerweise auf dem TCP/IP-Port 80 eines Rechners. Nur jeweils ein laufender Prozess kann an einen gegebenen Port empfangen. Wenn also andere Prozesse auf Ihrem Rechner – z.B. SCADA-Fremdsoftware – bereits an Port 80 empfangen, dann können dieser Prozess und der Server-Prozess miteinander kollidieren. Sie müssen entweder einen der Prozesse zum Empfangen an einem anderen Port konfigurieren oder den entsprechenden Prozess mit dem Task Manager beenden. Wenn Sie den störenden Prozess nicht identifizieren können, öffnen Sie in Windows eine Eingabeaufforderung und geben den folgenden Befehl zum Auflisten aller Netzwerkprozesse ein:

netstat -a -o

5. Schritt: Ausführen des Laufzeitprojekts

1. Klicken Sie im Kontextmenü von WinStudio auf **Anwendung starten**.

6. Schritt: Microsoft Internet Explorer starten und mit dem Webserver verbinden

1. Klicken Sie zum Zugreifen auf das Programm Microsoft Internet Explorer auf die Schaltfläche **Start** (oder Alt-Tab).
2. Starten Sie den Internet Explorer und geben Sie die Startadresse ein (Start-Webseite). z.B. http://127.0.0.1/startup.html
3. Bitte beachten Sie, dass **startup.sg** (oder der Name Ihrer Startanzeige oder Bildgruppe) die Dateierweiterung HTML erhält, wenn über den Internetexplorer darauf zugegriffen wird.
4. Melden Sie sich ohne Passwort als **Guest** an, wenn keine Sicherheit aktiviert wurde.

Vorgehensweise B: Thin Client verwendet NTWebServer, Netzwerk-IP

Nach Ausführung von Vorgehensweise A werden im nächsten Schritt über das Netzwerk verbundene Thin Clients anstelle des Local Loopback aktiviert. Sie müssen die IP-Adresse Ihres Rechners aus Sicht des Netzwerks kennen. Diese Informationen erhalten Sie aus den Netzwerkeinstellungen in der Systemsteuerung oder durch Einfügen der Funktion **GetComputerIP()** in ein Rechteckobjekt in ein Bild und Ausführen des Projekts.

1. Schritt: Konfigurieren der "WinStudio"-Einstellungen

1. Halten Sie alle aktiven Laufzeitprojekte an.
2. Öffnen Sie den Web-Konfigurationsdialog und geben Sie im Feld IP-Adresse Datenserver die IP-Adresse Ihres Rechners ein (aus Sicht des Netzwerks). z.B. 192.168.1.100
3. Führen Sie das Werkzeug **Anwendung verifizieren** aus. Dies legt die IP-Adresse des Datenservers auf den Webseiten fest, so dass der Thin Client automatisch Daten mit dem richtigen Datenserver austauscht

2. Schritt: Ausführen des Laufzeitprojekts

1. Klicken Sie im Kontextmenü von WinStudio auf **Anwendung starten**.
2. Vergewissern Sie sich, dass der NTWebServer noch ausgeführt wird.

3. Schritt: Microsoft Internet Explorer starten und mit dem Webserver verbinden

1. Klicken Sie zum Zugreifen auf das Programm Microsoft Internet Explorer auf die Schaltfläche **Start** (oder Alt-Tab).
2. Starten Sie den Internet Explorer und geben Sie die Startadresse ein (Start-Webseite). z.B. in. http://192.168.1.100/startup.html
3. Melden Sie sich ohne Passwort als **Guest** an, wenn keine Sicherheit aktiviert wurde.

15.3 Zugrunde liegende Technologie

In einem "WinStudio"-Projekt werden verschiedene Bestandteile zur Implementierung der Thin-Client-Fähigkeit verwendet.

Dies sind:

Datenserver Der Datenserver ist in die "WinStudio"-Laufzeit integriert. Der Datenserver hat direkten Zugriff auf die Tag-Datenbank des "WinStudio"-Projekts (Laufzeit) und ist verantwortlich für das Arbeiten mit ISSymbol, um sicherzustellen, dass alle auf Webseiten auf Thin Clients angezeigten Tag-Daten mit den neuesten Werten aktualisiert werden.

"WinStudio" kann einen sekundären oder Backup-Datenserver unterstützen, der verwendet wird, falls der primäre Datenserver nicht verfügbar ist. Der Thin Client schaltet automatisch ohne Benutzereingriff zum sekundären Server um.

Webserver Der primäre Webserver ist verantwortlich für die bedarfsgerechte Bereitstellung von Webseiten (d.h. bei Anforderung durch den Client) durch Navigation zu verschiedenen Projektbildern durch den Thin Client. Der Webserver kommuniziert mithilfe des HTTP-Protokolls über TCP/IP mit dem Thin Client. Mit SSL (Secure Socket Layer) verschlüsselte Kommunikation kann aktiviert werden. Der Webserver muss sich nicht auf dem gleichen Rechner wie das "WinStudio"-Laufzeitprojekt befinden. In der Tat könnte der Webserver ein Unternehmens-Webserver ohne Windows sein. Allerdings benötigt der Webserver Zugriff auf die HTML-Dateien, die die Projekt-Webseiten sind.

"WinStudio" unterstützt einen sekundären Webserver, zu dem (durch den Client) automatisch umgeschaltet wird, falls der primäre Webserver nicht verfügbar ist.

Webbrowser Der Webbrowser befindet sich auf dem Thin-Client-Rechner und stellt die Funktion der grafischen Schnittstelle mit dem Benutzer zur Verfügung. Webseiten (HTML) werden bei Be-

darf an den Browser weitergegeben („pull“) und Daten werden vom Datenserver zum Browser geliefert („push“), wenn ein oder mehrere in dem auf dem Web-Client angezeigten Bild referenzierten Tags in der Tag-Datenbank aktualisiert werden.

ISSymbol

ISSymbol ist ein von "Bosch Rexroth" zur Verfügung gestelltes ActiveX-Steuerelement, das die Wechselwirkung zwischen dem Browser auf dem Web-Client und dem Webserver sowie dem Datenserver erleichtert. Es stehen Versionen von ISSymbol ActiveX Control für Windows XP/Vista/7 und alle Windows-Embedded-Plattformen zur Verfügung.

Das ISSymbol ActiveX-Steuerelement wird für Browser auf Basis des Internet Explorers und des Secure Viewers verwendet.

Web Tunneling Gateway

Der (primäre) Web Tunneling Gateway ist eine Brücke zwischen dem Server und dem Datenserver, die in einer von zwei Situationen verwendet wird. Bei der ersten Situation wird Datensicherheit benötigt (z.B. mit dem Thin Client ausgetauschte "WinStudio"-Daten müssen verschlüsselt werden). Bei der zweiten Situation ist der Datenserver hinter einer Unternehmens-Firewall „versteckt“ und nur die IP-Adresse (oder URL) des Webservers ist sichtbar.

"WinStudio" unterstützt einen (sekundären) Backup-Web-Tunneling-Gateway für den Fall, dass der primäre Web Tunneling Gateway nicht verfügbar ist. Der Thin Client schaltet automatisch zum sekundären Web Tunneling Gateway um.

Der Web Tunneling Gateway wird bei der Installation von "Rexroth WinStudio" automatisch auf Ihrem Rechner installiert, wenn das Installationsprogramm erkennt, dass IIS vorhanden ist.

- Der Web Tunneling Gateway wird automatisch installiert, wenn IIS während des Installationsvorgangs erkannt wird. Andernfalls muss er manuell installiert werden.
- Die Hauptfunktion des Web Tunneling Gateway ist die Kapselung von Datenpaketen in HTTP oder HTTPS zur Kommunikation durch eine Firewall.

15.4 ISSymbol Control Layer

ISSymbol ist eine von "Bosch Rexroth" entwickelte Komponente, die die mit "WinStudio" erstellten Bilder im Web-Browser anzeigen und Daten (Tag-Werte und historische Daten) mit dem TCP/IP-Server des Projekts austauschen kann. Auf der Thin-Client-Station ist der Webbrowser (z.B. Internet Explorer) der Container, der das ISSymbol-Steuerelement betreibt.

ISSymbol dient als Steuerungsebene zwischen dem Projekt und dem Webbrowser – genau wie die Java Virtual Machine für Java-basierte Anwendungen. Dieser Ansatz bietet ein hohes Maß an Sicherheit, da ISSymbol dem Projekt keinen direkten Zugriff auf das Betriebssystem erlaubt.

Wenn der Webbrowser die vom Benutzer angegebene HTML-Seite herunter lädt, sucht er die ISSymbol-Steuerungsregistrierung auf dem aktuellen Rechner. Wird diese nicht gefunden, versucht der Browser, die Registrierung von der in den Web-Einstellungen des Projekts angegebenen URL (in den erweiterten Eigenschaften des Dialogs **Web-Einstellungen** im Kontextmenü von

WinStudio) herunter zu laden. Der Webbrowser kann die Bilder aus dem Projekt nicht anzeigen, wenn die ISSymbol-Steuerung nicht ordnungsgemäß in der Thin-Client-Station registriert ist.

HINWEIS

Vergewissern Sie sich, dass Ihr Webbrowser für das Herunterladen von signierten ActiveX-Steuerelementen freigegeben ist, damit ISSymbol automatisch herunter geladen werden kann. Andernfalls muss ISSymbol manuell in der Thin-Client-Station registriert werden.

Bei Fragen zur Konfiguration dieser Einstellungen ziehen Sie bitte den Abschnitt Sicherheitseinstellungen in der Dokumentation zu Ihrem Webbrowser zu Rate.

Manuelles Installieren des ISSymbol-Steuerelements

Das ISSymbol-Steuerelement kann auch manuell in der Thin-Client-Station installiert werden. Die Vorgehensweise zum Installieren von ISSymbol ist nachfolgend für jedes Betriebssystem beschrieben:

Windows PC

1. Kopieren Sie die folgenden Dateien...
   - `[…]\WinStudio\ Bin\ISSymbolReg.exe`
   - `[…]\WinStudio\ Bin\ISSymbolVM.cab`

   ...in ein beliebiges Verzeichnis in der Thin-Client-Station. Stellen Sie sicher, dass beide Dateien im gleichen Verzeichnis gespeichert sind.
2. Führen Sie zum Registrieren des ISSymbol-Steuerelements im Rechner **ISSymbolReg.exe** aus.

Windows Embedded

1. Bestimmen Sie die Version des Betriebssystems und den Prozessortyp für das Windows-Embedded-Gerät und suchen Sie dann den entsprechenden Verteilerordner im Programmverzeichnis von "WinStudio":

   `[…]\WinStudio\ Redist\CE_version\prozessor_typ\Bin\`

   Zum Beispiel für einen MIPS IV Prozessor mit Windows CE 5.0 oder höher:

   `[…]\WinStudio\Redist\WinCE 5.0\MIPSIV\Bin\`
2. Kopieren Sie aus diesem Verzeichnis die folgenden Dateien...
   - `IndHTTP.dll`
   - `IndVkStd.dll`
   - `ISSymbolCE.ocx`

   ...und **eines** der folgenden Layouts für die virtuelle Tastatur...
   - `VKEN.ini` (für Englisch)
   - `VKGE.ini` (für Deutsch)

   ...in ein Verzeichnis in einen permanenten Speicher auf dem Windows-Embedded-Gerät. Stellen Sie sicher, dass alle Dateien im gleichen Verzeichnis gespeichert sind.
3. Führen Sie in der Windows-Eingabeaufforderung den folgenden Befehl aus:

   `regsvrce.exe "\ ISSymbol_directory\ISSymbolCE.ocx"`

Zum Beispiel:

```
regsvrce.exe "\Storage Card\ISSymbolCE.ocx"
```

4. Speichern Sie die Registry-Einstellungen, um **ISSymbolCE.exe** beim erneuten Hochfahren der Windows-Embedded-Geräts in der Registry zu behalten.

Windows Mobile / Pocket PC

1. Befolgen Sie Schritt 1 und 2 der Anleitung für Windows Embedded (siehe oben).
2. Führen Sie das Programm

   "RegSvrCE.exe" auf dem Windows-Mobile-Gerät aus. Während Sie sich in dem Programm befinden, führen Sie die folgenden Schritte aus:

   2.1 Suchen Sie die Datei "ISSymbolCE.ocx" in dem Verzeichnis, in das sie diese kopiert haben.

   2.2 Wählen Sie die Option **Register** aus.

   2.3 Klicken Sie auf **OK**.



Ihr Windows-Mobile-Gerät enthält evtl. nicht das Programm "RegSvrCE.exe", weil solche Verbrauchergeräte (z.B. Smartphones und PDAs) normalerweise nicht für diese Art der Verwendung bestimmt sind. Wenn Sie das Programm benötigen – oder "VBScript.dll", das ebenfalls auf den meisten Windows-Mobile-Geräten fehlt – wenden Sie sich bitte an den Kundendienst.

Funktionsweise

Nach dem Öffnen des Webbrowsers müssen Sie die URL für eine in der Serverstation verfügbare Webseite (z.B. **http://127.0.0.1/main.html**) in das Feld **Adresse** eingeben. An dieser Stelle führt der Thin Client den folgenden Vorgang aus:

1. Der Webbrowser lädt die HTML-Seite des angegebenen Bildes herunter.
2. Der Webbrowser sucht auf dem lokalen Rechner nach der Registrierung für das ISSymbol-Steuerelement. Wird diese nicht gefunden, versucht der Webbrowser, die ISSymbol-Komponente von der im Projekt konfigurierten URL herunter zuladen (in der HTML-Seite gespeicherte Einstellungen). Da das ISSymbol-Steuerelement ordnungsgemäß in der Thin-Client-Station registriert ist, wird es vom Webbrowser geladen. Von diesem Punkt an übernimmt ISSymbol die Kommunikation mit der Serverstation und der Webbrowser wird nur als Host für ISSymbol verwendet.
3. ISSymbol stellt eine Verbindung zum Datenserver her. Die IP-Adresse für den Datenserver wird in den Web-Einstellungen des Projekts konfiguriert (Eintrag **Web-Einstellungen** im Kontextmenü von WinStudio). Diese Einstellung ist in die HTML-Seite eingebettet.
4. ISSymbol öffnet ein Fenster auf dem Thin Client und fragt den **Benutzernamen** und das **Passwort** ab. Die eingegebenen Daten werden durch die Binärsteuerung auf Seite 81 verschlüsselt und an den Server gesendet. Die Serverstation prüft die Gültigkeit der Daten und ob die Rechte zum Öffnen des Startbildschirms vorliegen. Falls ja, wird der Vorgang fortgesetzt. Falls nicht, erhalten Sie eine Fehlermeldung, dass der Benutzername und das Passwort ungültig sind. In diesem Fall wird der Vorgang nicht fortgesetzt.

☞ Schritt 4 wird übersprungen, wenn das Sicherheitssystem bei der Konfiguration des Projekts deaktiviert ist.

5. ISSymbol lädt die zur Anzeige des vom Benutzer angegebenen Bildes erforderlichen Dateien herunter (Bilddateien, Tag-Datenbank, Übersetzungsdateien usw.).
6. ISSymbol verbindet sich mit dem Datenserver und liest den Wert des im angegebenen Bild angezeigten Tags aus.
7. ISSymbol zeigt das Bild im Webbrowser an und aktualisiert die Objekte kontinuierlich gemäß den vom Server gelesenen Werten. Jedes Mal, wenn sich der Wert eines im geöffneten Bild angezeigten Tags auf dem Server ändert, wird der neue Wert an den Thin Client gesendet (und umgekehrt). Daher findet zwischen dem Thin Client und der Serverstationen kein Polling statt. Diese Vorgehensweise verbessert die Kommunikationsleistung und optimiert den Datenverkehr im Netzwerk.

Bitte beachten Sie, dass an diesem Prozess zwei Server beteiligt sind:

- **Webserver** (HTTP-Server): Liefert die Dateien mithilfe des HTTP-Protokolls über TCP/IP vom Server an den Thin Client.
- **Datenserver** (TCP/IP-Servermodul aus "WinStudio"): Liefert Tag-Werte und/oder historischen Daten vom auf dem Server ausgeführten Projekt zu den Thin-Client-Stationen.

Obwohl beide Server normalerweise auf dem gleichen Rechner laufen, stellt "WinStudio" die Flexibilität zur Verfügung, bei Bedarf jeden Server auf einer anderen Station auszuführen. Weitere Informationen unter Typische Architekturen für web-basierte Anwendungen auf Seite 458.

15.5 Beispiele für Client/Server-Architektur

Dieser Abschnitt beschreibt einige auf web-basierte Lösungen angewendete Beispielarchitekturen und gibt Informationen zur Konfiguration des Projekts für jede Architektur. Dieser Abschnitt beschreibt nicht alle möglichen Architekturen, stellt aber auf der Grundlage der nachfolgend beschriebenen grundlegenden Architekturen die für Entwicklung und Konfiguration verschiedener Szenarien notwendigen Konzepte zur Verfügung.

Die Webeinstellungen werden im zugehörigen Konfigurationsdialog editiert. Zum Öffnen dieses Dialogs klicken Sie im Kontextemenü von WinStudio auf **Werkzeuge ▸ Web-Einstellungen**. Durch Drücken der Schaltfläche "Erweitert" erhalten Sie Zugriff auf zusätzliche Einstellungen. Die folgenden Abbildungen zeigen diese Dialoge:

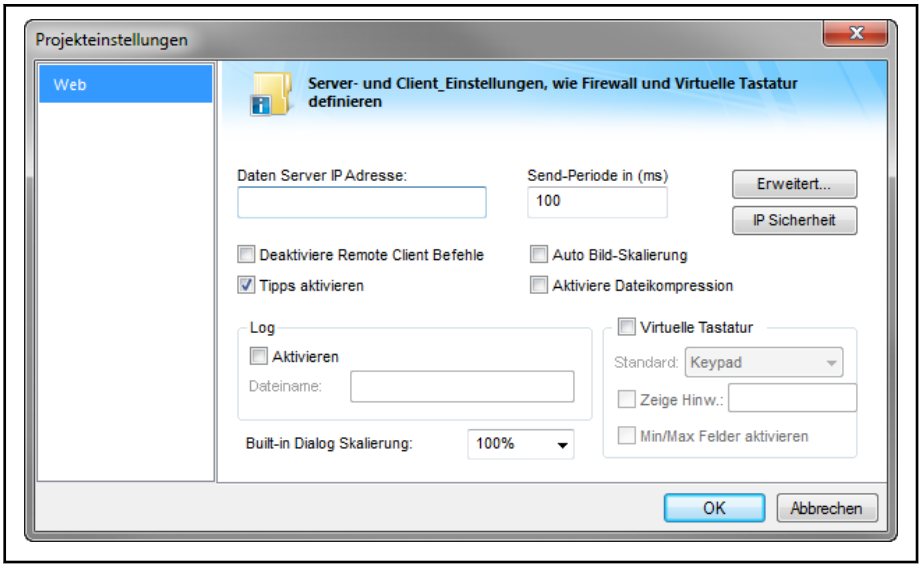


Abb.15-4: Haupteinstellungen Web (1)

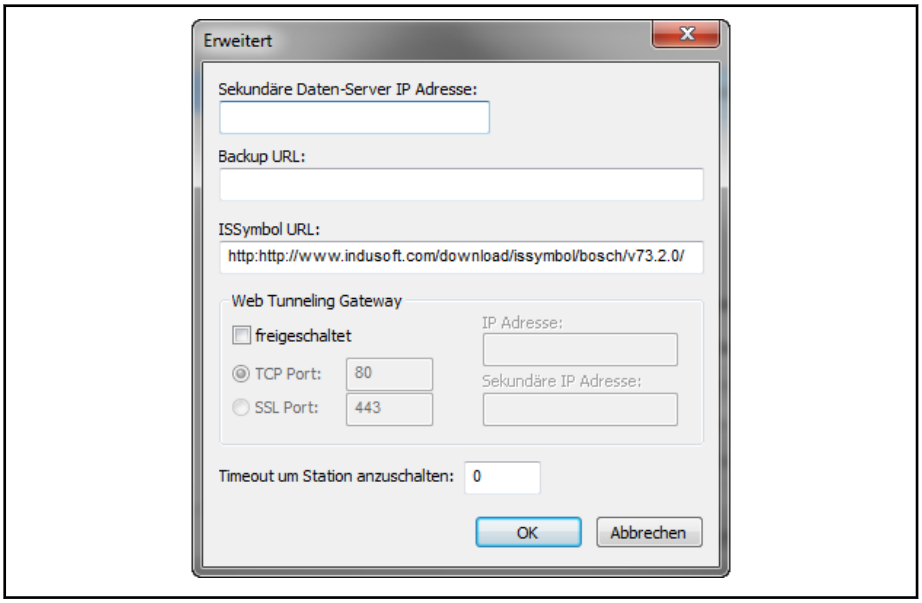


Abb.15-5: Haupteinstellungen Web (2)

Die folgende Tabelle erläutert die in den vorangehenden Dialogen gezeigten Haupteinstellungen Web:

Einstellung	Beschreibung
IP-Adresse Datenserver	Wenn der Web Tunneling Gateway **deaktiviert** ist: Das Thin-Client-Steuerelement (ISSymbol) verwendet die IP-Adresse des Datenservers für die Verbindung mit dem TCP/IP-Servertask von "WinStudio". Wenn der Web Tunneling Gateway **aktiviert** ist: Der Web Tunneling Gateway verwendet die IP-Adresse Datenservers für die Verbindung mit dem TCP/IP-Servertask von "WinStudio".
Sekundäre Datenserver IP-Adresse	Entspricht der IP-Adresse des Datenservers. Allerdings wird die sekundäre IP-Adresse nur verwendet, wenn die Verbindung mit der IP-Adresse des Datenservers nicht zustande kommt.

Bereitstellung als Web-Anwendung

Einstellung	Beschreibung
IP-Adresse Web Tunneling Gateway	Das Thin-Client-Steuerelement (ISSymbol) verwendet die IP-Adresse des Web Tunneling Gateways für die Verbindung mit dem Web Tunneling Gateway.
Sekundäre IP-Adresse Web Tunneling Gateway	Entspricht der IP-Adresse des Web Tunneling Gateways. Allerdings wird die sekundäre Web Tunneling Gateway IP-Adresse nur verwendet, wenn die Verbindung mit der Web Tunneling Gateway IP-Adresse nicht zustande kommt.

Die sekundären Adressen können in den folgenden Szenarien verwendet werden:

- wenn der Thin Client sich mit einem von zwei redundanten Servern (Web oder Daten) verwenden kann oder
- wenn die Thin Clients sich über das Intranet (LAN – Local Area Network) oder über das Internet (WAN – Wide Area Network) mit dem Server verbinden können. In diesem Fall sollten die primären Adressen aufgrund des häufiger von den Thin Clients verwendeten Netzwerks konfiguriert werden. In den folgenden Beispielen werden die LAN-Adressen als primäre und die WAN-Adressen als sekundäre Adressen verwendet.

Die folgenden Tabellen erläutern die Bedeutung einiger in den folgenden Beispielen verwendeten Begriffe:

Begriff	Beschreibung
LAN	Local Area Network (z.B. Intranet)
WAN	Local Area Network (z.B. Internet)
Server	Station, auf der die folgenden Komponenten ausgeführt werden: Obwohl "WinStudio" nicht auf der gleichen Station wie die anderen Komponenten ausgeführt werden muss, gehen die folgenden Beispiele davon aus. • "WinStudio" (TCP/IP Server-Task) • Webserver (z.B. Internet Information Services von Microsoft – IIS) • Web Tunneling Gateway für IIS (falls aktiviert)
Thin Client LAN	Thin-Client-Station (Webbrowser + ISSymbol-Steuerelement), die den Server über LAN verbindet
Thin Client WAN	Thin-Client-Station (Webbrowser + ISSymbol-Steuerelement), die den Server über WAN verbindet
IP_SERVER_LAN	IP-Adresse des Servers im LAN
IP_SERVER_WAN	IP-Adresse des Servers im WAN
IP_ROUTER_LAN	IP-Adresse des Routers im LAN
IP_ ROUTER_WAN	IP-Adresse des Routers im WAN
BildName	Name des als HTML gespeicherten Projektbildes, das auf der Thin-Client-Station geöffnet ist

Beispiel 1: Web Server und Thin Client im selben Intranet (LAN)

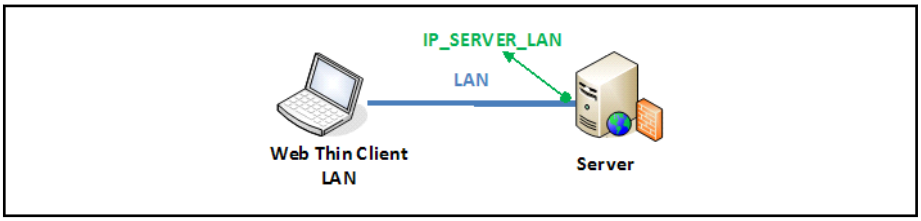


Abb.15-6: Web Server und Thin Client im selben Intranet (LAN)

Dies ist die am häufigsten verwendete Architektur und auch die am leichtesten zu konfigurierende. Bei dieser Architektur laufen sowohl der Webserver

(z.B. Microsoft IIS) als auch der Datenserver (z.B. das "WinStudio" TCP/IP-Servermodul) auf dem gleichen Rechner. Der Thin Client verbindet sich mit dem Webserver, um die HTML-Bilddatei(en) herunter zuladen. Dann verbindet er sich mit dem Datenserver, um Daten mit dem "WinStudio"-Laufzeitprojekt auszutauschen. Da sowohl der Thin Client als auch die Serverstation mit dem gleichen Netzwerk verbunden sind, kann der Thin Client über die IP-Adresse (oder den Host-Namen) direkt auf die Serverstation zugreifen.

Konfiguration:

Einstellung	WTG aktiviert	Web Gateway deaktiviert
Datenserver IP-Adresse	IP_SERVER_LAN	IP_SERVER_LAN
Sekundäre Datenserver IP-Adresse	-	-
IP-Adresse Web Tunneling Gateway	IP_SERVER_LAN	-
Sekundäre IP-Adresse Web Tunneling Gateway	-	-

- URL von Thin Client LAN: **http://IP_SERVER_LAN/BildName.html**
- Stammverzeichnis des Webservers (HTTP-Server) auf der Serverstation: Unterordner "Web" des Projekts

Beispiel 2: Web Server mit Intranet-(LAN) und Internet-(WAN) Verbindungen

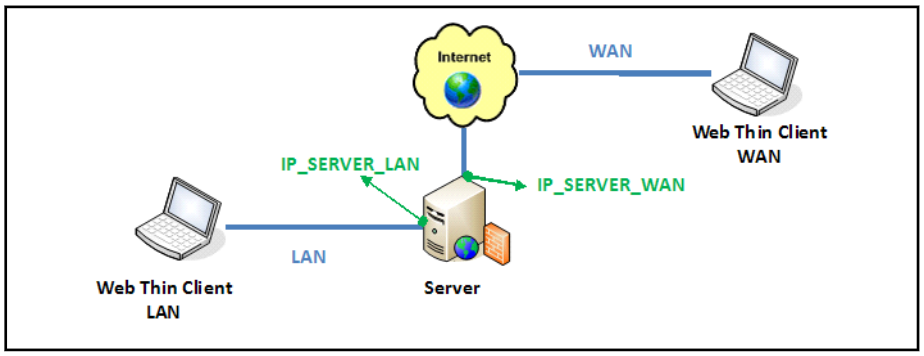


Abb.15-7: Web Server mit Intranet-(LAN) und Internet-(WAN) Verbindungen

Bei dieser Architektur laufen sowohl der Webserver (z.B. Microsoft IIS) als auch der Datenserver (z.B. das "WinStudio" TCP/IP-Servermodul) auf dem gleichen Rechner. Thin Clients können sich entweder über eine Intranetverbindung (LAN) oder über eine Internetverbindung (WAN) mit dem Server verbinden (z.B. zwei verschiedene Ethernet-Ports).

Konfiguration:

Einstellung	Web Gateway aktiviert	Web Gateway deaktiviert
IP-Adresse Datenserver	IP_SERVER_LAN	IP_SERVER_LAN
Sekundäre Datenserver IP-Adresse	IP_SERVER_LAN	IP_SERVER_WAN
IP-Adresse Web Tunneling Gateway	IP_SERVER_LAN	-
Sekundäre IP-Adresse Web Tunneling Gateway	IP_SERVER_WAN	-

Bereitstellung als Web-Anwendung

- URL von Thin Client LAN: **http://IP_SERVER_LAN/BildName.html**
- URL von Thin Client WAN: **http://IP_SERVER_WAN/BildName.html**
- Stammverzeichnis des Webservers (HTTP-Server) auf der Serverstation: Unterordner "Web" Ihres Projektordners
- Sie müssen dem Webserver im Internet (WAN) eine feste IP-Adresse zuweisen und das Projekt muss auf diesem Server ausgeführt werden. Weitere Informationen zum Erhalt einer festen IP-Adresse für Ihren Server erhalten Sie von Ihrem Internetanbieter oder von Ihrer IT-Abteilung.

Beispiel 3: Web Server mit Intranet-(LAN) und Router-Internet-(WAN) Verbindungen

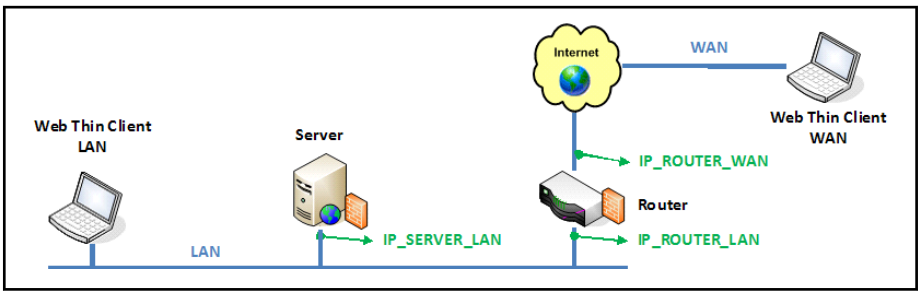


Abb.15-8: Web Server mit Intranet-(LAN) und Router-Internet-(WAN) Verbindungen

Bei dieser Architektur laufen sowohl der Webserver (z.B. Microsoft IIS) als auch der Datenserver (z.B. das "WinStudio" TCP/IP-Servermodul) auf dem gleichen Rechner. Thin Clients können sich entweder über eine Intranetverbindung (LAN) oder über eine Internetverbindung (WAN) mit dem Server verbinden. Zwischen Intranet (LAN) und Internet (WAN) befindet sich ein Router.

Konfiguration:

Einstellung	Web Gateway aktiviert	Web Gateway deaktiviert
Datenserver IP-Adresse	IP_SERVER_LAN	IP_SERVER_LAN
Sekundäre Datenserver IP-Adresse	IP_SERVER_LAN	IP_ROUTER_WAN
IP-Adresse Web Tunneling Gateway	IP_SERVER_LAN	-
Sekundäre IP-Adresse Web Tunneling Gateway	IP_ROUTER_WAN	-

- URL von Thin Client LAN: **http://IP_SERVER_LAN/BildName.html**
- URL von Thin Client WAN: **http://IP_ROUTER_WAN/BildName.html**
- Der Router muss für die Weiterleitung des/der TCP-Ports von dessen öffentlicher IP (IP_ROUTER_WAN) zu der privaten IP des Servers (IP_SERVER_LAN) konfiguriert werden.

  Wenn der Web Gateway **aktiviert** ist, dürfen nur der HTTP-Port (80, Standard) oder der HTTPS-Port (SSL-Port 443, Standard) vom IP_ROUTER_WAN zum IP_SERVER_LAN weitergeleitet werden.

  Wenn der Web Gateway **deaktiviert** ist, müssen sowohl der HTTP-Port (80, Standard) als auch der Studio-TCP/IP-Server-Port (1234, Standard) vom IP_ROUTER_WAN zum IP_SERVER_LAN weitergeleitet werden. Weitere Informationen zur Konfiguration der Port-Weiterleitung finden Sie in der Dokumentation Ihres Routers.
- Stammverzeichnis des Webservers (HTTP-Server) auf der Serverstation: Unterordner "Web" Ihres Projektordners
- Sie müssen dem Router im Internet (WAN) eine feste IP-Adresse zuweisen und das Projekt muss auf diesem Server ausgeführt werden. Weitere Informationen zum Erhalt einer festen IP-Adresse für Ihren Server erhalten Sie von Ihrem Internetanbieter oder von Ihrer IT-Abteilung.

15.6 Konfigurieren des Datenservers

"WinStudio" verfügt über zwei Dialoge, die zur Konfiguration des Datenservers und des Webservers verwendet werden. Diese Information ist in den HTML-Webseiten der entsprechenden Bilder eingebettet. Der Datenserver ist Teil des "WinStudio"-Laufzeitprojekts und verwendet das TCP/IP-Servermodul. Diese Konfiguration besteht aus drei (3) grundlegenden Schritten:

1. Schritt: Konfiguration Dialog Webeinstellungen

1. Wählen Sie im Kontextmenü von WinStudio den Eintrag **Werkzeuge ▸ Web-Einstellungen**.

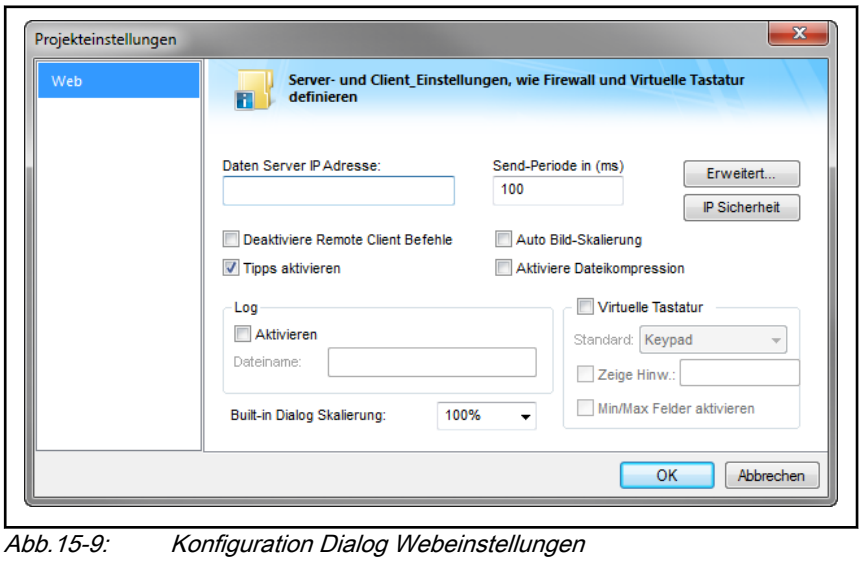


Abb. 15-9: Konfiguration Dialog Webeinstellungen

2. Gehen Sie im Feld **IP-Adresse Datenserver** die IP-Adresse des Rechners ein, auf der das Laufzeitprojekt abgelegt werden soll.
3. Vergewissern Sie sich, dass die Kontrollkästchen **Automatische Bildskalierung, Dateikomprimierung aktivieren** und **ToolTips aktivieren** ausgewählt sind. Wenn Sie die Ausgabe von Befehlen vom Web-Client an den Server deaktivieren wollen, aktivieren Sie das Kontrollkästchen **Remote-Client-Befehle deaktivieren**.
4. Wenn der Web-Client eine virtuelle Tastatur verwenden soll, aktivieren Sie das Kontrollkästchen **Virtuelle Tastatur** und alle zusätzlichen Einstellungen für die virtuelle Tastatur im Thin Client.
5. Durch Klicken auf die Schaltfläche **Erweitert** können eine Backup-URL (z.B. Backup-Website für Webseiten) und eine sekundäre IP-Adresse für den Datenserver definiert werden. Dies gilt für einen redundanten Webserver bzw. einen redundanten Datenserver. Außerdem kann in diesem Dialog Web Tunneling aktiviert werden. Die URL für die ISSymbol OCX kann auch definiert werden, wenn diese nicht auf dem Webserver gefunden wurde.

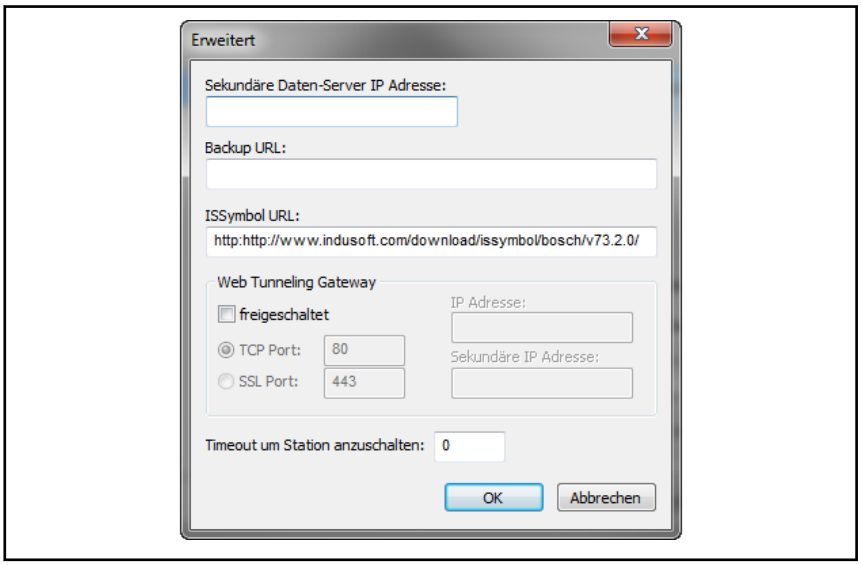


Abb.15-10: Konfiguration Dialog Webeinstellungen, Weitere Einstellungen

6. Durch Klicken auf die Schaltfläche **IP-Sicherheit** können Sie einen Bereich von IP-Adressen definieren, die vom Datenserver akzeptiert werden sollen. Wenn ein Datenanfrage von einer IP-Adresse außerhalb der festgelegten Bereiche kommt, wird die Anfrage nicht bestätigt. Dies ist eine Funktion der Embedded Firewall.

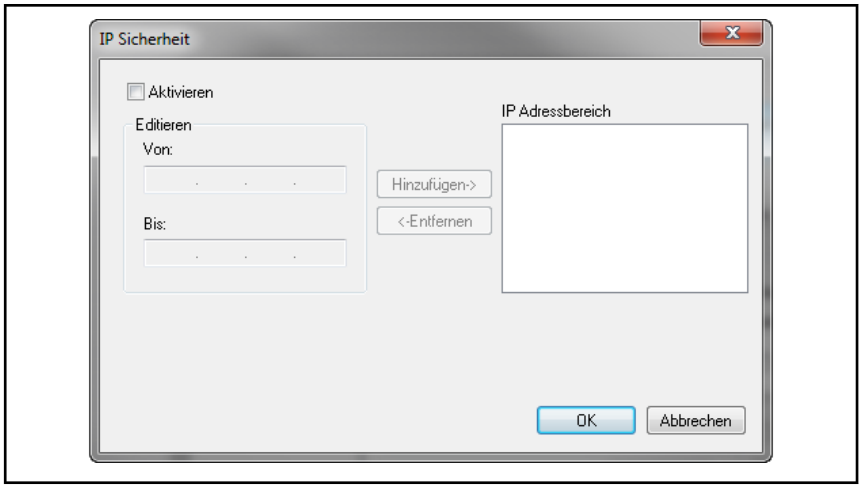


Abb. 15-11: IP-Sicherheit

2. Schritt: Konfiguration Dialog Kommunikationseinstellungen

1. Öffnen Sie den WinStudio-Eigenschaftendialog und klicken Sie auf das Register **Kommunikation**.

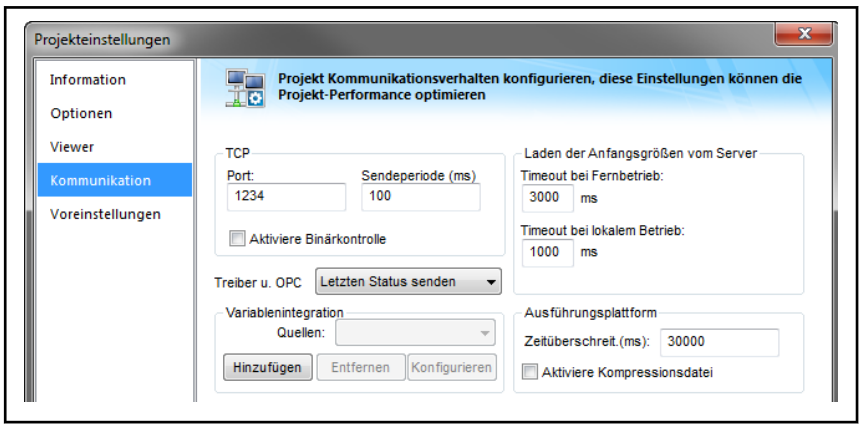


Abb. 15-12: Konfiguration Dialog Kommunikationseinstellungen

2. Geben Sie für den Datenserver die Port-Nummer ein (1234 ist der Standard). Es kann außerdem die Datensendezeit definiert werden (d.h. die Zeitspanne für aktualisierte Kommunikation von Datenwerten zum Web-Client).
3. Optional kann die **Binärsteuerung** der Daten aktiviert werden. Dies ist sicherer, aber langsamer. Die Standardeinstellung ist deaktiviert.

3. Schritt: Aktivieren Sie den TCP/IP-Task

1. Öffnen Sie die Task-Konfiguration, indem Sie den Eintrag **Werkzeuge ▸ Status** im Kontextmenü von WinStudio wählen.

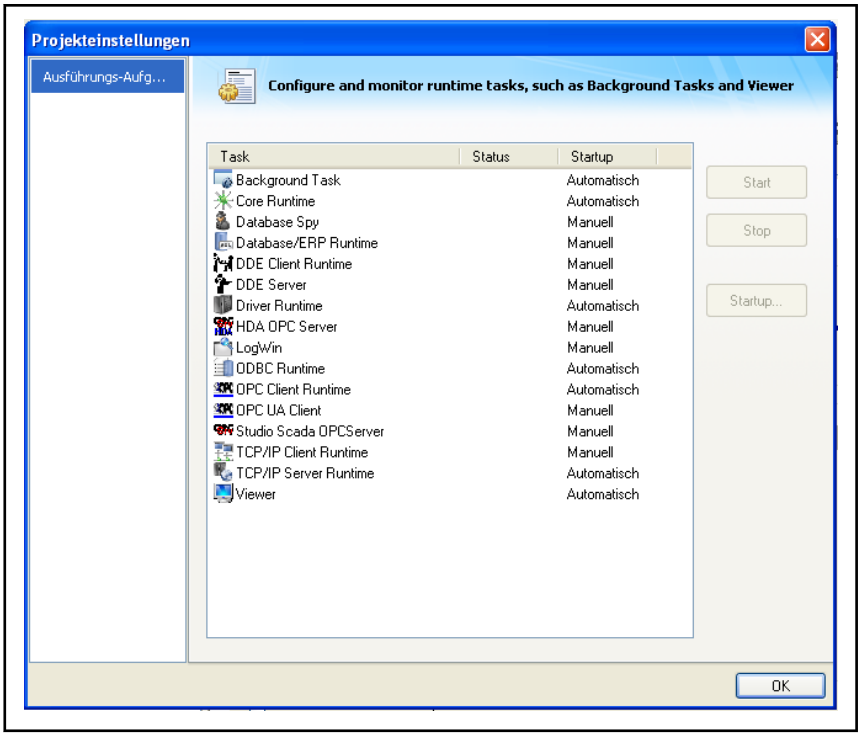


Abb.15-13: Aktivieren Sie den TCP/IP-Task

2. Vergewissern Sie sich, dass der TCP/IP-Server auf Automatisch eingestellt ist. Dies sollte der Standardzustand sein, kann aber manuell durch Auswahl der Schaltfläche **Starten** konfiguriert werden.
3. Vergewissern Sie sich, dass die Nummer des TCP-Ports korrekt eingestellt ist (siehe Kommunikationseinstellungen oben), andernfalls startet der TCP/IP-Server und stoppt gleich wieder.

- Diese Konfigurationsdaten (aus Schritt 1 + 2) werden beim **Speichern als HTML** in die von der "WinStudio"-Entwicklungsumgebung erzeugten Webseiten eingebettet.
- Wenn diese Einstellungen verändert werden, **MUSS** die Funktion **Anwendung verifizieren** ausgeführt werden.

Verwendung der Funktion SetWebConfig

Die eingebaute Funktion **SetWebConfig()** erlaubt dem Entwickler das Konfigurieren von Datenserver und Web-Client per Programm und die daraus resultierenden Konfigurationseinstellungen werden automatisch in den HTML-Dateien Ihres Projekts aktualisiert (gespeichert im Unterordner "Web"). Diese Funktion wird immer auf dem "WinStudio"-Server ausgeführt und funktioniert in Windows- und Windows-Embedded-Umgebungen.

Weitere Informationen unter Funktion SetWebConfig auf Seite 709.

Konfigurieren des Webservers zur Unterstützung von IE und Secure Viewer

"WinStudio" kann gleichzeitig Browser sowohl auf Grundlage des Internet Explorers (IE) als auch des Secure Viewers unterstützen. Dafür muss allerdings das System sehr sorgfältig konfiguriert werden.

Auf der Seite des Webservers

- Alle HTML-Dateien kommen in den Projektunterordner "Web".
- Konfigurieren Sie Ihren Webserver (z.B. IIS) so, dass sein Stammverzeichnis auf das Projektverzeichnis zeigt.
- Wenn keine Internetverbindung zur Website von "Bosch Rexroth" verfügbar ist und das ISSymbol ActiveX-Steuerelement nicht auf dem IE-

basierten Web Client installiert ist, legen Sie die Datei ISSymbolVM.cab im Projektordner ab.

Auf der Seite des Web-Clients

- Der Browser auf Grundlage des Secure Viewers sollte so konfiguriert werden, dass er auf die .APP Datei im Projektordner zeigt.
- Die Ausgangs-URL des Browsers auf IE-Grundlage sollte wie folgt aussehen:

  http://192.168.1.100/Web/startup.html

  ...wobei startup.html Ihre Startwebseite ist.

15.7 Konfigurieren eines Webservers als Host Ihrer Projektseiten

Als Teil der Bereitstellung Ihres "WinStudio"-Projekts über das Web müssen Sie einen Webserver als Host für Ihre Projektbilder konfigurieren.

Es ist nicht zwingend erforderlich, einen Windows-PC oder ein Windows-Embedded-Gerät als Host für Ihre Projektseiten zu verwenden. Diese Seiten sind im Prinzip statische Dateien, die auf das Herunterladen warten; alle Laufzeitrechenarbeiten werden vom Projekt-Viewer (d.h. Internet Explorer mit installiertem ISSymbol auf Seite 455 oder Secure Viewer) auf dem Thin Client übernommen. Als solches können alle Standard-Webserver auf einer beliebigen Computerplattform als Host für Ihre Seiten verwendet werden.

Wenn Sie zum Beispiel bereits über einen Unix-basierten Intranetserver verfügen, können Sie den Unterordner "Web" Ihres Projekts (bzw. den Ordner mit Ihren gespeicherten Projektseiten) auf den Server kopieren und Ihren Thin Client auf diese Serveradresse zeigen lassen.

Bitte beachten Sie aber, dass der von Ihnen gewählte Webserver evtl. nicht robust genug ist, um in einer Produktionsumgebung für Ihr Projekt als Server zu dienen und er evtl. nicht alle Funktionen von "Rexroth WinStudio" unterstützt. Wenn Sie diese Funktionen benötigen, sollten Sie in den meisten Fällen Microsoft IIS wie unten beschrieben verwenden. Im Detail:

- Zur Unterstützung von <Mobile Access> (SMA), muss der Server Collaborative Data Objects (CDO) und Active Server Pages (ASP) verarbeiten können; und
- Zur Unterstützung von Web Tunneling Gateway auf Seite 470 (WTG), muss der Server Internet Service API (ISAPI) Erweiterungen erlauben.

Vor dem Installieren und Konfigurieren von Software studieren Sie bitte sorgfältig die entsprechende Dokumentation.



Ein Webserver arbeitet oder „hört“ normalerweise auf dem TCP/IP-Port 80 eines Rechners. Nur jeweils ein laufender Prozess kann an einen gegebenen Port empfangen. Wenn also andere Prozesse auf Ihrem Rechner – z.B. SCADA-Fremdsoftware – bereits an Port 80 empfangen, dann können dieser Prozess und der Server-Prozess miteinander kollidieren. Sie müssen entweder einen der Prozesse zum Empfangen an einem anderen Port konfigurieren oder den entsprechenden Prozess mit dem Task Manager beenden. Wenn Sie den störenden Prozess nicht identifizieren können, öffnen Sie in Windows eine Eingabeaufforderung und geben den folgenden Befehl zum Auflisten aller Netzwerkprozesse ein:

netstat -a -o

Bereitstellung als Web-Anwendung

NTWebServer und CEWebServer

NTWebServer und CEWebServer sind schlanke, konfigurationsfreie Webserver, die kostenlos in "Rexroth WinStudio" enthalten sind. Sie können diese zur Demonstration Ihres Projekts und zum Ausführen von einfachen Tests verwenden, ohne finanziell und technisch in einen vollständigen Webserver investieren zu müssen. Allerdings sind sie nicht robust und unterstützen weder SMA noch WTG. Für reale Anwendungen empfehlen wir die unten beschriebene Anwendung von Microsoft IIS.

NTWebServer kann auf jedem unterstützten auf Seite 39 Windows-PC ausgeführt werden. Er befindet sich im Programmverzeichnis von "WinStudio" unter

[…]\WinStudio\Bin\NTWebServer.exe.

Kopieren Sie diese Datei vor dem Ausführen in den Unterordner "Web" Ihres Projekts; der Server muss sich im gleichen Ordner wie die später bedienten Seiten befinden. Nach dem Ausführen wird dieser Ordner die oberste Ebene (das „Start“-Verzeichnis) der Website.

CEWebServer kann auf jedem unterstützten auf Seite 39 Windows-Embedded-Gerät ausgeführt werden. Er befindet sich im Programmverzeichnis von "WinStudio" unter

[…]\WinStudio\Redist\CE_version\CE_processor\CEWebServer.exe.

Wenn der Unterordner "Web" Ihre Projekts bereits in den permanenten Speicher des Geräts kopiert wurde, kopieren Sie nun **CEWebServer.exe** in den gleichen Ordner und führen diese Datei aus. Auch hier wird dieser Ordner zur obersten Ebene der Website.

☞ Sowohl NTWebServer als auch CEWebServer müssen als normale Programme ausgeführt werden, sie können nicht als Windows-Dienste ausgeführt werden.

Microsoft IIS

Internet Information Services (IIS) ist die vollausgestattete Server-Software, die mit Windows Server und „professionellen“ Windows-Versionen zusammen ausgeliefert wird:

IIS-Versionen

Windows-Version	IIS-Version	Hinweise
Windows XP Professional	IIS 5,1	maximal 10 gleichzeitige Verbindungen
Windows Server 2003 Windows XP Professional x64 Edition	IIS 6,0	Keine Begrenzungen für Verbindungen.
Windows Server 2008 Windows Vista (die meisten Versionen)	IIS 7,0	Keine Begrenzungen für Verbindungen.
Windows Server 2008 R2 Windows 7 (alle Versionen)	IIS 7.5	Keine Begrenzungen für Verbindungen.
Windows CE und Windows Mobile 5.0 und höher	IIS für CE .NET	Muss im Platform Builder enthalten sein. Standard ist maximal 10 gleichzeitige Verbindungen

IIS unterstützt alle Funktionen von "Rexroth WinStudio" und ist robust genug, um fast jedes "WinStudio"-Projekt in einer Produktionsumgebung zu bedienen. Dieser Webserver wird für die meisten Benutzer empfohlen. Zum ordnungsgemäßen Installieren und Konfigurieren sollten Sie allerdings Erfahrung mit der Verwaltung von Windows in einem Netzwerk haben.

Aus Gründen der Systemsicherheit ist IIS bei der Installation des Betriebssystems standardmäßig deaktiviert. Um es zu aktivieren, verwenden Sie den

Dialog "Windows-Funktionen" (in Windows 7, **Systemsteuerung** ▸ **Programme** ▸ **Programme und Funktionen** ▸ **Windows-Funktionen ein-/ausschalten**):

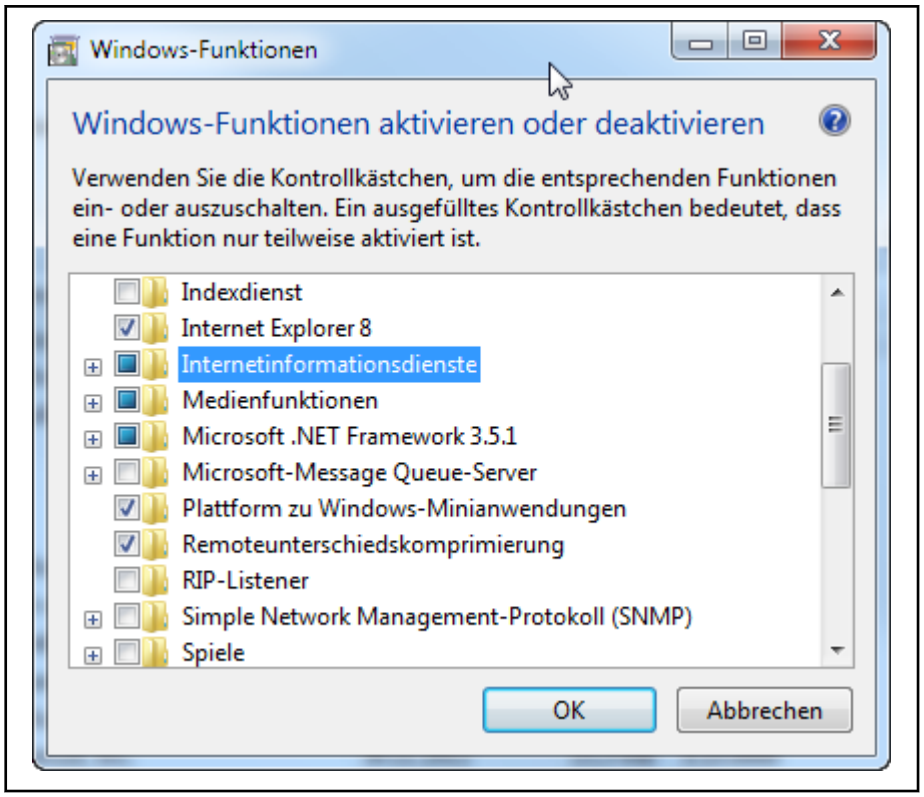


Abb.15-14: Aktivieren von IIS im Dialog "Windows-Funktionen"

Wenn Sie SMA und/oder WTG in Ihrem "WinStudio"-Projekt verwenden wollen, müssen Sie sicherstellen, dass die ASP- und ISAPI-Erweiterungen ebenfalls aktiviert sind:

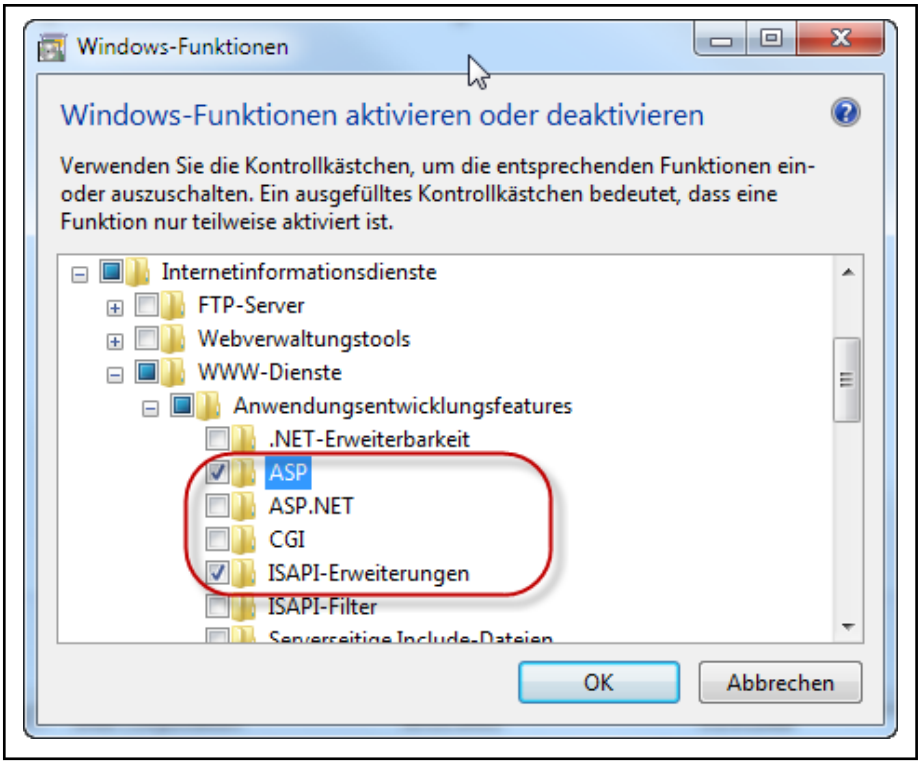


Abb. 15-15: Aktivieren von ASP- und ISAPI-Erweiterungen im Dialog "Windows-Funktionen"

Wenn IIS aktiviert ist, können Sie die Verwaltungswerkzeuge (in Windows 7, **Systemsteuerung ▸ System und Sicherheit ▸ Verwaltungswerkzeuge**) zum Konfigurieren verwenden. Weitere Informationen finden Sie in der ausführlichen Dokumentation von Microsoft.

Apache für Windows

Wenn Ihnen IIS nicht zur Verfügung steht oder Sie ihn nicht verwenden möchten, dann ist der zweitbeliebteste Webserver für Windows der Open-Source-Server Apache. Allerdings erfordert dieser für Installation und Konfiguration noch mehr Fachwissen als IIS, Sie sollten also die Dokumentation vor der Verwendung sorgfältig studieren.

15.8 Installieren des Web Tunneling Gateway

Der "Web Tunneling Gateway" ("WTG", auch bekannt als "IndWebSvc") ermöglicht es einem mit dem Internet verbundenen Rechner – üblicherweise Ihr Webserver – Projektdaten zwischen Remote-Thin-Clients und einem "nicht" mit dem Internet verbundenen Datenserver zu leiten. Dieser Abschnitt beschreibt die Installation von WTG.

Bevor Sie mit dieser Aufgabe beginnen, sollten Sie zunächst Microsoft IIS als Webserver und Host für Ihre Projektseiten einrichten; WTG läuft nur auf Microsoft IIS. Außerdem muss Ihr Webserver mit dem Internet verbunden sein und über eine statische IP-Adresse verfügen und nicht über eine dynamisch zugewiesene IP-Adresse. Wenden Sie sich zum Erhalt einer statischen IP-Adresse an Ihre IT-Abteilung oder Ihren Internetanbieter. Schließlich müssen sowohl Ihr Web- als auch Ihr Datenserver auf dem selben internen TCP/IP-Netzwerk laufen, auch wenn nur Ihr Webserver mit dem Internet verbunden ist.

Damit Thin Clients auf Ihr Projekt zugreifen können, müssen sie mit dem Webserver, der als Host für Ihre Projektseiten fungiert, und mit dem Datenserver, der Ihr "WinStudio"-Projekt tatsächlich ausführt, kommunizieren können. Das ist kein Problem, wenn Ihre Server und Clients sich alle auf dem

selben internen Netzwerk befinden oder wenn Sie beide Server mit dem Internet verbinden, so dass Remote-Thin-Clients mit ihnen kommunizieren können.

Wenn Sie allerdings Ihren Datenserver nicht mit dem Internet verbinden wollen (aus Netzwerksicherheits- oder Topologiegründen), müssen Sie die Projektdaten durch einen anderen, verbundenen Rechner leiten.

WTG stellt diese Weiterleitungsfähigkeiten zur Verfügung. Es ist eine ISAPI-Erweiterung für Microsoft IIS und wird üblicherweise auf dem selben Webserver installiert, der als Host für Ihre Projektseiten fungiert. Es kapselt die zwischen Datenserver und Thin Clients gesendeten Daten und garantiert so die Projektsicherheit. Außerdem kann es Daten gleichzeitig zwischen mehreren Datenservern leiten.

Diese Anleitung gilt für:

- Microsoft IIS 7.0 ausgeführt auf Windows Vista und Windows Server 2008; und
- Microsoft IIS 7.5 ausgeführt auf Windows 7 und Windows Server 2008 R2.

1. Stellen Sie auf Ihrem Webserver (bzw. dem an das Internet angeschlossene Rechner, der als Gateway fungiert) sicher, dass Microsoft IIS läuft und ISAPI-Erweiterungen aktiviert sind. Weitere Informationen unter Konfigurieren eines Webservers als Host für Ihre Projektseiten auf Seite 467.
2. Kopieren Sie das WTG-Installationsprogramm aus dem Anwendungsverzeichnis von "WinStudio" in das Stammverzeichnis Ihrer Website (z.B. "C:\inetpub\wwwroot\").

   Das Installationsprogramm befindet sich unter

   ..\WinStudio\Bin\WebTunnelingGateway.exe
3. Führen Sie das Installationsprogramm und die Schritte des Installationsassistenten aus. Es können keine Optionen ausgewählt werden. Das Installationsprogramm erstellt ein neues Verzeichnis im Stammverzeichnis der Website und installiert dort die WTG-Erweiterungsdateien. Es erstellt außerdem einen neuen Anwendungs-Pool.
4. Öffnen Sie den IIS-Manager. Das Fenster "Internet Information Services (IIS) Verwaltung" wird angezeigt.
   - 4.1 Klicken Sie auf **Start** und dann auf **Systemsteuerung**.
   - 4.2 Im Fenster für die Systemsteuerung wählen Sie eine der folgenden Vorgehensweisen.
     - Für Windows Server 2008 klicken Sie auf **Verwaltungswerkzeuge**.
     - Für Windows Vista klicken Sie auf **System und Wartung** und dann auf **Verwaltungswerkzeuge**.
     - Für Windows 7 oder Windows Server 2008 R2 klicken Sie auf **System und Sicherheit** und dann auf **Verwaltungswerkzeuge**.
   - 4.3 Im Fenster für die Verwaltungswerkzeuge wählen Sie eine der folgenden Vorgehensweisen.
     - Für Windows Vista oder Windows 7 führen Sie einen Doppelklick auf **Internet Information Services (IIS) Verwaltung** aus.
     - Für Windows Server 2008 oder Windows Server 2008 R2 führen sie einen Doppelklick auf **Serververwaltung** aus und

klicken dann im Fenster der Serververwaltung auf **Internet Information Services (IIS) Verwaltung**.

Das Fenster Internet Information Services (IIS) Verwaltung wird angezeigt.

5. Stellen Sie sicher, dass der Anwendungs-Pool vom Installationsprogramm erstellt wurde. Wenn "IndWebSvcPool" nicht auf der Liste steht, wurde das WTG-Installationsprogramm nicht ordnungsgemäß beendet. Kehren Sie zum Anfang dieser Vorgehensweise zurück.
   5.1 Erweitern Sie im Bereich **Verbindungen** den Serverknoten und klicken Sie auf Anwendungs-Pools.
   5.2 Suchen Sie auf der Seite Anwendungs-Pools in der Liste der Pools nach IndWebSvcPool.

   Wenn IndWebSvcPool sich nicht auf der Liste befindet, wurde das WTG-Installationsprogramm nicht ordnungsgemäß beendet. Kehren Sie zum Anfang dieser Vorgehensweise zurück.
6. Hinzufügen einer ISAPI-Beschränkung für IndWebSvc.
   6.1 Klicken Sie im Bereich **Verbindungen** auf den Serverknoten.
   6.2 Führen Sie auf der **Startseite** des Serverknotens einen Doppelklick auf **ISAPI- und CGI-Beschränkungen** aus.
   6.3 Klicken Sie auf der Seite **ISAPI- und CGI-Beschränkungen** im Bereich **Aktionen** auf **Hinzufügen**.

   Der Dialog ISAPI- und CGI-Beschränkungen wird angezeigt.
   6.4 Geben Sie im Feld **ISAPI- und CGI-Pfad** die Pfad zur Datei **IndWebSvc.dll** ein.

   (z.B., **C:\inetpub\wwwroot \indwebsvc\IndWebSvc.dll**), oder klicken Sie auf die Suchschaltfläche (…) und navigieren Sie zur Datei.
   6.5 Geben Sie im Feld **Beschreibung** IndWebSvc ein.
   6.6 Wählen Sie **Ausführen des Erweiterungspfads erlauben**.
   6.7 Klicken Sie auf **OK**.
7. Konfigurieren Sie die Handler-Mappings der Website, um ISAPI-Erweiterungsdateien ausführbar zu machen.
   7.1 Erweitern Sie **Sites** im Bereich **Verbindungen** und klicken Sie auf die Website, die als Gateway fungiert (d.h. normalerweise **Standard Website** bzw. die Site, die als Host für Ihre WinStudio-Projektseiten fungiert).
   7.2 Führen Sie auf der **Startseite** der Website einen Doppelklick auf **Handler Mappings** aus.
   7.3 Wählen Sie auf der Seite **Handler Mappings** aus der Liste der Mappings den Eintrag **ISAPI-dll**.

   Dieses Element ist standardmäßig deaktiviert.
   7.4 Klicken Sie im Bereich Aktionen auf **Funktionsberechtigungen bearbeiten**.

   Der Dialog Funktionsberechtigungen bearbeiten wird angezeigt.
   7.5 Wählen Sie **Ausführen**.
   7.6 Klicken Sie auf **OK**.
8. Stellen Sie sicher, dass die Website und die Gateway-Erweiterung laufen – starten Sie den Internet Explorer auf dem Server und navigieren Sie dann zu http://127.0.0.1/indwebsvc/indwebsvc.dll.

`127.0.0.1` ist die „localhost-“ oder „loopback-“Adresse, der Browser betrachtet also die Website, die auf dem selben Rechner läuft.

Die resultierende Webseite sollte wie folgt aussehen:

```
Studio Web Gateway
Status: Ok
Version: 1.3
Aktive Verbindungen: 0
```

Wenn dies nicht erscheint, läuft entweder die Website nicht oder die Gateway-Erweiterung wurde nicht korrekt konfiguriert. Kehren Sie zum Anfang dieser Vorgehensweise zurück.

9. Schließen Sie die IIS-Verwaltung.
10. Konfigurieren Sie Ihr "WinStudio"-Projekt so, dass es die Adresse des Gateways enthält.

    Weitere Informationen unter Projekteinstellungen: Web auf Seite 86.

15.9 Konfigurieren des Thin Client

"WinStudio" unterstützt zwei Arten von Webbrowsern für die Verwendung mit Thin Clients: Microsoft Internet Explorer, der ActiveX-Steuerelemente unterstützt, und den von "Bosch Rexroth" entwickelten Secure Viewer. Beide Browser können gleichzeitig mit dem Webserver verwendet werden und die Wahl des Browsers hängt von den Projektanforderungen ab. Der Secure Viewer ist für die Verwendung in der Produktionsumgebung gedacht, während der Internet Explorer sich generell mehr für Desktop-Anwendungen oder für Situationen, in denen sich der Thin Client mit verschiedenen Projektservern verbinden muss, eignet.

Konfigurieren des Microsoft Internet Explorer

Die Konfiguration von Microsoft Internet Explorer für die Verwendung als Webbrowser für einen Thin Client beinhaltet nur wenige Schritte. Die wichtigsten Elemente sind:

- Stellen Sie sicher, dass ActiveX-Steuerelemente aktiviert sind. Die Einstellungen hierfür befinden sich im allgemeinen unter **Extras** ▸ **Internetoptionen** ▸ **Sicherheit**.

  Klicken Sie auf die Schaltfläche **Stufe anpassen** und stellen Sie sicher, dass die ISSymbol ActiveX-Steuerung installiert werden kann. Dann kehren Sie zu einer höheren Sicherheitsstufe zurück. (Andernfalls installieren Sie bitte die ISSymbol ActiveX-Steuerung manuell.)

- Vergewissern Sie sich, dass VBScript auf Ihrem Rechner unterstützt wird. Dies ist bei Desktop-PCs normalerweise kein Problem, kann aber bei Windows-Embedded-Geräten ein Problem sein. Sie müssen sicher sein, dass VBRun.dll auf Ihrem Rechner installiert und registriert ist.

Wenn Sie Microsoft Internet Explorer als Webbrowser für Thin-Client-Projekte verwenden wollen, sollten Sie ggf. einige Navigationsleisten und Schaltflächen entfernen, da diese sehr viel Platz einnehmen.

Sie können auf dem Desktop eine Verknüpfung erstellen für den schnellen Zugriff auf Ihre Startwebseite auf dem Webserver. Damit gelangen Sie durch Klicken auf das Symbol zur Webseite. Zusätzlich können Sie diese Verknüpfung in den Startordner legen, damit die Webseite automatisch beim Starten des Browsers gestartet wird.

Installieren von Secure Viewer

Installieren auf einem Windows PC

Sie müssen den Secure Viewer von Ihrer CD installieren (oder von der Website von "Bosch Rexroth" herunterladen). Führen Sie die folgenden Schritte aus:

1. Fordern Sie das Installationsprogramm für Secure Viewer von Ihrem Softwarehersteller an, speichern Sie es auf dem entsprechenden Rechner und führen Sie es aus.
2. Befolgen Sie die Anweisungen des Installationsassistenten. Bei der Installation müssen nur zwei Einstellungen konfiguriert werden:
   - **URL**: Geben Sie die URL oder den Dateipfad der Projektdatei (**projekt_name.app**) auf dem Webserver ein.
   - **Server IP**: Geben Sie die IP-Adresse oder den Host-Namen des Datenservers (TCP/IP-Server) ein.
3. Beenden Sie die Installation und klicken Sie auf **Fertigstellen**, um das Installationsprogramm zu schließen.

Der Installationsassistent installiert und registriert automatisch die Thin-Client-ActiveX-Komponente (ISSymbol.ocx). Wenn die Einstellungen **URL** und **Server-IP** korrekt konfiguriert wurden, sollte der Secure Viewer startbereit sein.

☞ Zur Erhöhung der Sicherheit der Laufzeitumgebung fügen Sie dem Startverzeichnis in Windows eine Verknüpfung zum Secure Viewer (Viewer.exe) hinzu. Viewer.exe installiert sich im Unterordner \Bin, in dem sich das "WinStudio"-Programm installiert.

Installieren auf Windows Embedded

Der Secure Viewer kann durch Kopieren der notwendigen Dateien in dem permanenten Speicher des Geräts auch auf einem Windows-Embedded-Gerät installiert werden. Zur Installation des Secure Viewers:

1. Bestimmen Sie die Version des Betriebssystems und den Prozessortyp für das Windows-Embedded-Gerät und suchen Sie dann den entsprechenden Unterordner im Programmverzeichnis von "WinStudio": Zum Beispiel: Für Windows CE 5.0 auf einem MIPS-Prozessor, suchen Sie den Unterordner **[…]\WinStudio\ Redist\WinCE 5.0\MIPS\Bin**.
2. *Wählen Sie im Unterordner die folgenden Dateien aus:*
   - IndVkStd.dll
   - ISSymbolCE.ocx
   - Tagi.bin
   - Viewer.exe
   - ViewerCfg.exe
3. Kopieren Sie die Dateien in den permanenten Speicher auf dem Windows-Embedded-Gerät.
4. Registrieren Sie die ISSymbol-Steuerung auf dem Windows-Embedded-Gerät, indem Sie den folgenden Befehl aus dem Fenster **Eingabeaufforderung** ausführen:

   regsvrce.exe "\ISSymbolPath \ISSymbolCE.ocx"

   Beispiel:

   regsvrce.exe "\Storage Card\ISSymbolCE.ocx"
5. Speichern Sie die Registry-Einstellungen, um **ISSymbolCE.exe** beim erneuten Hochfahren der Windows-Embedded-Geräts in der Registry zu behalten.

Informationen zum Speichern der Registry-Einstellungen finden Sie in der Herstellerdokumentation des Geräts.

6. Starten Sie den Secure Viewer durch Ausführen von **Viewer.exe**.

Ändern der Konfiguration des Secure Viewer

Nach der erstmaligen Installation wird die Konfiguration des Secure Viewer in der Datei **viewer.ini** gespeichert, die sich im selben Verzeichnis wie **Viewer.exe** befinden sollte. (Dies sollte im Unterordner **\Bin** sein, wo das Programm Secure Viewer installiert ist.) Nach der Installation kann die Konfiguration des Secure Viewer auf zwei Arten geändert werden. Erstens kann das ebenfalls installierte Hilfsprogramm für die Viewer-Konfiguration (**ViewerCfg.exe**) ausgeführt werden.

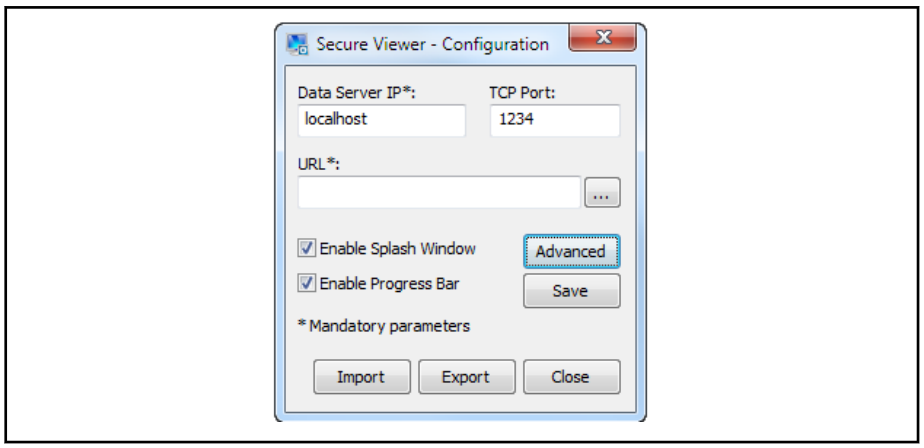


Abb.15-16: Dialog ViewerCfg

Auf Windows Vista und Windows 7 sind für das ordnungsgemäße Ausführen des Hilfsprogramms für die Viewer-Konfiguration Administratorrechte erforderlich. Es sollte standardmäßig mit diesen Berechtigungen installiert werden. Wenn weiterhin Probleme auftreten, überprüfen Sie die Dateieigenschaften für "ViewerCfg.exe" und stellen Sie sicher, dass **Ausführen als Administrator** ausgewählt ist.

Das Hilfsprogramm für die Konfiguration stellt die folgenden Optionen zur Verfügung:

- Schaltfläche **Speichern**: Sofortiges Speichern der aktuellen Einstellungen in eine "viewer.ini" im selben Verzeichnis wie das Hilfsprogramm "ViewerCfg.exe".
- Schaltfläche **Importieren**: Laden einer Datei "viewer.ini" aus einem anderen Verzeichnis.
- Schaltfläche **Exportieren**: Speichern der Datei "viewer.ini" in einem anderen Verzeichnis.
- Feld **Datenserver-IP**: Geben Sie die IP-Adresse (bzw. den Host-Namen) Ihrer Datenserverstation ein. Die Datenserverstation ist der Rechner bzw. das Gerät, auf dem das TCP/IP-Server-Modul ausgeführt wird.
- Feld **TCP-Port**: Geben Sie die Port-Nummer des Datenservers ein, wenn diese sich vom Standard-Port 1234 unterscheidet.
- Feld **URL**: Geben Sie die URL oder den Dateipfad der Projektdatei ("*.app") auf dem Webserver ein.

- Option **Splash Screen aktivieren**: Durch Anklicken (Aktivieren) dieser Option wird beim Starten des Secure Viewer ein Splash Screen (Startbildschirm) angezeigt.
- Option **Fortschrittsbalken aktivieren**: Durch Anklicken (Aktivieren) dieser Option wird ein Fortschrittsbalken angezeigt, während der Secure Viewer die Projektdatei lädt.
- Schaltfläche **Weitere Einstellungen**: Durch Anklicken erhalten Sie Zugriff auf weitere Konfigurationsoptionen:

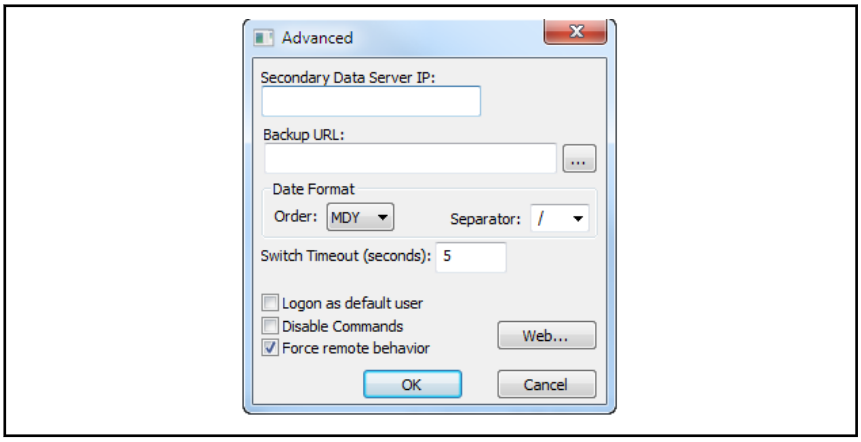


Abb.15-17: Hilfsprogramm zur Konfiguration von Secure Viewer – Weitere Einstellungen

  - Feld **Sekundäre Datenserver-IP**: Tragen Sie hier die IP-Adresse (bzw. den Host-Namen) Ihrer sekundären Datenserverstation ein. Wenn der primäre Datenserver ausfällt, versucht der Secure Viewer automatisch, sich mit dem sekundären Datenserver zu verbinden.
  - Feld **Backup-URL**: Geben Sie die URL des Backup-Speicherorts der Projektdatei ein.
  - **Datumsformat**: Anpassen des Datumsformats auf der Secure-Viewer-Station an den Datenserver. Hiermit wird die Kompatibilität mit Steuerungsobjekten für Alarme/Ereignisse oder Trends in Ihrem Projekt gewahrt.
  - Feld **Zeitüberschreitung für Umschalten zwischen Stationen**: Geben Sie die Anzahl der Sekunden ein, die der Secure Viewer bei Ausfall des primären Servers bis zum Verbindungsversuch mit dem sekundären Server warten soll. (Weitere Informationen unter Registerkarte Web Seite 86.)
  - Option **Als Gast anmelden**: Durch Anklicken (Aktivieren) wird der Secure Viewer automatisch als Gast angemeldet und es sind weder Benutzername noch Passwort erforderlich.
  - Option **Befehle deaktivieren**: Auf Durch Anklicken (Aktivieren) wird verhindert, dass der Secure Viewer Befehle zurück an den Server sendet. Es werden nur die aktuellen Informationen von Server angezeigt.
  - Schaltfläche **Web**: Wenn Sie zum Verbinden Ihres Intranets mit dem Internet einen Web Tunneling Gateway konfiguriert haben, geben Sie hier die Adressen für den Gateway ein.

Die zweite Art zum Ändern der Konfiguration des Secure Viewer ist das manuelle Bearbeiten der Datei **viewer.ini** mit einem Texteditor. Die Dateistruktur ist wie folgt:

Dateistruktur

```
[Options]
nosplash=            // Enable (0) or disable (1) the splash window
noprogressbar=       // Enable (0) or disable (1) the progress bar
ds1=                 // Data Server Primary
ds2=                 // Data Server Secondary
dsp=                 // Data Server Port
wtg1=                // Web Tunneling Gateway Primary
wtg2=                // Web Tunneling Gateway Secondary
url=                 // URL from project file (*.app)
proxyip=             // Proxy Address
proxyp=              // Proxy Port
ceemul=              // Enable (1) or disable (0) CEView emulation
```

Beispiel

PC mit "WinStudio"-Laufzeitprojekt (hat sowohl Webserver (NTWebServer oder IIS) als auch Datenserver):

- Die IP-Adresse lautet 192.168.1.106
- Der Projektname ist SecureViewerTest
- Die Projektdatei ist SecureViewerTest.app

Dialog Konfiguration Secure Viewer:

- Datenserver-IP = 192.168.1.106
- TCP-Port = 1234
- URL = http://192.168.1.106//SecureViewerTest.app

Alternativ kann die Datei viewer.ini bearbeitet werden:

viewer.ini

```
[Options]
url=http://192.168.1.106//SecureViewerTest.app
noprogressbar=1
ds1=192.168.1.106
nosplash=1
ds2=
dsp=1234
wtg1=
wtg2=
user=Guest

[OEM]
Splash=Splash.bmp //this is a splash screen
```



- Ihre "Rexroth WinStudio"-Laufzeitlizenz muss einen Secure Viewer (Thin Client) unterstützen, andernfalls wird die Verbindung mit dem Datenserver verweigert.
- Stellen Sie sicher, dass NTWebServer in Ihrem Projektordner (nicht im Unterordner "Web") abgelegt wird oder dass IIS auf Ihrem Projektordner zeigt.

15.10 Sicherheit für webbasierte Anwendungen implementieren

Es gibt verschiedene Methoden zur Implementierung der Sicherheit von webbasierten Anwendungen. Welcher Ansatz sich für Sie eignet, kann von einer Reihe von Faktoren abhängen und eine oder mehrere Methoden der Sicherheitsimplementierung beinhalten.

Methode 1: Passwortschutz

"WinStudio" stellt die Fähigkeit zum Erstellen von Benutzergruppen und einzelnen Benutzern innerhalb einer Gruppe zur Verfügung. Jede Gruppe (z.B. Bediener, Aufsichtsführende, Wartung) kann verschiedene Sicherheitsebenen haben und auf verschiedene Funktionsebenen zugreifen. Für jeden Benutzer können individuelle Passwörter konfiguriert werden.

Bereitstellung als Web-Anwendung

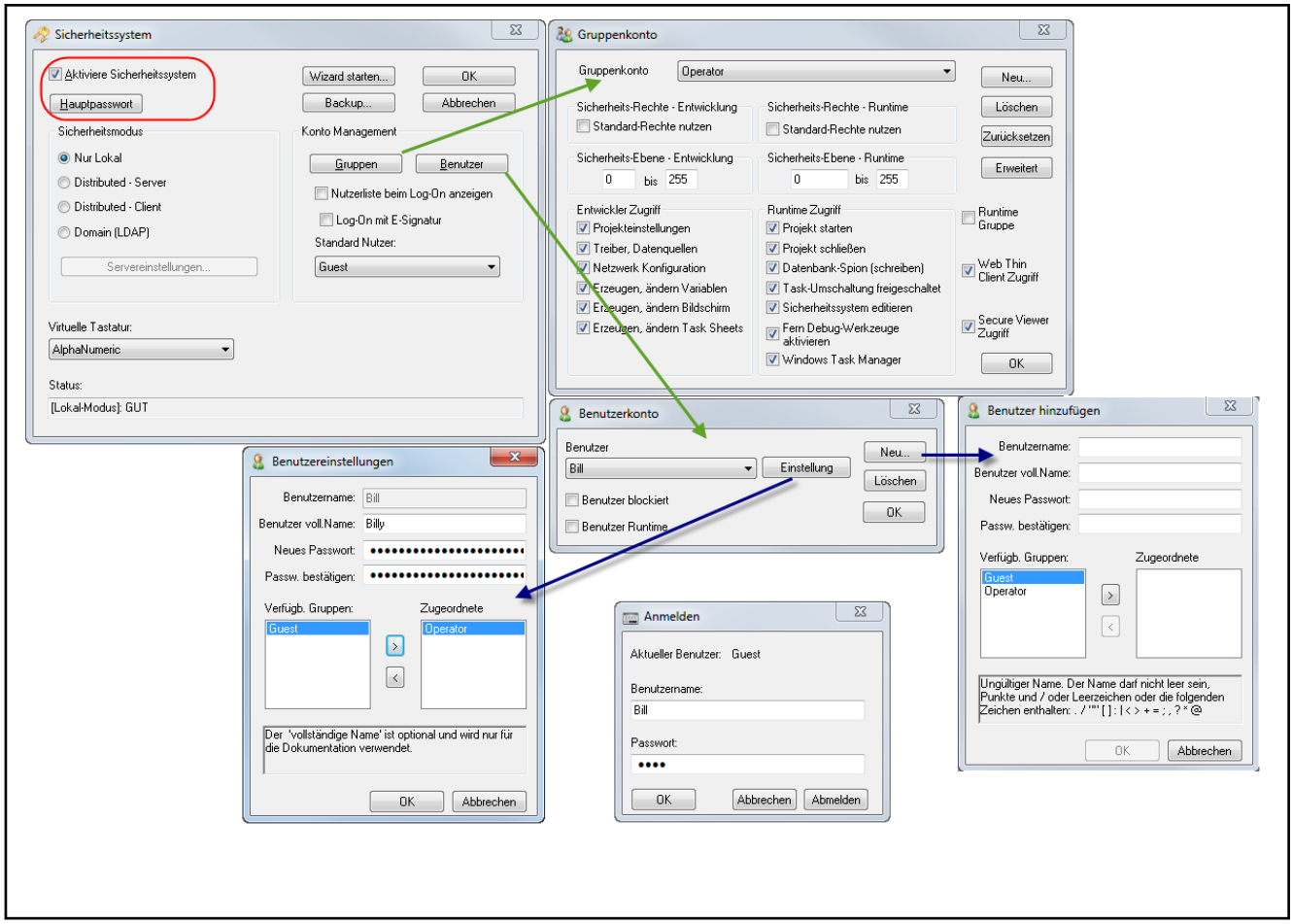


Abb. 15-18: Sicherheitsgruppen und -benutzer

Zusätzlich können Gruppen mit erweiterten Einstellungen ausgestattet werden, die Funktionen wie Mindestpasswortgröße, Passwortalterung, E-Signatur auf Objekten mit Befehlsanimationen, automatische Kontosperre (z.B. Sperren nach einer bestimmten Anzahl ungültiger Zugriffsversuchen) sowie das Blockieren von Benutzerkonten (vorübergehend deaktivieren – z.B. bei Urlaub eines Mitarbeiters) zulassen.

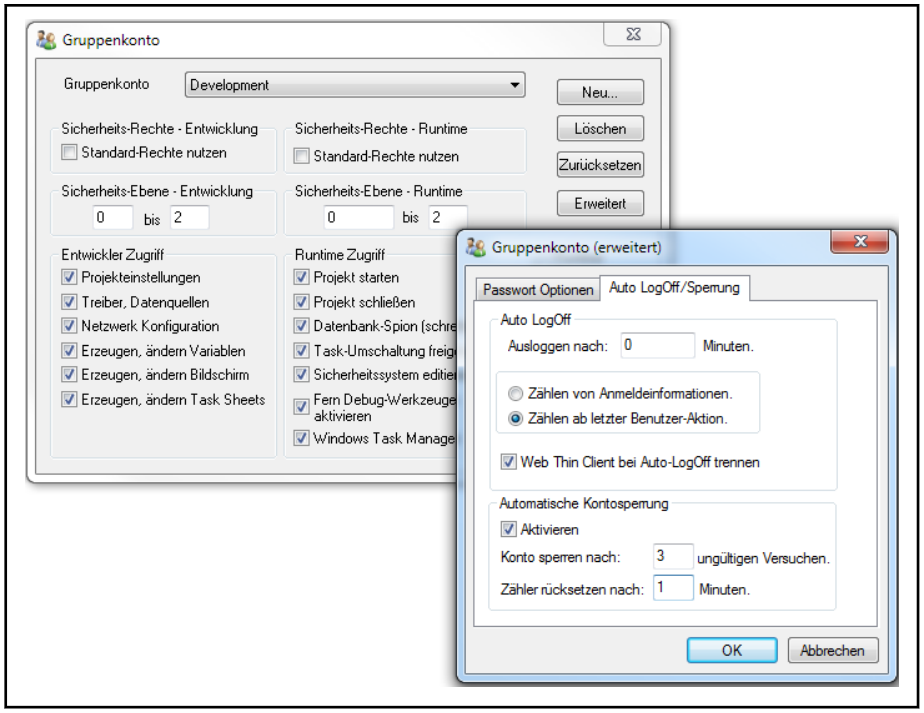


Abb.15-19: Automatische Kontosperre und Blockieren von Benutzerkonten

Wenn die Systemsicherheit aktiviert ist, stehen diese Funktionen für den Passwortschutz auch auf der Thin-Client-Station zur Verfügung. Wenn ein Benutzer auf einer Thin-Client-Station versucht, eine Verbindung mit dem Webserver aufzubauen, werden Benutzername und Passwort abgefragt. Wenn diese ungültig sind, erhält der Benutzer keinen Zugriff auf das System.

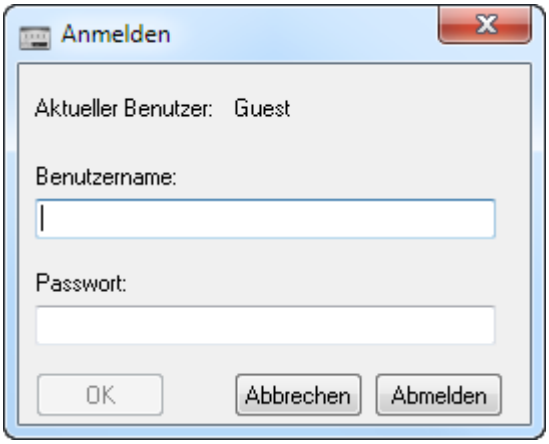


Abb.15-20: Dialog Anmelden

Innerhalb eines Projekts können den verschiedenen Bildobjekten und ihren Animationen sowie dem Bildzugriff verschiedene Sicherheitsebenen zugewiesen werden. Der aktuell angemeldete Benutzer muss über einen Zugriffsebenenbereich verfügen, der dem gewünschten Objekt bzw. dem gewünschten Bild entspricht. Nachfolgend wird eine repräsentative Methode zum Zuweisen von Sicherheitszugriffsebenen pro Gruppe dargestellt.

Bereitstellung als Web-Anwendung

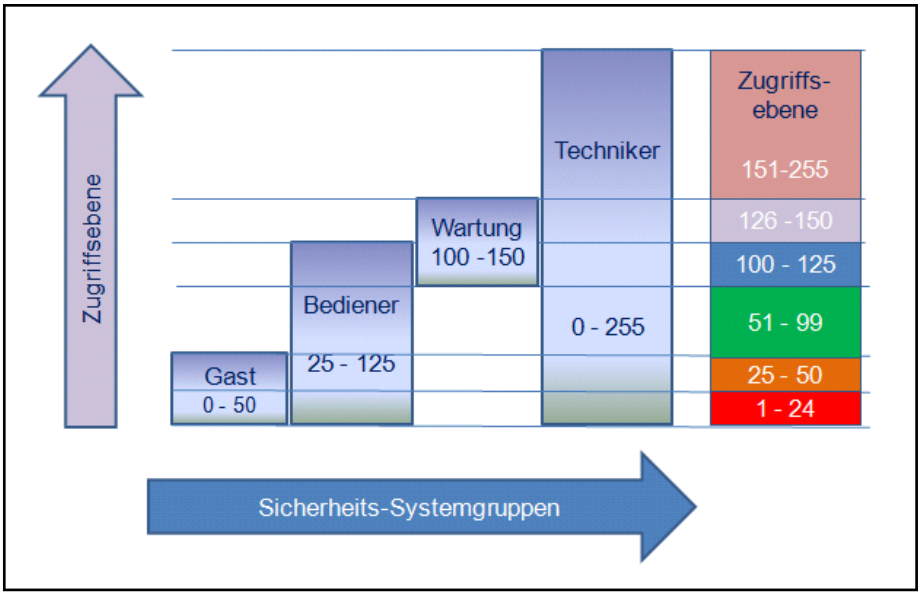


Abb. 15-21: Zugriff auf Bilder

Weitere Informationen unter Sicherheit auf Seite 407.

Methode 2: Deaktivieren von Thin-Client-Befehlen

"WinStudio" lässt den bidirektionalen Datenaustausch zwischen dem Thin Client und dem Datenserver zu. Aus Sicherheitsgründen kann es aber von Vorteil sein, nur dem Thin Client das Betrachten der Prozess- oder Maschinendaten zu erlauben und keine Daten zurück an den Datenserver zu senden.

Durch Auswahl (Aktivierung) der Option **Remote-Client-Befehle deaktivieren** in den Web-Einstellungen (im Kontextmenü von WinStudio) wird sichergestellt, dass alle Befehle von einer Thin-Client-Station blockiert werden. Die Kommunikation wird einseitig (vom Server zu den Thin Clients):

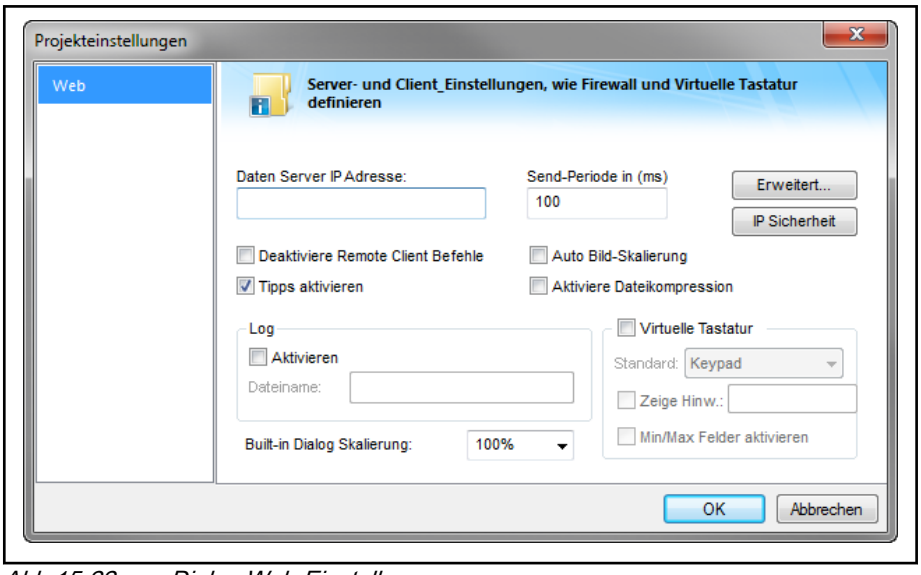


Abb. 15-22: Dialog Web-Einstellungen

Methode 3: Embedded Firewall

Mit dieser Funktion kann der Benutzer den Zugriff auf das Projekt aufgrund der IP-Adresse der Thin Clients filtern. Wenn ein Thin Client versucht, eine Verbindung zur Serverstation aufzubauen, überprüft der Server, ob die IP-Adresse der Thin-Client-Station für den Zugriff auf das Projekt befugt ist. Die Bereiche der autorisierten IP-Adressen können durch Klicken auf **IP-Sicherheit** in den Web-Einstellungen.

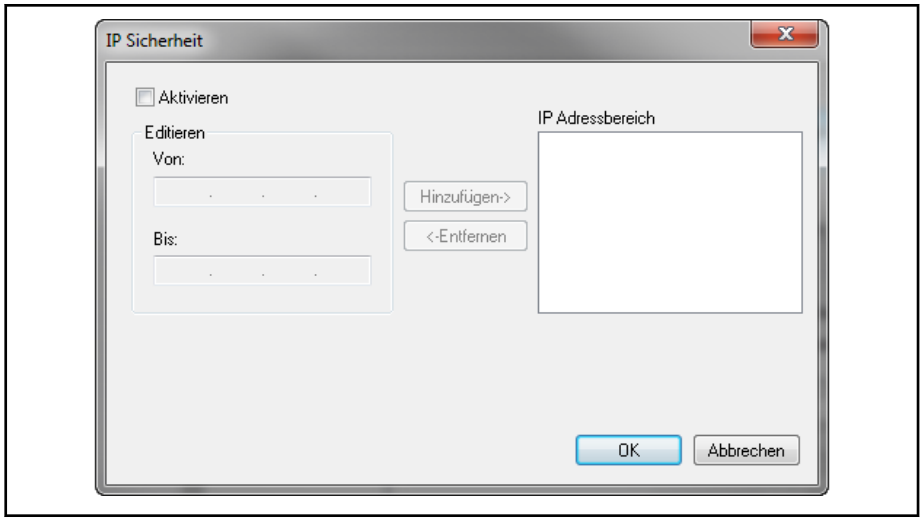


Abb. 15-23: Dialog IP-Sicherheit

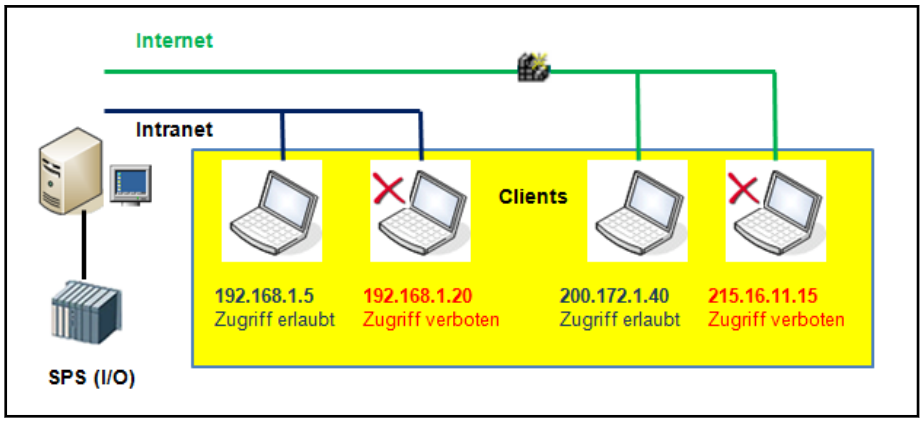


Abb. 15-24: Zugriffserlaubnis nach IP-Adresse

Methode 4: Verschlüsselte Kommunikation (SSL)

Durch Aktivierung des Web Tunneling Gateway (WTG) kann die gesamte Kommunikation zwischen Datenserver + Webserver und dem Thin Client mithilfe von RC6, einem höchst sicheren 128 Bit Verschlüsselungsstandard, verschlüsselt werden. Zum Verwenden von SSL gehen Sie wie folgt vor:

1. Klicken Sie im Dialog "Web-Einstellungen" auf **Weitere Einstellungen**. Wählen Sie die Option **Web Tunneling Gateway aktiviert** aus. Klicken Sie auf das Optionsfeld **SSL** und stellen Sie sicher, dass der SSL-Port auf 443 eingestellt ist. Klicken Sie auf **OK**.

Bereitstellung als Web-Anwendung

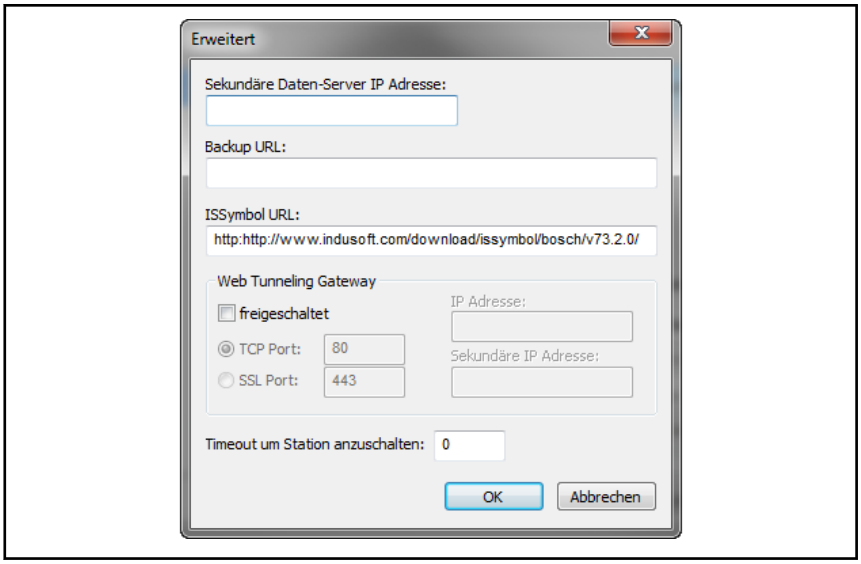


Abb. 15-25: Web-Einstellungen – Dialog Erweiterte Einstellungen

2. Vergewissern Sie sich, dass SSL-Fähigkeiten auf Ihrem Webserver aktiviert sind, und dass ein SSL-Echtheitszertifikat vorliegt.
3. Stellen Sie sicher, dass SSL im Web-Client aktiviert ist.
4. Richten Sie alle anderen Web-Konfigurationen zum Unterstützen des WTG ein.

Methode 5: VPN

Ein VPN ist ein virtuelles, privates Netzwerk (Virtual Private Network). Es wird alles virtuell bezeichnet, da es in Wirklichkeit das öffentliche Internet für den Datentransport von einem Rechner zum anderen verwendet. Da dieses Netzwerk aber verschlüsselt ist, andere vom Internetanbieter aktivierte Sicherheitsmechanismen verwendet, ist es ein sehr sicheres privates Netzwerk. Da VPNs von sich aus sicher sind, sind sie teurer als eine einfache öffentliche Internetverbindung.

15.11 Port-Belegung

"WinStudio" und/oder damit verbundene Software verwenden verschiedene Ports. Dies sind:

Port-Nr.	Programm
20	FTP-Server (Daten)
21	FTP-Server (Befehl)
25	SMTP-Server
80	Microsoft IIS Server für HTTP-Pakete
110	POP3
118	Microsoft SQL Server Services
161	SNMP
162	SNMP Trap
389	LDAP
443	Microsoft IIS Server für HTTP-Pakete (SSL)
502	Modbus TCP/IP-Protokoll

Port-Nr.	Programm
663	LDAP über SSL
1028	FTP-Client (Befehl)
1029	FTP-Client (Daten)
1234	Projekt-TCP/IP-Server
1443	Microsoft SQL Server
1444	Microsoft SQL Server Standard-Port (Monitor)
1521	Oracle
1526	Oracle
2030	Oracle
3001	A-B Ethernet TCP/IP Protokoll (Standard)
3306	MySQL (kann zur Verwendung von 3306-3309 konfiguriert werden)
3872	Oracle Management Remote Agent
3997	Studio ADO Gateway
4322	Remote Agent (CEServer)
5432	PostgreSQL
47808	BACNet UDP Protocol (Standard)

Gegebenenfalls muss die Belegung eines oder mehrerer dieser Ports in die Firewall-Einstellungen eingearbeitet werden.

15.12 Übung: Betrachten Ihres Projekts im Web

Zur Anzeige Ihres Projekts gehen Sie wie folgt vor:

1. Erweitern Sie den Ordner "WinStudio" und führen Sie einen Doppelklick auf Ihrem Bild **main.scr** aus.
2. Zum Speichern des Bildes im HTML-Format klicken Sie im Kontextmenü des Bildes auf **Als HTML speichern**.

   Webdateien werden im Ordner "Web" gespeichert.

   Öffnen Sie also den Ordner und überprüfen Sie, dass das Bild "Haupt" erfolgreich gespeichert wurde. Es sollte **main.html** angezeigt werden.
3. Öffnen Sie den Dialog "Web-Einstellungen".

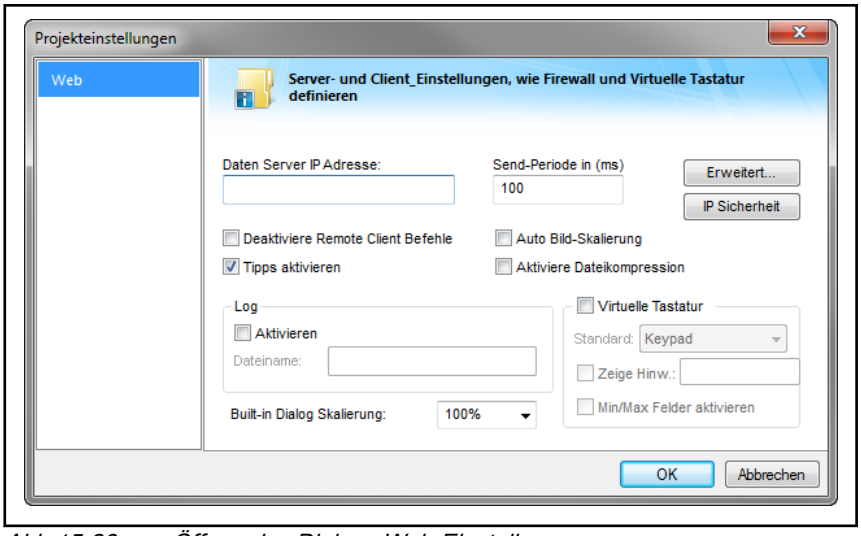


Abb. 15-26: Öffnen des Dialogs Web-Einstellungen

4. Konfigurieren Sie die **IP-Adresse Datenserver** zur Verwendung der IP-Adresse der Serverstation (Rechner, auf dem sie das Programm ausführen) während der Laufzeit. Die Thin-Client-Station tauscht mit der in diesem Feld festgelegten Station Online-Daten (Tag-Werte) aus.
5. Geben Sie im Feld **URL** den URL-Pfad zu Ihrer Datei **main.html** (im Ordner "Web") ein.

   Die URL hängt vom auf dem Webserver Ihrer Serverstation konfigurierten Startverzeichnis ab.

☞ Microsoft stellt Webserver für alle Microsoft-Betriebssysteme zur Verfügung. Informationen zum Installieren und Konfigurieren eines Webservers finden Sie in Ihrer Microsoft-Dokumentation.

6. Nach dem Konfigurieren der Webeinstellungen schließen Sie den Dialog durch Klicken auf **OK**.
7. Speichern und schließen Sie alle Bilder und Arbeitsblätter und überprüfen Sie dann das Projekt (**Anwendung verifizieren** im Kontextmenü von WinStudio), um die Webeinstellungen auf Ihrer Webseite zu aktualisieren.

HINWEIS	**Das Projekt muss nach jeder Änderung der Web-Einstellungen erneut verifiziert werden.**

Zum Testen Ihrer webbasierten Anwendung gehen Sie wie folgt vor:

1. Klicken Sie auf **Anwendung starten** (im Kontextmenü "WinStudio"), um Ihr Projekt lokal auf der Serverstation auszuführen.
2. Öffnen Sie einen Internetbrowser (Microsoft Internet Explorer oder Netscape) und geben Sie die URL-Adresse zum Öffnen Ihres Bildes haupt.html von der Serverstation ein.
3. Wenn der Dialog "Anmelden" im Browser angezeigt wird, geben Sie im Feld **Benutzername Gast** ein und klicken Sie auf **OK**, um das Bild haupt.html im Browser anzuzeigen.

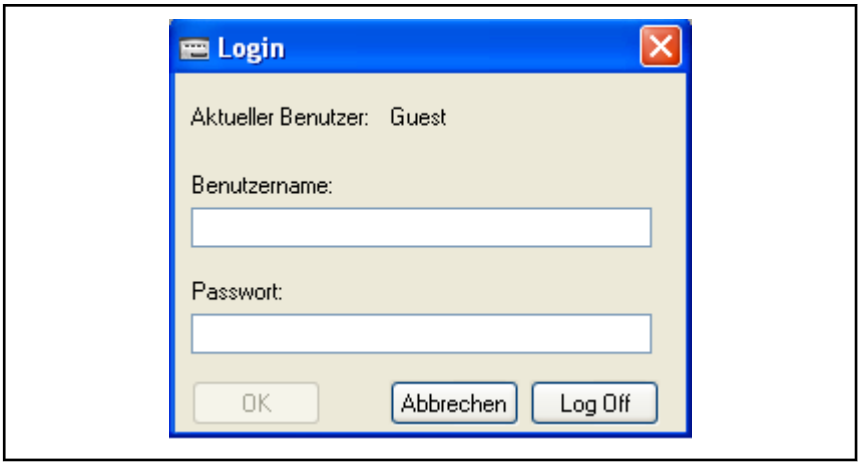


Abb. 15-27: Dialog Anmelden

Beachten Sie, dass Sie die Ebene jedes Tanks lokal (Serverstation) mit dem Viewer-Laufzeitmodul oder per Fernzugriff (Thin Client) mit dem Browser verändern können.

☞ Ein Thin Client erfordert eine ActiveX-Komponente (**ISSymbol.ocx**) zum Handhaben von Bildern im Browser. Wenn Sie den Thin Client mit dem Internet verbinden, wird die Komponente automatisch herunter geladen und registriert.

16 Herunterladen auf ein Remote-Gerät

16.1 Konfigurieren des Zielsystems

Nach dem Konfigurieren und lokalen Testen eines Projekts (auf Ihrer Entwicklungs-Arbeitsstation) können Sie das Projekt auf eine Remote-Laufzeitstation herunterladen, auf der "WinStudio" auf Windows XP, XPe, W7 oder CEView auf einem Windows-Embedded-Gerät läuft.

1. Überprüfen Sie, ob der Remote Agent ("CEServer.exe") auf der Zielstation (Remote-Station) läuft.
   - Auf einer Windows Plattform befindet sich die Datei "CEServer.exe" im

     Ordner **[…]\WinStudio\Bin**.
   - Auf einem Windows-Embedded-Gerät befindet sich die Datei im Ordner **[…]\WinStudio**.
2. Führen Sie "CEServer.exe" auf der Zielstation aus und klicken Sie auf "Setup", wenn der Dialog **Remote Agent** angezeigt wird.

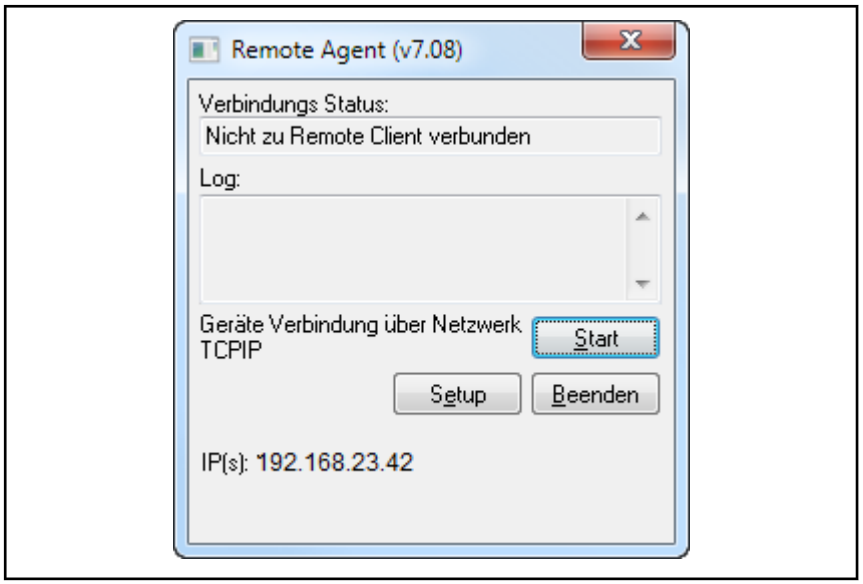


Abb. 16-1: Dialog Remote Agent

Der Dialog "Setup" wird angezeigt:

Herunterladen auf ein Remote-Gerät

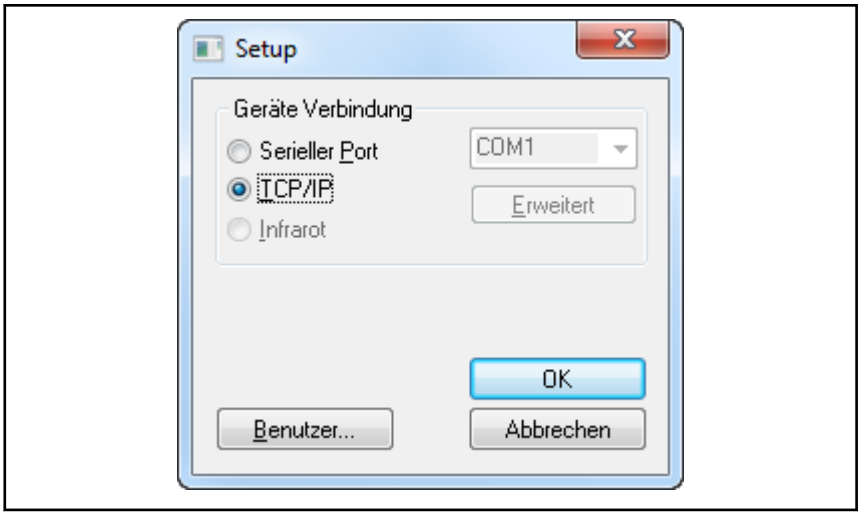


Abb. 16-2: Dialog Setup

Die Parameter in diesem Dialog dienen zur Konfiguration der Kommunikation zwischen Entwicklungs- und Zielsystemen:

- **Serieller Port**: Zum Herstellen einer seriellen Verbindung zum Projekt Server wählen Sie "Serieller Port" aus und wählen Sie dann den spezifischen Port aus der Auswahlliste aus. Durch Klicken auf **Erweitert** wird der Dialog "Erweiterte Einstellungen" angezeigt.

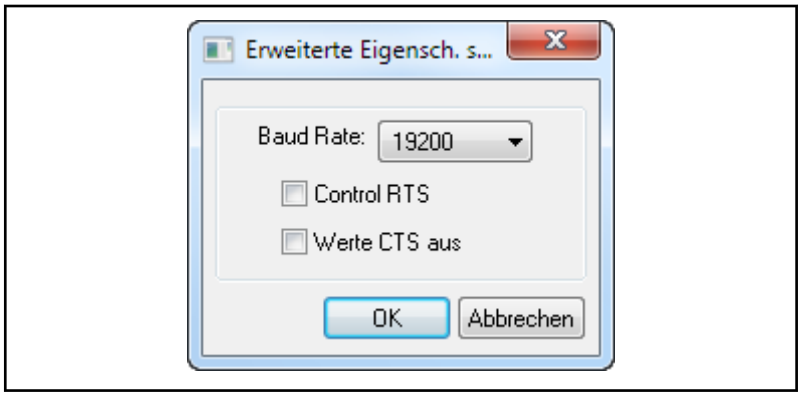


Abb. 16-3: Dialog Erweiterte Einstellungen

Die Parameter in diesem Dialog dienen der Steuerung des Flusses zwischen Ihren Ziel- und Entwicklungsstationen: Ändern Sie diese bei Bedarf und klicken Sie auf **OK** zum Schließen des Dialogs "Seriell - Weitere Einstellungen".

- **TCP/IP**: Aktivieren Sie diese Option, um die Verbindung zur Entwicklungsstation über TCP/IP herzustellen.



Für bessere Leistung empfehlen wir die Verwendung einer TCP/IP-Verbindung anstelle einer seriellen Verbindung.

---

3. Nach Beendigung klicken Sie zum Schließen des Dialogs "Setup" auf **OK**. Lassen Sie das Programm Remote Agent aber in der Remote-Arbeitsstation weiter laufen.

Weiter mit „Konfigurieren der Entwicklungsstation auf Seite 489“.

16.2 Konfigurieren der Entwicklungsstation

Nach dem Konfigurieren der Zielgeräte zum Empfangen von Daten befolgen Sie die folgenden Schritte, um die Projektdaten von der Entwicklungsstation auf das Zielgerät zu übertragen:

1. Wählen im Kontextmenü von WinStudio den Eintrag **WinStudio-Zielplattform** . Der Dialog "Remote-Verwaltung" wird angezeigt.

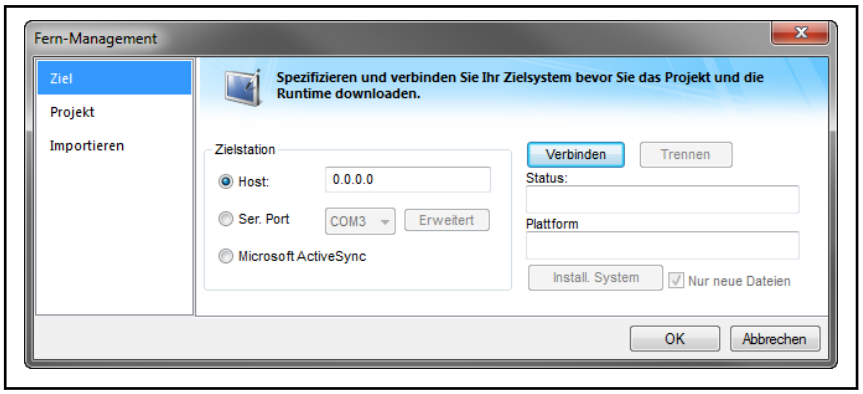


Abb. 16-4: Dialog Remote-Verwaltung

2. Wählen Sie die Registerkarte **Ziel** und verwenden Sie die folgenden Optionen zum Festliegen des Zielsystems, mit dem Sie eine Verbindung aufbauen möchten.
   - **Host**: Wählen Sie diese Option aus und geben sie die IP-Adresse des Zielsystems in das Textfeld ein, wenn Sie beim Konfigurieren des Zielsystems eine TCP/IP-Verbindung angegeben haben.
   - **Serieller Port**: Wählen Sie diese Schaltfläche aus und wählen Sie einen Port aus der Liste der Combobox aus, wenn Sie in beim Konfigurieren des Zielsystems eine serielle Verbindung angegeben haben.
   - **Microsoft ActiveSync**: Wählen Sie diese Schaltfläche zum Verbinden mit Geräten aus, auf denen ActiveSync aktiviert ist.
3. Klicken Sie auf die Schaltfläche **Verbinden**, um eine Verbindung mit dem Zielsystem herzustellen.
4. Wählen Sie die Registerkarte **Projekt** aus.
5. Überprüfen Sie im Feld **Ziel** das Verzeichnis für die Projektdateien auf dem Zielsystem. Wenn der Pfad nicht korrekt ist, klicken Sie auf die Suchschaltfläche rechts neben dem Feld **Ziel**, um ein neues Verzeichnis auszuwählen. (Wenn Sie korrekt mit dem Zielsystem verbunden sind, sollten Sie es wie alle anderen Netzlaufwerke durchsuchen können.)

   Sie können die Projektdateien nur in Ordner herunterladen, die sich unterhalb der Ebene befinden, auf der CEServer installiert ist und ausgeführt wird.

   Wenn CEServer zum Beispiel unter **\StorageCard\CEView** installiert ist, sollten die Projektdateien unter **\StorageCard\CEView\<projekt_name>** installiert werden.
6. Klicken Sie auf **Download**, um das gesamte Projekt auf das Zielsystem herunter zuladen. (Oder klicken Sie auf **Datei senden**, um eine einzelne Datei zu senden.)

HINWEIS

Beim Senden eines Projekt an das Zielsystem ersetzt das neue Projekt automatisch und sofort das alte, selbst wenn das alte Projekt noch ausgeführt wird.

Wenn außerdem die Option Nur neuere Dateien abgewählt (deaktiviert) wird, löscht die Entwicklungsanwendung vor dem Herunterladen der neuen Dateien alle alten Dateien im Projektordner.

Im Zweifelsfall ist es sinnvoll, das alte Projekt (durch Klicken auf Stopp) manuell anzuhalten, bevor das neue gesendet wird. Auf diese Weise stellen Sie sicher, dass die Anwendung ordnungsgemäß beendet wurde und durch das Löschen von Dateien nicht beeinträchtigt wird.

Es ist allerdings nicht unbedingt notwendig, das Projekt anzuhalten. Wenn Ihr Projekt robust konfiguriert wurde, können Sie neue Dateien zu jedem beliebigen Zeitpunkt senden.

- Nach der erstmaligen Konfiguration der Einstellungen für die Remote-Verwaltung können Sie mit **Projekt an Ziel senden** aktualisierte Projektdateien ohne Öffnen dieses Dialogs senden.
- Wenn das Herunterladen unterbrochen wird, fordert die Entwicklungsanwendung eine Bestätigung zum Fortfahren an und warnt, dass das Projekt ggf. nicht ordnungsgemäß läuft.

7. Klicken Sie nach Abschluss des Herunterladens auf **Start**, um das Projekt auf dem Zielsystem zu starten.

16.3 Automatisches Ausführen eines Projekts

Standardmäßig müssen Sie Ihr fertiges Projekt manuell auf dem Windows-Embedded-Gerät ausführen, indem Sie sich entweder von Ihrem Entwicklungsrechner über den Dialog WinStudio-Zielplattform mit dem Zielgerät verbinden und dort die Anwendung starten(siehe oben) oder am Gerät selbst durch Klicken auf die Schaltfläche **Start** im Dialog "Remote Agent".

Allerdings kann das Windows-Embedded-Gerät für das automatische Ausführen eines festgelegten Projekts konfiguriert werden. Dazu ergänzen Sie die Datei **CEServer.ini** auf dem Windows-CE-Gerät um die folgende Einstellung:

[Setup]

AppName=Pfad Anwendung

Wobei **projekt_name** der Speicherort der "WinStudio"-Projektdateien auf dem Windows-Embedded-Gerät ist.

Zum Beispiel:

[Setup]

AppName=\StorageCard\Test\CEServerTest

Beim nächsten Hochfahren des Windows-Embedded-Geräts und Öffnen des Dialogs "Remote Agent" (**CEServer.exe**) wird diese Einstellung gelesen und das angegebene Projekt automatisch ausgeführt.

Die Datei **CEServer.ini** kann auf drei Arten bearbeitet werden:

- Direktes Bearbeiten der Datei auf dem Windows-Embedded-Gerät mithilfe einer angeschlossenen Tastatur oder des Tastenfelds auf dem Touchscreen. Die Datei sollte sich im gleichen Verzeichnis befinden wie die <Datei CEServer.exe>.
- Verbinden Sie das Windows-Embedded-Gerät als gemeinsam benutztes Laufwerk mit Ihrem Rechner und bearbeiten Sie die Datei von dort aus.
- Bearbeiten Sie die Datei im Verzeichnis **[...]\WinStudio\ Redist** "bevor" Sie die Systemdateien auf dem Windows-Embedded-Gerät installieren.

HINWEIS	**Diese letzte Methode ändert die Standardkopie von CEServer.ini, die in "WinStudio" enthalten ist.**

Verwenden Sie diese Methode nur, wenn:

- die Dateien vor dem Bearbeiten gespeichert wurden;
- die gleichen Systemdateien auf mehreren identischen Windows-Embedded-Geräten installiert werden und
- wenn Sie bereits den Speicherort (Dateipfad) der "WinStudio"-Projektdateien auf dem Gerät kennen (möglicherweise durch Verwendung der normalen Installationsmethode auf einem Testgerät).

17 Datenbankschnittstelle

17.1 Datenbankschnittstelle – Allgemeines

Das Konfigurieren einer Datenbankschnittstelle mit "WinStudio" besteht im Prinzip aus dem Verknüpfen von Tasks aus "WinStudio" (Alarme, Ereignisse oder Trends) mit Tabellen aus externen Datenbanken über einen speziellen Datenbankanbieter, der die gewählte Datenbank unterstützt.

Jeder historische Task (Alarm, Ereignis oder Trend) kann zum Speichern von Daten konfiguriert werden, entweder in Dateien mit dem proprietären Format von "WinStudio" oder in externe relationale SQL-Datenbanken. Verwenden Sie die Registerkarte Optionen zum Konfigurieren der Datenbank für das Speichern der Alarm- und Ereignishistorie. (Eine Anleitung zum Speichern der Historie für Trend-Tasks finden Sie unter Ordner Trend auf Seite 301.)

"WinStudio" unterstützt ADO.NET, um eine intuitive, einfache, flexible und leistungsstarke Schnittstelle mit Standardtechnologien von MDAC (Microsoft Data Access Components) wie z.B. OLE-DB (Object Linking Embedded — Database) und ODBC (Open Database Connectivity) zur Verfügung zu stellen. Durch Verwendung dieser Fähigkeit können Sie Verbindungen zu allen MDAC-kompatiblen Datenbanken herstellen (eine Liste bereits getesteter Datenbanken finden Sie in der Konformitätstabelle).

Die folgenden Tasks unterstützen die Datenbankschnittstelle:

- **Alarme**: WinStudio kann Meldungen der Alarmhistorie in einer relationalen Datenbank speichern und/oder sie daraus abrufen.
- **Ereignisse**: WinStudio kann die Ereignismeldungen in einer relationalen Datenbank speichern und/oder sie daraus abrufen.
- **Trends**:WinStudio kann die Trendhistorie in einer relationalen Datenbank speichern und/oder sie daraus abrufen.
- **Viewer**: Datenbankinformationen können sowohl in tabellarischer (Objekte Alarm-/Ereignissteuerung auf Seite 290 und Raster auf Seite 345) als auch in grafischer Form (Objekt Trendsteuerung auf Seite 309) dargestellt werden.
- **Web**: Da die unten aufgeführten Elemente bereits in der Webschnittstelle von "WinStudio" verfügbar sind, können Sie eine Anwendung zum Speichern von Daten in einer relationalen Datenbank bereitstellen und diese über das Web laufen lassen.

Durch die Verwendung der eingebetteten Datenbankschnittstelle kann "WinStudio" auf einfache Weise Daten von der Produktion für Fremdsysteme (z.B. ERP) bereitstellen oder Daten von diesen einholen.

"WinStudio" kann mit jeder relationalen Datenbank kommunizieren, die von einem gültigen ADO.NET-Anbieter, OLE-DB-Anbieter oder ODBC-Treiber unterstützt wird. Die Konformitätstests wurden allerdings mit den folgenden Datenbanken durchgeführt:

Tabelle Konformitätstest

Datenbank	Datenbankversion	ADO.NET-Anbieter	Assembly Version
Microsoft SQL Server 2000	8.0	System.Data.SqlClient	1.0.5000.0
Microsoft Access 2000	9.0.3821 SR-1	System.Data.OleDbClient	1.0.5000.0
Microsoft Excel 2000	9.0.3821 SR-1	System.Data.OleDbClient	1.0.5000.0
Oracle	10g Release 1 for Windows	System.Data.OracleClient	1.0.5000.0

Datenbankschnittstelle

Datenbank	Datenbankversion	ADO.NET-Anbieter	Assembly Version
Sybase	Anywhere 9.0.1.1751	iAnywhere.Data.AsaClient	9.0.1.1751
MySQL	4.0.20a	ByteFX.MySqlClient	0.7.6.15073



Weitere Informationen zur Konfiguration einer bestimmten Datenbank finden Sie in den folgenden Abschnitten:

- Verwendung von ODBC-Datenbanken auf Seite 513
- Verwendung von Microsoft SQL Server auf Seite 515
- Verwendung von ORACLE-Datenbanken auf Seite 517
- Verwendung von Microsoft Access Datenbanken auf Seite 520
- Verwendung von SQL Server CE auf Seite 523
- Verwendung von Sybase auf Seite 525
- Verwendung von Microsoft Excel auf Seite 526
- Verwendung von MySQL auf Seite 530

17.2 Relationale SQL-Datenbanken

Eine relationale SQL-Datenbank besteht aus Datensätzen, die in Tabellen mit Feldern und Registern gespeichert sind, und unterstützt SQL-Befehle.

Jede Datenbank kann eine oder mehrere Tabellen enthalten. Jede Tabelle besteht aus Feldern (Spalten) und Registern (Zeilen). Üblicherweise sind die Felder vordefiniert und das Projekt fügt ein oder mehrere Register hinzu bzw. liest diese, je nach Abfragebedingung.

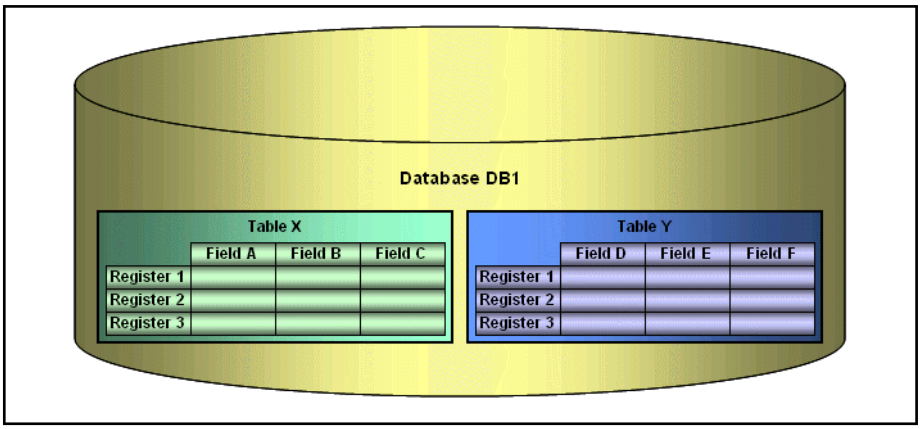


Abb.17-1: Relationale SQL-Datenbanken

"WinStudio" verwendet Datenbankanbieter (ADO.NET) als Schnittstelle zu relationalen SQL-Datenbanken. Datenbankanbieter sind Bibliotheken, die für den Datenzugriff auf verschiedene Datenbanken mit Hilfe von SQL-Befehlen entwickelt wurden. Der ADO-NET-Anbieter für eine bestimmte Datenbank kann vom Betriebssystem oder vom Datenbankhersteller zur Verfügung gestellt werden.

Die folgende Abbildung zeigt, wie "WinStudio" mit verschiedenen Datenbanken kommunizieren kann, unter Verwendung eines anderen Datenbankanbieters für jede Datenbank.

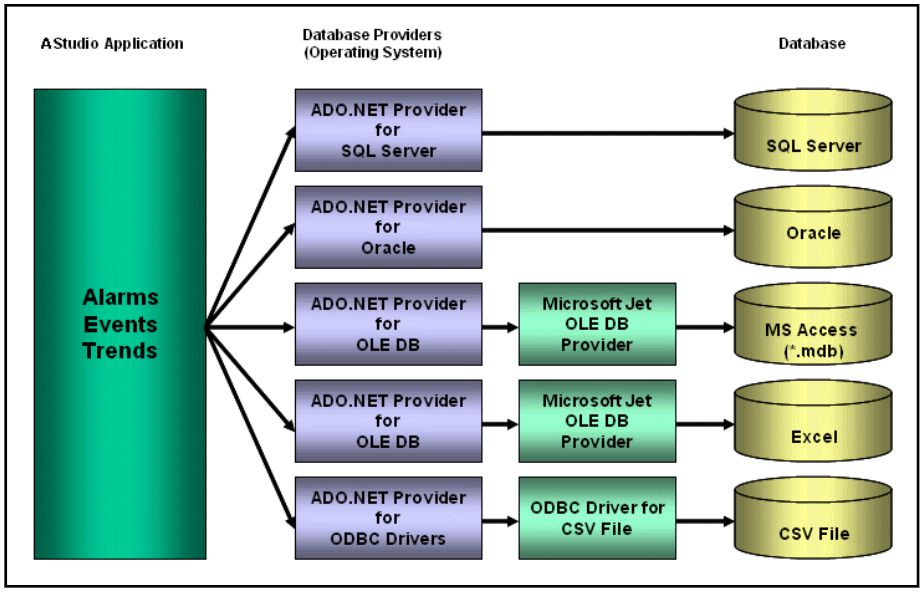


Abb.17-2: "WinStudio" mit verschiedenen Datenbanken...

Die obige Abbildung zeigt einige ADO-NET-Anbieter für Datenbanken. Es ist zu sehen, dass der "Microsoft ADO-NET-Anbieter für ODBC-Treiber" den Zugriff auf die Datenbank über einen ODBC-Treiber ermöglicht. Siehe Datenbank Anhang A: Verwendung von ODBC-Datenbanken auf Seite 513. Es ist auch möglich, dass Sie keinen ADO-NET-Anbieter haben, aber ein OLE-DB-Anbieter verfügbar ist. Durch Verwendung von "Microsoft ADO.NET Provider for OLE DB" erhalten Sie Zugriff auf die Datenbank; der Anbieter Microsoft Jet OLE DB ermöglicht mit diesem Ansatz Zugriff auf Applikationen im Microsoft-Office-Paket.



Es ist wichtig, dass "WinStudio" die Schnittstelle für ADO.NET-Anbieter zur Verfügung stellt. Allerdings müssen die ADO-NET-Anbieter bzw. die ODBC-Treiber/OLE-DB-Anbieter entweder vom Betriebssystem oder vom Datenbankhersteller zur Verfügung gestellt werden. Wenn Ihr Verbindungsstring sich nicht auf einen gültigen ADO.NET-Anbieter bezieht, wird der OLE.DB-Anbieter verwendet.

Obwohl die meisten Projekte in der Regel nur mit einem Datenbanktyp verknüpft sind, bietet "WinStudio" die Flexibilität, jeden Task mit einer anderen unterstützten Datenbank zu verknüpfen. Darüber hinaus müssen Sie sich bei Verwendung dieser Architektur keine Gedanken über die spezifischen Eigenschaften der einzelnen Datenbanken machen (dies wird größtenteils vom Datenbankanbieter der jeweiligen Datenbank oder von der Datenbank-Gateway-Schnittstelle von "WinStudio" übernommen). Daher sind die Projekteinstellungen größtenteils einheitlich, ungeachtet der von Ihnen jeweils ausgewählten Datenbank.

17.3 Verknüpfen der Datenbank über einen Remote-Datenbankanbieter

Je nach Architektur Ihres Projekts steht der ADO.NET-Anbieter für die relationale SQL-Datenbank evtl. nicht auf den gleichen Stationen, auf denen "WinStudio" ausgeführt wird, zur Verfügung. Dieses Szenario tritt besonders häufig auf, wenn das Projekt auf einem Windows-Embedded-Zielsystem ausgeführt wird (derzeit werden die meisten Anbieter nicht für Windows Embedded unterstützt). Zur Lösung dieses Problems haben wir eine flexible Lösung ent-

wickelt, mit der wie in der folgenden Abbildung gezeigt verteilte Systeme konfiguriert werden können:

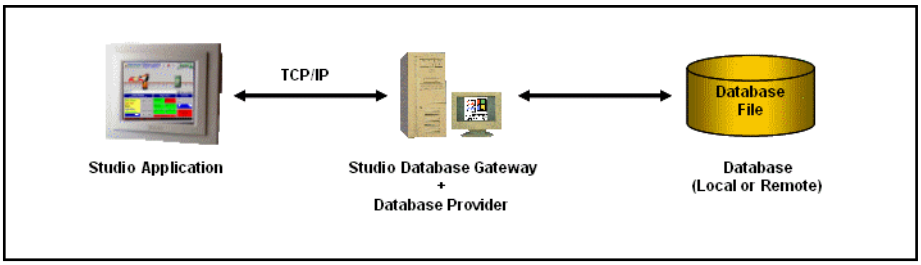


Abb.17-3: Verknüpfen der Datenbank über einen Remote-Datenbankanbieter

Das Projekt wird auf einer Serverstation ausgeführt (auf der "Rexroth WinStudio" oder CEView installiert ist). Das Projekt kann über TCP/IP mit dem (auf einem Remote-Rechner ausgeführten) Datenbank-Gateway von "WinStudio" kommunizieren. Der Gateway implementiert die Schnittstelle mit der Datenbank über den Datenbankanbieter, der auf dem Rechner, auf dem der Gateway ausgeführt wird, zur Verfügung steht.

Das Studio Datenbank-Gateway erfordert keine komplexe Konfiguration. Kopieren Sie einfach die Dateien **STADOSvr.exe** und **StADOSrv.ini** aus dem Unterordner **\BIN** von "WinStudio" und fügen Sie diese in ein beliebiges Verzeichnis des Computers, der als Gateway-Station verwendet werden soll, ein und führen Sie das Programm **STADOSvr.exe** aus. Für den "WinStudio" Datenbank-Gateway stehen erweiterte Einstellungen zur Verfügung, diese sollten aber nur unter besonderen Umständen geändert werden. Nähere Informationen zur Konfiguration der Einstellungen für den "WinStudio" Datenbank-Gateway finden Sie unter Studio-Datenbank-Gateway auf Seite 496.



Der Studio-Datenbank-Gateway ist ein TCP/IP-Server für das "WinStudio"-Projekt und verwendet standardmäßig den TCP-Port 3997. Beim Ausführen des Programms **STADOSvr.exe** kann eine andere Port-Nummer gemäß der folgenden Syntax festgelegt werden:

STADOSvr.exe Port-Nummer

Beispiel:

StADOSvr 3998

17.4 Studio-Datenbank-Gateway

Das Studio-Datenbank-Gateway ist ein TCP/IP-Server, der mit Hilfe von Microsoft .Net Framework mit Datenbanken kommuniziert. Er kann auf dem selben Rechner, auf dem das "WinStudio"-Projekt ausgeführt wird, oder auf einem anderen Rechner laufen. Der Eintrag "Host" im Bereich "Datenbank Gateway" der erweiterten Einstellungen (siehe Dialog Datenbankkonfiguration auf Seite 502) legt fest, ob das Gateway auf dem lokalen Rechner laufen soll. Wenn Sie den lokalen Rechner verwenden, sollten Sie als Host-Namen entweder "localhost" oder **127.0.0.1** eingeben. Über das Starten oder Stoppen des Gateways müssen Sie sich nicht kümmern, dies wird automatisch von den "WinStudio"-Tasks durchgeführt. Andererseits muss bei Remote-Ausführung des Gateways dieses manuell gestartet werden. Dazu kopieren Sie die Dateien **StADOSvr.exe** und **StADOSrv.ini** aus dem Ordner **\BIN** in das Remote-Verzeichnis des Computers und führen das Programm **StADOSvr.exe** aus.

Das Gateway kann mehrere Male für verschiedene TCP/IP-Port-Nummern gestartet werden. Die Standard-Port-Nummer ist 3997 und kann durch Anga-

be der gewünschten Port-Nummer in der Eingabeaufforderung geändert werden (z.B. **StADOSvr 1111**). Bei Ausführung von StADOSvr wird das folgende Symbol im Infobereich eingefügt:



Abb.17-4: Studio-Datenbank-Gateway

Durch einen Rechtsklick auf das Symbol wird das folgende Kontextmenü angezeigt:



Abb.17-5: Studio-Datenbank-Gateway, Kontextmenü

Die Option **Verbergen** steuert, ob das Fenster für das "Studio-Datenbank-Gateway" auf dem Desktop angezeigt wird. (Die Gateway-Software läuft nach dem Starten kontinuierlich, auch wenn das Fenster nicht angezeigt wird.) Durch Deaktivieren der Option **Verbergen** wird das Fenster angezeigt:

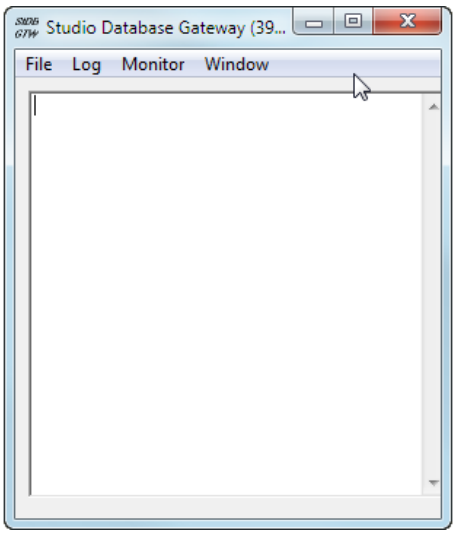


Abb.17-6: Studio-Datenbank-Gateway

Alle bei Datenbankoperationen auftretenden Fehler werden sowohl in diesem Fenster als auch im Fenster "LogWin" angezeigt. Die Meldungen werden durch vom ADO.NET-Anbieter erzeugte Ausnahmen berichtet. (Weitere Informationen zu Fehlermeldungen im Gateway-Modul finden Sie unter Fehlerbeseitigung Datenbank auf Seite 509.)

Die Ausgabe in diesem Fenster kann im Menü **Log** konfiguriert werden:

- Menüoption **Protokoll anzeigen**: Zeigt die Protokolldateien des "WinStudio" Datenbank-Gateways an.
- Menüoption **Optionen**: Öffnet den Dialog "Meldungen konfigurieren".

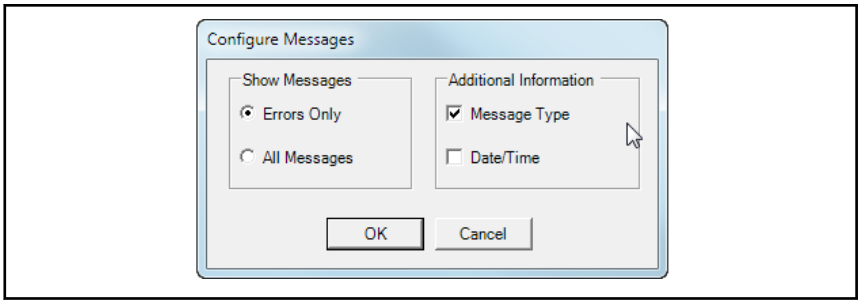


Abb.17-7: Studio-Datenbank-Gateway: Dialog Meldungen konfigurieren

- "Meldungen anzeigen": Wählen Sie **Nur Fehler**, um nur Fehlermeldungen im Protokoll anzuzeigen, oder wählen Sie **Alle Meldungen**, um alle Datenbankmeldungen anzuzeigen.
- "Zusatzinformationen": Hier können Sie Anzeige von Zusatzinformationen zu jeder Datenbankmeldung konfigurieren.
  - **Meldungsart**: Durch Anklicken (Aktivieren) dieser Option wird die Meldungsart angezeigt.
  - **Datum/Uhrzeit**: Durch Anklicken (Aktivieren) dieser Option wird der Zeitstempel der Meldung angezeigt.

Weitere Einstellungen

Das Studio-Datenbank-Gateway verfügt über erweiterte Einstellungen, die in der Datei "StADOSvr.ini" konfiguriert werden. Wenn Probleme mit der Kommunikation mit einer bestimmten Datenbank auftreten, müssen Sie wahrscheinlich einige dieser Einstellungen ändern oder der Datei neue Anbieter hinzufügen. Folgende Parameter stehen zur Verfügung:

Abschnitt der .INI-Datei	Parameter	Akzeptierte Werte	Beschreibung
Providers (1)	SaveMSec	1 : Deaktivieren 2 : Aktivieren 3 : Separate Spalte	Diese Einstellung legt das Standardverhalten des Anbieters beim Speichern von Millisekunden fest. Diese Standardeinstellung kann in den Weiteren Einstellungen in den Dialogen zur Datenbankkonfiguration geändert werden.
	Assembly	Alle Zeichenketten, die eine .Net Framework Assembly enthalten	Assembly-Option für alle Anbieter. Die Assembly verfügt über alle zur Kommunikation mit der Datenbank benötigten Klassen. Die meisten Anbieter befinden sich in der Assembly System.Data.
	ConnectionClass	Alle Verbindungsklassen in der Assembly	Die Verbindungsklasse implementiert die Schnittstelle System.Data.IDbConnection.
	DateAdapterClass	Alle Datenadapterklassen in der Assembly	Die Datenadapterklasse wird für Operationen verwendet, bei denen eine Aktualisierung der Datenbank erforderlich ist. Sie muss mit der angegebenen Verbindungsklasse kompatibel sein und sollte IDbDataAdapter implementieren.
	CommandBuilder-Class	Alle CommandBuilder-Klassen in der Assembly	Die CommandBuilder-Klasse ist außerdem für Aktualisierungen in Datenbanken verantwortlich. Sie muss mit der Verbindungsklasse kompatibel sein.
	Provider	Name des Anbieters	Einer der Parameter im Verbindungs-String ist der „Provider“ (Anbieter). Der Studio ADO Gateway vergleicht den Wert im Verbindungs-String mit dem Wert für diesen Parameter in jedem Anbieter und legt fest, welcher verwendet wird.
	ColumnDelimiter-Prefix	Beliebige Zeichen oder Zeichengruppen	Angabe eines Zeichens, das den Spaltennamen in SQL-Statements vorangestellt wird
	ColumnDelimiterSuffix	Beliebige Zeichen oder Zeichengruppen	Angabe eines Zeichens, das den Spaltennamen in SQL-Statements nachgestellt wird
	TableDelimiterPrefix	Beliebige Zeichen oder Zeichengruppen	Angabe eines Zeichens, das den Tabellennamen in SQL-Statements vorangestellt wird
	TableDelimiterSuffix	Beliebige Zeichen oder Zeichengruppen	Angabe eines Zeichens, das den Tabellennamen in SQL-Statements nachgestellt wird
	ValueString	Beliebige Zeichenkette	Dieser Wert gibt an, wie Konstanten in SQL-Statements bezeichnet werden. Für Microsoft-SQL-Datenbanken z.B. sollte der Wert @Wert sein, für ODBC ein Fragezeichen (?)
	ValueStringPrefix	Beliebige Zeichenkette	Dieser Wert gibt den Präfix an, der vor den Werten verwendet wird. Oracle-Werte erfordern z.B. diesen Präfix. Die SQL-Statements verwenden Wertbezeichner durch Verwendung von deren Präfixen, die Parameter in der Verbindungsklasse verwenden den Präfix aber nicht.

Datenbankschnittstelle

Abschnitt der .INI-Datei	Parameter	Akzeptierte Werte	Beschreibung
Providers (2)	ValueAddNumber	0 oder 1	Gibt an, ob dem ValueString eine fortlaufende Nummer zur Kennzeichnung der Parameter hinzugefügt wird oder nicht. Für Microsoft-SQL-Datenbanken sollte dieser Parameter den Wert 1 haben, da Parameter durch die Verwendung von @Wert1, @Wert2, ..., @WertN gekennzeichnet werden. Für ODCB sollte dieser Parameter 0 sein.
	BoolType	Alle Zeichenketten, die einen gültigen Datentyp für die Datenbank darstellen	Der in diesem Parameter festgelegte Datentyp wird beim Erstellen von Spalten zum Speichern von booleschen Werten verwendet. Stellen Sie sicher, dass der angegebene Datentyp boolesche Werte speichern kann.
	IntegerType	Alle Zeichenketten, die einen gültigen Datentyp für die Datenbank darstellen	Der in diesem Parameter festgelegte Datentyp wird beim Erstellen von Spalten zum Speichern von ganzzahligen Werten verwendet. Stellen Sie sicher, dass der hier angegebene Datentyp 32-bit-Werte speichern kann.
	RealType	Alle Zeichenketten, die einen gültigen Datentyp für die Datenbank darstellen	Der in diesem Parameter festgelegte Datentyp wird beim Erstellen von Spalten zum Speichern von reellen Werten verwendet. Stellen Sie sicher, dass der hier angegebene Datentyp reelle 64-bit-Werte speichern kann.
	StringType	Alle Zeichenketten, die einen gültigen Datentyp für die Datenbank darstellen	Der in diesem Parameter festgelegte Datentyp wird beim Erstellen von Spalten zum Speichern von Zeichenketten-Werten verwendet. Stellen Sie sicher, dass der angegebene Datentyp die Anzahl der Zeichen, die Sie in ihrem Projekt speichern wollen, speichern kann.
	TimeStampType	Alle Zeichenketten, die einen gültigen Datentyp für die Datenbank darstellen	Der in diesem Parameter festgelegte Datentyp wird beim Erstellen von Spalten zum Speichern von Zeitstempelwerten verwendet.
	EnableTop	0 oder 1	Wird dieses Feld auf 1 eingestellt, setzt der ASO den TOP in das SQL-Statement zum Begrenzen der Anzahl der erforderlichen Register.
	SingleConnection	0 oder 1	Wird dieses Feld auf 1 eingestellt, öffnet der ADO nur eine Verbindung zur Datenbank, egal wie viele Tasks oder Rechner Dienste von der Datenbank anfordern. Die Synchronisation zwischen den Tasks wird vom Gateway durchgeführt und wenn diese Option aktiviert ist, können sie nicht gleichzeitig ausgeführt werden.

Abschnitt der .INI-Datei	Parameter	Akzeptierte Werte	Beschreibung
Communication	TimeOut	2	Zeitüberschreitung für die Ausführung von Einfüge- und Aktualisierungsoperationen
	LongTimeOut	5	Zeitüberschreitung für die Ausführung von Verbindungs- und Abfrageaktualisierungen
	SyncTimeOut	60	Zeitüberschreitung zum Ausführen der Synchronisation
Connection	RegBufSize	128	Größe des vom Datenbank-API verwendeten internen Puffers.
Options	DisableCloseQuestion	0 oder 1	Wird dieses Feld auf 1 eingestellt, ist zum Schließen von "StADOSvr.exe" keine Bestätigung durch den Benutzer erforderlich. Dies ist wichtig für Geräte, die über alternative Methoden zum Verlassen und Neustarten von Anwendungen verfügen.

Die Parameter sind in vier Abschnitte unterteilt – "Providers", "Communication", "Connection" und "Options" – aber die Parameter zur Konfiguration von Datenbankanbietern sind allesamt im Dateiabschnitt "Providers" aufgeführt. Die Standardwerte für alle Parameter werden am Anfang der Datei unter Verwendung des Präfix „Default“ festgelegt wie nachfolgend gezeigt:

Anbieter

```
[Providers]
DefaultSaveMSec=3
DefaultAssembly=System.Data
DefaultConnectionClass=System.Data.OleDb.OleDbConnection
DefaultDataAdapterClass=System.Data.OleDb.OleDbDataAdapter
DefaultCommandBuilderClass=System.Data.OleDb.OleDbCommandBuilder
DefaultValueString=@Value
DefaultValueAddNumber=1 DefaultBoolType=INTEGER
DefaultIntegerType=INTEGER DefaultRealType=REAL
DefaultStringType=VARCHAR(255) DefaultTimeStampType=DATETIME
DefaultSingleConnection=0
```

Das nächste Element in der Datei gibt die Anzahl der Anbieter an:

Count=5

Die Anbieter werden durch den Parameter „Provider“, gefolgt von einer Nummer, gekennzeichnet. Beim Herstellen einer Verbindung zu einer Datenbank wird der Anbieterparameter im Verbindungs-String mit der Kennzeichnung des Anbieters verglichen, um festzustellen, welcher Anbieter verwendet werden soll. Wenn es keinen Anbieter mit dem Wert aus dem Verbindungs-String gibt, werden alle Standardwerte angenommen. Außer seiner Bezeichnung kann jeder Anbieter für jeden Parameter seinen eigenen Wert haben. Auch hier wird der Standard verwendet, wenn kein Wert angegeben ist. Nachfolgend ein Beispiel mit sieben Anbietern:

Beispiel mit sieben Anbietern

```
Count=7

Provider1=MICROSOFT.JET.OLEDB
SaveMSec1=3
ColumnDelimiterPrefix1=[
ColumnDelimiterSuffix1=]
SingleConnection1=1

Provider2=SQLOLEDB
```

Datenbankschnittstelle

```
ConnectionClass2=System.Data.SqlClient.SqlConnection
DataAdapterClass2=System.Data.SqlClient.SqlDataAdapter
CommandBuilderClass2=System.Data.SqlClient.SqlCommandBuilder
ColumnDelimiterPrefix2=[
ColumnDelimiterSuffix2=]
TableDelimiterPrefix2=[
TableDelimiterSuffix2=]
RealType2=FLOAT

Provider3=MSDASQL
ConnectionClass3=System.Data.Odbc.OdbcConnection
DataAdapterClass3=System.Data.Odbc.OdbcDataAdapter
CommandBuilderClass3=System.Data.Odbc.OdbcCommandBuilder
ValueString3=?
ValueAddNumber3=0
StringType3=VARCHAR(128)
EnableTop3=0

Provider4=ORAOLEDB
Assembly4=System.Data.OracleClient
ConnectionClass4=System.Data.OracleClient.OracleConnection
DataAdapterClass4=System.Data.OracleClient.OracleDataAdapter
CommandBuilderClass4=System.Data.OracleClient.OracleCommandBuilder
ValueString4=Value
ValueAddNumber4=1
ValueStringPrefix4=:
BoolType4=Number(1)
IntegerType4=Number(10)
RealType4=Number
StringType4=VARCHAR(255)
TimeStampType4=TIMESTAMP(0)
EnableTop4=0

Provider5=ASAPROV
Assembly5=iAnywhere.Data.AsaClient
ConnectionClass5=iAnywhere.Data.AsaClient.AsaConnection
DataAdapterClass5=iAnywhere.Data.AsaClient.AsaDataAdapter
CommandBuilderClass5=iAnywhere.Data.AsaClient.AsaCommandBuilder
ValueString5=?
ValueAddNumber5=0
ColumnDelimiterPrefix5=[
ColumnDelimiterSuffix5=]
TableDelimiterPrefix5=[
TableDelimiterSuffix5=]

Provider6=MYSQLPROV
Assembly6=ByteFX.MySqlClient
ConnectionClass6=ByteFX.Data.MySqlClient.MySqlConnection
DataAdapterClass6=ByteFX.Data.MySqlClient.MySqlDataAdapter
CommandBuilderClass6=ByteFX.Data.MySqlClient.MySqlCommandBuilder
ValueString6=@Value
ValueAddNumber6=1
StringType6=VARCHAR(128)
EnableTop6=0

Provider7=MSDAORA
Assembly7=System.Data.OracleClient
ConnectionClass7=System.Data.OracleClient.OracleConnection
DataAdapterClass7=System.Data.OracleClient.OracleDataAdapter
CommandBuilderClass7=System.Data.OracleClient.OracleCommandBuilder ValueString7=Value
ValueAddNumber7=1
ValueStringPrefix7=:
BoolType7=Number(1)
IntegerType7=Number(10)
RealType7=Number
StringType7=VARCHAR(255)
TimeStampType7=TIMESTAMP(0)
EnableTop7=0
```

17.5 Datenbankkonfiguration

Datenbankverbindung konfigurieren

Der Austausch von Daten mit einer externen Datenbankdatei erfolgt analog zu den Gerätetreibern. Zunächst müssen Sie eine neue Datenbankverbindung konfigurieren. Anschließend können Sie Arbeitsblätter erstellen, in denen Sie die zuvor erstellte Datenbankverbindung auswählen, und WinStudio-Tags mit einzelnen Datensätzen der externen Datenbank verknüpfen können.

Um eine neue Datenbankverbindung zu konfigurieren, öffnen Sie das Kontextmenü des Ordners **WinStudio ▸ Tasks ▸ Externe Datenbanken** und wählen den Eintrag "Datenbankverbindung konfigurieren" aus.



Der Ordner "Externe Datenbanken" ist nur verfügbar, wenn Sie in den WinStudio-Eigenschaften als Zielsystem "WinStudio 1.5k" oder höher auswählen und die Geräteversion in den VPx-/VEx-Eigenschaften "13V06" oder höher ist.

Im Übersichtsdialog können Sie mit der Schaltfläche "Hinzufügen" den Konfigurationsdialog für eine neue Datenbankverbindung öffnen. Geben Sie dort im Feld "Name" einen beliebigen Namen ein, den Sie später in den Arbeitsblättern auswählen können.

Mit Hilfe des Dialogs "Datenbankkonfiguration" können Sie die erforderlichen Einstellungen zum Anbinden von "WinStudio" an eine externe Datenbankdatei konfigurieren.

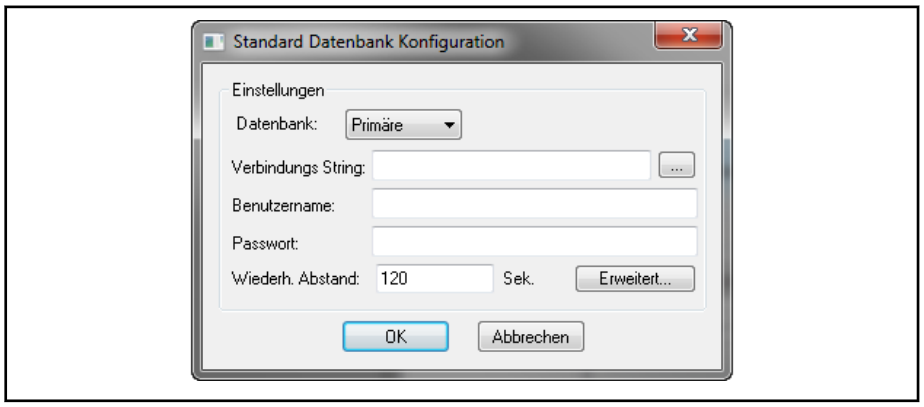


Abb.17-8: Dialog Datenbankkonfiguration

- Combobox **Datenbank**: Zur Auswahl von Primär oder Sekundär. Bei der Auswahl "Primär" betreffen alle Einstellungen der Datenbankkonfiguration die Primärdatenbank-Schnittstelle. Ansonsten betreffen sie die Sekundärdatenbank-Schnittstelle. Die Sekundärdatenbank können Sie in folgenden Modi konfigurieren:
  - **Deaktiviert**: In diesem Modus speichert "WinStudio" Daten nur in der Primärdatenbank. Wenn die Primärdatenbank aus irgendeinem Grund nicht zur Verfügung steht, werden die Daten nicht anderweitig gespeichert. Bei dieser Option können Daten verloren gehen, wenn die Primärdatenbank nicht verfügbar ist.
  - **Redundant**: In diesem Modus speichert "WinStudio" Daten in der Primär- und auch in der Sekundärdatenbank. Wenn eine dieser Datenbanken nicht zur Verfügung steht, speichert "WinStudio" die Daten weiterhin, und zwar nur in der zur Verfügung stehenden Datenbank. Wenn die nicht verfügbare Datenbank wieder verfügbar wird, synchronisiert "WinStudio" beide Datenbanken automatisch.
  - **Speichern und Weiterleiten**: In diesem Modus speichert "WinStudio" Daten nur in der Primärdatenbank. Wenn die Primärdatenbank nicht mehr zur Verfügung steht, speichert "WinStudio" die Daten in der Sekundärdatenbank. Wenn die Primärdatenbank wieder zur Verfügung steht, verschiebt "WinStudio" die Daten aus der Sekundär- in die Primärdatenbank.

☞ Bei der Primär- und Sekundärdatenbank kann es sich um unterschiedliche Datenbanktypen handeln. Sie müssen jedoch die gleichen Felder enthalten.

Durch Verwendung der Sekundärdatenbank können Sie die Systemzuverlässigkeit erhöhen und die Sekundärdatenbank als Backup verwenden, wenn die Primärdatenbank nicht zur Verfügung steht. Diese Architektur ist besonders nützlich, wenn die Primärdatenbank in der Remote-Station angesiedelt ist. In diesem Fall können Sie eine Sekundärdatenbank in der lokalen Arbeitsstation so konfigurieren, dass die Daten vorübergehend dort gespeichert werden, wenn die Primärdatenbank (z.B. während eines Netzausfalls) nicht zur Verfügung steht.

- **Projektstandard verwenden**: Wenn diese Option angehakt ist, verwendet "WinStudio" für den gerade konfigurierten Task die Einstellungen in der Standarddatenbank (Verbindungszeichenkette, Benutzername, Passwort, Intervall für Wiederholungsversuche und Weitere Einstellungen). Wenn diese Option nicht aktiviert ist, können diese Einstellungen individuell für den derzeitigen Task konfiguriert werden.
- **Verbindungszeichenkette**: In diesem Feld wird die Datenbank festgelegt, in die "WinStudio" Werte schreibt bzw. aus der es sie liest, sowie die Parameter, die es hauptsächlich beim Verbinden mit der Datenbank einsetzt. Statt der manuellen Eingabe der Verbindungszeichenkette kann auch die Durchsuchen-Schaltfläche (...) angeklickt und der Datenbanktyp aus dem Fenster **Datenverbindungseigenschaften** ausgewählt werden.

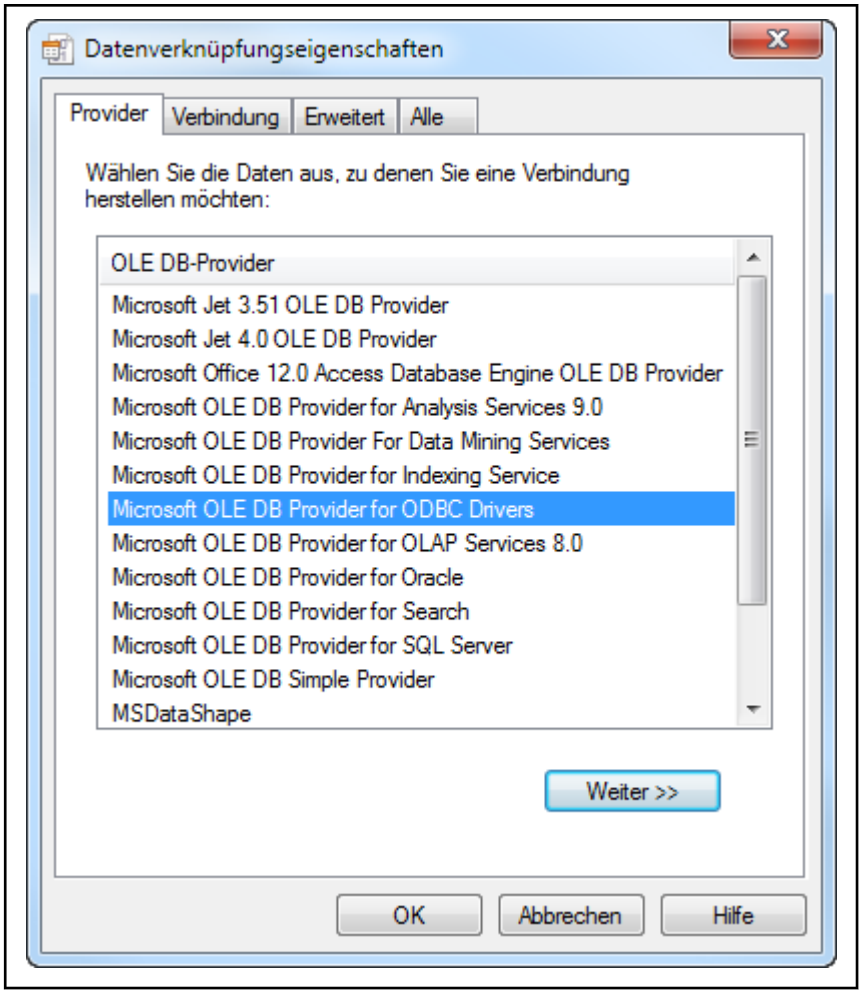


Abb. 17-9: Dialog Datenverbindungseigenschaften

Die im Fenster Datenverbindungseigenschaften aufgeführte Liste der Datenbankanbieter ist abhängig von den tatsächlich auf dem Rechner mit "WinStudio" installierten und verfügbaren Datenbankanbietern. Konsultieren Sie die Dokumentation des Betriebssystems (bzw. der Datenbank) für weitere Informationen über das Einstellen des Datenbankanbieters für die von Ihnen verwendete Datenbank.

- **Benutzername**: Der Benutzername, der für die Verbindung zur Datenbank verwendet wird. Der in diesem Feld eingetragene Benutzername muss zum Benutzernamen passen, der in der Datenbank konfiguriert ist.
- **Passwort**: Das Passwort, das für die Verbindung zur Datenbank verwendet wird. Das in diesem Feld eingetragene Passwort muss zum Passwort passen, das in der Datenbank konfiguriert ist.
- **Intervall für Wiederholungsversuche**: Wenn sich "WinStudio" aus irgendeinem Grund nicht mit der Datenbank verbinden kann, versucht es automatisch nach Verstreichen der in diesem Feld konfigurierten Sekunden erneut, sich mit der Datenbank zu verbinden.
- **Weitere Einstellungen**: Durch Klicken auf diese Schaltfläche erhalten Sie Zugang zur Konfiguration erweiterter Einstellungen. Für die meisten

Projekte muss der Standardwert dieser Einstellungen nicht geändert werden und sollte beibehalten werden.

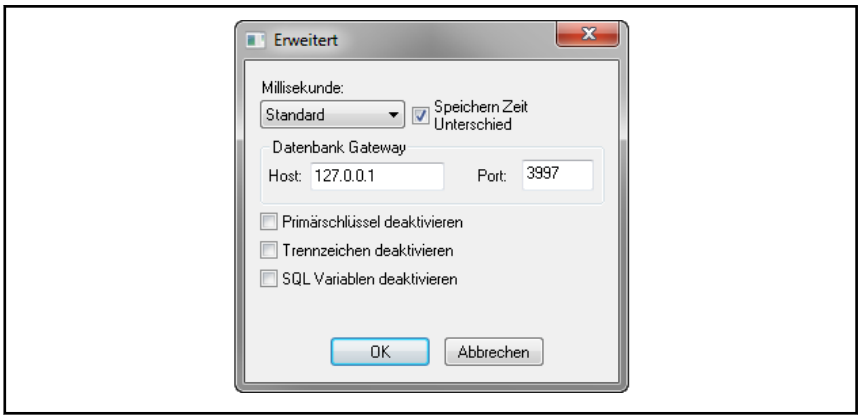


Abb. 17-10: Datenbankkonfiguration: Dialog Weitere Einstellungen

– **Millisekunden**

  Hier können Sie einstellen, wie die Millisekunden beim Speichern des Datums in der Datenbank abgespeichert werden. Verschiedene Datenbanken speichern das Datum in verschiedenen Formaten. Beispielsweise unterstützen manche Datenbanken in einem **Datumsfeld** keine Millisekunden. Die folgenden Optionen sind verfügbar:

  – **Standard**: Verwendet das für die aktuelle Datenbank vordefinierte Format. Die bereits von "Bosch Rexroth" getesteten Datenbanken sind mit der Option vorkonfiguriert, die jeweils am besten passt. Bei Auswahl von "Standard" verwendet "WinStudio" die für den derzeitigen Datenbanktyp vorkonfigurierte Einstellung. Wenn Sie eine Datenbank verwenden, die nicht vorkonfiguriert ist, wird bei Auswahl der Option Standard versucht, die Millisekunden in einem eigenen Feld abzuspeichern.

☞ Die Standardoption für jede Datenbank ist in der Datei **StADOSvr.ini** vorkonfiguriert; diese liegt im Unterverzeichnis **\BIN** von "WinStudio". Nähere Informationen zur Konfiguration der Datei **StADOSvr.ini** finden Sie unter Studio-Datenbank-Gateway auf Seite 496.

  – **Deaktivieren**: Speichert die Millisekunden gar nicht, wenn das Datum in der Datenbank abgespeichert wird.

  – **Aktivieren**: Speichert die Millisekunden im gleichen Feld, in dem das Datum gespeichert wird.

  – **Eigene Spalte**: Speichert die Millisekunden in einer eigenen Spalte. In diesem Fall wird das Datum (ohne Millisekunden) in einem Feld und die Anzahl der Millisekunden in einem anderen Feld gespeichert. Diese Option ist angezeigt, wenn Sie Zeitstempel in Millisekunden-Genauigkeit abspeichern wollen, die von Ihnen verwendete Datenbank für **Datumsfelder** aber keine Millisekunden unterstützt.

– **Zeitdifferenz speichern**: Wenn diese Option aktiviert ist (Standard), speichert "WinStudio" in jedem Register der Datenbank die Zeitzone des Rechners, auf dem das Projekt ausgeführt wird. Diese

Option muss aktiviert sein, um Probleme mit der Sommerzeit zu umgehen.

- **Datenbank-Gateway**: Tragen Sie hier den Hostnamen/ die IP-Adresse des Netzknotens ein, auf dem das "WinStudio"-Datenbank-Gateway laufen wird. Die Nummer des TCP-Ports kann zwar auch angegeben werden, aber wenn Sie nicht den Standard verwenden, müssen Sie das "WinStudio"-Datenbank-Gateway mit dem gleichen TCP-Port konfigurieren. Nähere Informationen zur Konfiguration der weiteren Einstellungen für das "WinStudio"-ADO-Gateway finden Sie unter Studio-Datenbank-Gateway auf Seite 496.
- **Primärschlüssel deaktivieren**: Für einige Module versucht "WinStudio", zum Beschleunigen der Abfragen einen Primärschlüssel für die Tabelle zu definieren. Wenn Sie eine Datenbank verwenden, die keine Primärschlüssel unterstützt (z.B. Microsoft Excel), dann sollten Sie diese Option durch Anklicken abwählen.
- **Feldbegrenzer deaktivieren**: Wählen Sie diese Fehlerbehandlungsoption zum Deaktivieren der Feldbegrenzer aus, die zur Formatierung der Kommunikation mit der Datenbank eingesetzt werden. Feldbegrenzer können zu Problemen führen, wenn ein Trend-Control bzw. ein Raster eine Abfrage aufbaut, die Gesamtsummen - wie z.B. Min und Max - beinhaltet.
- **SQL-Variablen deaktivieren**: Wählen Sie diese Fehlerbehandlungsoption zum Deaktivieren von SQL-Variablen - wie z.B. "@Value1" und "?" - aus, die in SQL-Befehlen und -Anfragen häufig eingesetzt werden. Einige bestimmte Datenbankanbieter unterstützen diese Variablen nicht.

Bereich Tabelle

In diesem Bereich können Sie die Einstellungen der Tabelle konfigurieren, in der die Daten gespeichert werden. Alle Tasks können die gleiche Datenbank verwenden. Jedoch muss jeder Task (Alarm-, Ereignis-, Trend-Arbeitsblätter) mit jeweils einer eigenen Tabelle verknüpft werden. "WinStudio" prüft dieses Feld nicht auf ungültige Konfigurationen ab. Sie sollten daher sichergehen, dass die jeweilige Einstellung sich für die von Ihnen verwendete Datenbank eignet.

- **Standardnamen verwenden**: Wenn diese Option angehakt ist (Standard), speichert bzw. lädt "WinStudio" die Daten in die Tabelle, wobei der Standardname in das Feld **Name** geschrieben wird.
- **Automatisch anlegen**: Wenn diese Option angehakt ist (Standard), legt "WinStudio" eine Tabelle mit dem Namen, der im Feld **Name** steht, automatisch an. Wenn diese Option nicht aktiviert ist, legt "WinStudio" die Tabelle nicht automatisch an. Daher können Daten in der Datenbank nur abgespeichert werden, wenn bereits in der Datenbank eine Tabelle mit dem im Feld **Name** konfigurierten Namen manuell angelegt wurde.
- **Name**: Benennt den Namen der Tabelle aus der Datenbank, in der die Historien-Daten abgespeichert werden. Um ein Arbeitsblatt in einer Microsoft-Excel-Arbeitsblatt-Datei anzugeben, verwenden Sie folgende Syntax:

  ```
  [arbeitsblattname$]
  ```

- **Aktualisieren**: Wenn die eingestellte Datenbank aktuell verfügbar ist, können Sie die Schaltfläche **Aktualisieren** drücken und so die Combobox **Name** mit dem Namen der derzeit in der Datenbank vorhandenen Tabellen füllen. So können Sie die Tabelle, in der die Historiendaten gespeichert werden sollen, auswählen, anstatt den Tabellennamen manuell in das Feld **Name** einzutragen.

Bereich Laufzeit

In diesem Bereich können Sie Laufzeitwerte festlegen. Folgende Felder stehen zur Verfügung:

- **Status** (Ausgabe): Der Tag in diesem Feld erhält einen der folgenden Werte:

Wert	Beschreibung
0	Von der Datenbank getrennt. Die Datenbank ist nicht verfügbar, ihre Konfiguration ist fehlerhaft oder es handelt sich um eine unzulässige Operation
1	Die Datenbank ist erfolgreich verbunden
2	Die Datenbank wird derzeit synchronisiert

- **Neu laden** (Ausgabe): Geben Sie einen „Neu laden“-Tag an, wenn Sie in einem Konfigurationsfeld geschweifte Klammern verwenden. Wenn Sie sich mit der Datenbank neu verbinden und hierbei die aktualisieren Werte Ihrer Tags verwenden wollen, setzen Sie den Tag auf diesem Feld auf 1. "WinStudio" aktualisiert dann die Konfiguration, wenn es mit der Datenbank eine Aktion ausführen will, und setzt den Tag zurück auf 0, wenn diese abgeschlossen ist.

Weitere Informationen unter:

Konfigurieren einer Standarddatenbank für die gesamte Task-Historie auf Seite 508.

17.6 Konfigurieren einer Standarddatenbank für alle Task-Historien

Sie können eine Standarddatenbank konfigurieren, die die Historie-Daten aller Tasks im Projekt speichert. Nachdem Sie eine Standarddatenbank konfiguriert haben, können Sie immer, wenn Sie ein neues Task-Arbeitsblatt anlegen, wählen, ob Sie die Standarddatenbank verwenden oder für dieses spezielle Arbeitsblatt eine neue Datenbank konfigurieren möchten.

Konfigurieren der Verbindungseinstellungen für die Standarddatenbank:

1. Öffnen Sie den Eigenschaftendialog des Ordners "WinStudio" und wählen Sie das Register "Optionen" an
2. Klicken Sie im Bereich "Standard Datenbank" auf **Konfigurieren**. Der Dialog "Konfiguration der Standarddatenbank" wird angezeigt

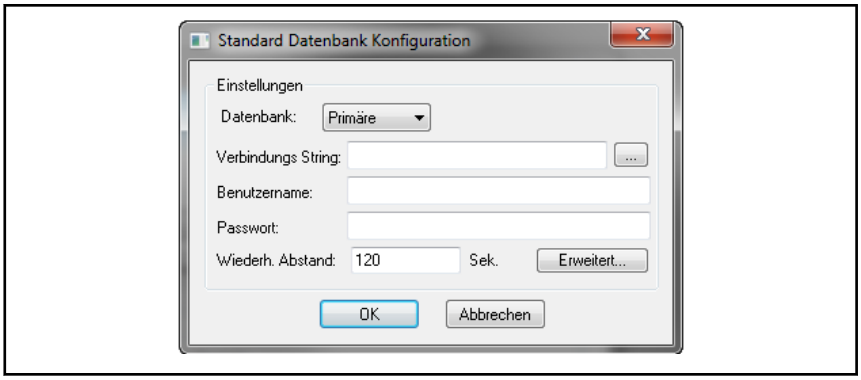


Abb.17-11: Dialog Konfiguration der Standarddatenbank

Hilfestellung zum Ausfüllen der Felder in diesem Fenster finden Sie unter Dialog Datenbankkonfiguration auf Seite 502.

17.7 Fehlerbeseitigung Datenbank

Die "WinStudio"-Datenbankschnittstelle stellt leistungsstarke Werkzeuge für das Auffinden von Konfigurationsproblemen mit Datenbanken zur Verfügung. Wenn Sie Probleme mit der Kommunikation mit einer Datenbank haben, sollten Sie zunächst **Datenbankmeldungen** im Fenster "LogWin" aktivieren. Gehen Sie dazu wie folgt vor:

1. Öffnen Sie das WinStudio-Ausgabefenster. Wählen Sie dazu den Eintrag **Werkzeuge** ▸ **LogWin** im Kontextmenü des Ordners "WinStudio" aus.
2. Führen Sie einen Rechtsklick auf das Fenster "LogWin" aus. Diese öffnet sich standardmäßig am unteren Bereich der Entwicklungsumgebung. Klicken Sie dann im Kontextmenü auf **Einstellungen**:



Abb. 17-12: Ausgabefenster, Einstellungen

3. Aktivieren Sie im Dialog "Logging-Einstellungen" die Option **Datenbankmeldungen**:

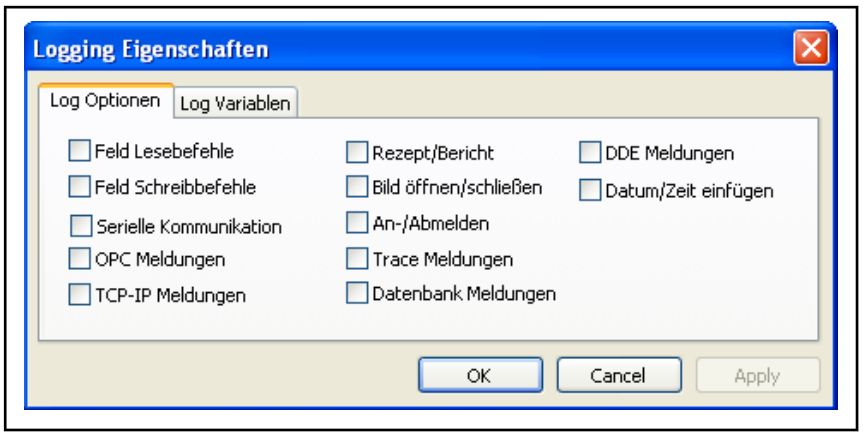


Abb. 17-13: Aktivieren von Datenbankmeldungen

Nach dem Aktivieren dieser Option zeigt das Fenster "LogWin" Fehlermeldungen mit Bezug auf die Datenbank an. Der folgende Abschnitt mit häufig gestellten Fragen führt einige häufig im Fenster "LogWin" angezeigte Fehler auf.

Häufige Fragen

F: Ich habe meine Datenbank konfiguriert, aber die Laufzeitmodule (Alarm, Trend und Ereignisse) werden nicht in der Datenbank gespeichert. Ich kann im Fenster "LogWin" nur die folgende Fehlermeldung sehen:

Datenbank-Fehler: Fehler beim Hinzufügen des neuen Registers[CMD_ADD].

A: Den meisten Datenbankfehlern im Fenster "Ausgabe" sind zusätzliche Informationen nachgestellt, wie z.B. der ausgeführte SQL-Befehl, der Verbindungs-String und der Tabellenname. Fehlermeldungen wie die oben beschriebene treten häufig nach einer detaillierteren Meldung auf. Wenn zum Beispiel Ihr Trend-Task der Datenbank kein Register hinzufügen kann, weil das Kabel nicht angeschlossen ist, sollten Sie zuerst einen Netzwerkfehler erhalten; wenn der Task versucht, vor Ablauf der im Feld **Wiederholen** festgelegten Zeit weitere Register hinzuzufügen (siehe Dialog Datenbankkonfiguration auf Seite 502), wird nur angezeigt **Datenbank-Fehler: Fehler beim Hinzufügen des neuen Registers[CMD_ADD].** Wenn Sie glauben, dass Ihre Konfiguration korrekt ist und Sie die Ursache für dieses Problem eingrenzen wollen, reduzieren Sie den Wert für Wiederholen. Dann sollten detailliertere Informationen angezeigt werden.

F: Wenn ich versuche, auf den MySQL-Datenbankserver zuzugreifen, erhalte ich die folgende Meldung: Object is not set to an instance of an object.

A: Dieses Problem wurde unter den folgenden Bedingungen festgestellt:

- Ein bekannter Programmfehler in MySQL Connector/Net v6.1.2 konnte den Zeichensatz nicht korrekt spezifizieren; und
- die Datenbanktabelle, auf die Sie zugreifen wollen, existiert nicht.

Stellen Sie zur Lösung dieses Problems sicher, dass Sie MySQL Connector/Net v6.2.0 verwenden und dass die Tabelle, auf die Sie zugreifen wollen, in der Datenbank existiert.

F: Warum schließt die Datenbankschnittstelle automatisch einige der Verbindungen?

A: Standardmäßig kann die Datenbankschnittstelle maximal 1000 Verbindungen haben. Wird dieser Maximalwert überschritten, wird die älteste Verbindung automatisch zugunsten der neuen Verbindung geschlossen und das Fenster "LogWin" zeigt eine erweiterte Meldung mit einer Beschreibung, welche Verbindung geschlossen wurde und welcher Befehl zuletzt ausgeführt wurde, an.

Zum Erhöhen der Höchstzahl der Datenbankverbindungen öffnen Sie die Projektdatei (**projekt_name.app**) in einem Texteditor und ändern Sie die folgende Einstellung:

[StDB] MaxConnections="anzahl_der_verbindungen"

Bedenken Sie, dass das Erhöhen der Höchstzahl der Verbindungen die Leistungsfähigkeit des Projekts mindern kann.

F: Ich habe meinen Verbindungs-String mit dem Browser und dem Fenster Eigenschaften Datenverknüpfung konfiguriert. Wenn ich auf die Schaltfläche Test klicke, erhalte ich die Meldung „Test erfolgreich“. Wenn ich aber mein Projekt ausführe, zeigt die Datenbankschnittstelle Fehlermeldungen an und ich kann keine Daten speichern.

A: Das Fenster Eigenschaften Datenverknüpfung verwendet OLE DB als Schnittstelle zur Datenbank. Die "WinStudio"-Datenbankschnittstelle verwendet ADO.NET. So kann es sein, dass Sie den OLE-DB-Anbieter auf Ihrem Rechner haben, aber nicht den ADO.NET-Anbieter. Es kann auch sein, dass Sie einen ADO.NET-Anbieter verwenden, der nicht in der Datei StADOSvr.ini aufgeführt ist. Weitere Informationen zum Hinzufügen eines ADO.NET-Anbieters zur Datei "StADOSvr.ini" finden Sie unter Studio-Datenbank-Gateway auf Seite 496.

F: Warum wird mehr als eine Zeile in meiner Datenbank aktualisiert, wenn ich Daten in einer Zeile in einem Rasterobjekt aktualisiere?

A: Das Rasterobjekt gibt einen Aktualisierungsbefehl in der Datenbank aus unter Verwendung der Werte aus allen Spalten für die betreffende Zeile, die Sie aktualisieren wollen. Dieses Problem kann bei Zeilen mit doppelten Werten auftreten. Wenn Ihre Tabelle einen Primärschlüssel oder andere eindeutige Felder hat, die Sie nicht im "Raster"-Objekt anzeigen wollen, können Sie es zu den **Spalten** hinzufügen, aber die **Breite** 0 angeben. Dies behebt das Problem.

F: Warum muss ich zum Speichern von Millisekunden in meiner Datenbank eine separate **Spalte** verwenden?

A: Einige Datenbanken unterstützen keine Millisekunden im Feld **Zeitstempel**. Die "WinStudio"-Datenbankschnittstelle benötigt standardmäßig eine weitere Spalte für Millisekunden. Wenn Ihre Datenbank Millisekunden verarbeiten kann oder wenn Sie die Millisekunden nicht erfassen wollen, können Sie das Standardverhalten in den Weiteren Einstellungen ändern. Beachten Sie, dass einige Datenbanken Millisekunden speichern können, diese haben aber eine geringere Genauigkeit. Wenn Sie im redundanten Modus verschiedene Datenbanken mit verschiedenen Genauigkeiten mischen, kann es zu Synchronisationsproblemen kommen.

F: Im Emulationsmodus funktioniert mein Projekt hervorragend. Aber wenn ich es an das Windows-Embedded-Gerät sende, kann es nicht mit meiner Datenbank kommunizieren.

A: Es kann sein, dass Ihr Windows-Embedded-Gerät nicht über .Net Framework oder den von Ihnen verwendeten Anbieter verfügt. Versuchen Sie, den Gateway entfernt zu verwenden, gemäß der Anleitungen unter Verknüpfen der Datenbank über einen Remote-Datenbankanbieter auf Seite 495.

F: Warum erhalte ich bei dem Versuch, eine Verbindung zur Datenbank herzustellen, die Meldung "Fehler beim Erstellen der Verbindungsklasse"?

A: Der .Net-Anbieter, den Sie zu verwenden versuchen, ist auf Ihrer Maschine nicht installiert. Dieser Fehlermeldung folgt normalerweise der Anbietername; wenn Sie zum Beispiel die Sybase-Datenbank verwenden, steht hinter der Meldung "[iAnywhere.Data.AsaClient.AsaConnection]". Der Anbieter ist der iAnywhere.Data.AsaClient. Unter "**SystemsteuerungVerwaltungswerkzeuge Konfiguration Microsoft .Net Framework x.x** " können Sie überprüfen, ob der Anbieter auf Ihrer Maschine installiert ist. Der Anbieter sollte im "Assembly Cache" aufgeführt sein.

F: Was ist, wenn ich die Anbieter-Assembly habe (normalerweise eine .dll-Datei), diese aber nicht im "Assembly Cache" aufgeführt ist?

A: Wenn Ihre Assembly keinen starken Namen hat, können Sie diese mit dem Programm „gcautil“ in der "Assembly Cache" registrieren. Es sollte auch funktionieren, wenn Sie Ihre Assembly in den gleichen Ordner wie die "StADOSvr.exe" kopieren (in der Regel der Ordner **[…]\WinStudio\Bin**).

F: Wenn ich eine spezifische Bedingung verwende, kann ich vom Raster aus nicht auf meine Tabelle zugreifen. Wenn keine Bedingung angewendet wird, funktioniert es problemlos. Woher kommt das?

A: Sie sollten die folgenden Punkte überprüfen:

1. Befolgen Sie die Schritte zur Fehlerbeseitigung auf Seite 509 und suchen Sie im Protokoll nach Fehlermeldungen. Aus einer Fehlermeldung können Sie ersehen, ob Sie einen Fehler gemacht haben, z.B. Eingabe mit einem falschen Spaltennamen oder Angabe eines ungültigen Datenformats.
2. Einige Datenbanken haben Probleme, wenn Sie reservierte Begriffe als Spaltennamen verwenden. Daher sollten Sie die Verwendung von Spaltennamen wie Zeit, Datum, Numerisch usw. vermeiden.

3. Wenn Ihr Spaltenname mit AND oder OR (z.B. ORange) beginnt, geben Sie den Namen in eckigen Klammern ein.

   Zum Beispiel

   `[ORange]=10` anstelle von `ORange=10`.

4. Wenn Sie SQL Server CE verwenden, kann es Probleme beim Abfragen von String-Feldern geben. Es wurde festgestellt, dass Filter nicht mit NCHAR-Datentypen funktionieren. Sie funktionieren aber, wenn Sie diese Felder als **NVARCHAR(<Zahl>)** deklarieren.

   Sie können versuchen, Ihre Tabelle unter Verwendung dieses Datentyps neu zu erstellen.

Beispiel für einen Befehl, der eine Tabelle mit Strings zur Abfrage erstellt:

```
CREATE TABLE Table1 (Name NVARCHAR(128), Alter Numerisch, Geschlecht NVARCHAR(1))
```

ORACLE

F: Wenn ich die Verbindung mit einer ORACLE-Datenbank verliere, wird sie nicht neu aufgebaut. In LogWin erhalte ich die folgende Meldung:

"Datenbank: Fehler: ORA-03114: not connected to ORACLE".

Ist dies ein Problem mit der Datenbankschnittstelle von "WinStudio"?

A: Der Oracle-.Net-Anbieter hat ein Problem mit der Verwaltung des Verbindungs-Pools. Sie müssen ein QFE 830173 installieren. Zum Zeitpunkt der Erstellung dieses Dokuments standen auf dieser Site (externer Link) weitere Informationen zu diesem Problem zur Verfügung.

F: Wenn ich versuche, auf die Datenbank zuzugreifen, erhalte ich die folgende Meldung:

ORA-00162: external dbid length 19 is greater than maximum (16).

Was soll ich tun?

A: Zum Zeitpunkt der Erstellung dieses Dokuments gab es ein Problem mit dem Oracle-.NET-Anbieter; der **Servername** (SERVER/TNS) durfte maximal 16 Zeichen lang sein. Um dieses Problem zu beheben, sollten Sie versuchen, Ihr Feld **Servername** zu reduzieren.

Eine Möglichkeit hierzu ist das Bearbeiten der Datei **\WINDOWS\system32\drivers\etc\hosts** durch Hinzufügen eines Eintrags mit einem kürzeren Servernamen.

Zum Beispiel hat der durch **192.168.89.98** bezeichnete Servername 13 Zeichen. Durch Hinzufügen der folgenden Zeile in der Datei könnte er auf 3 reduziert werden:

192.168.89.98 ora

Jetzt können Sie den **Servernamen** mit ORA/TNS anstelle von **192.168.89.98/TNS** konfigurieren.

MYSQL

F: Wenn ich versuche, von meinem lokalen Rechner auf die Datenbank zuzugreifen, funktioniert dies gut. Wenn ich aber mein Projekt auf ein Remote-Gerät verschiebe, erhalte ich die Meldung **Zugriff verweigert**.

A: Mit jedem Benutzer in einer MySQL-Datenbank wird eine Eigenschaft verbunden, die den Rechner angibt, von dem er Zugriff zur Datenbank erhalten kann. Standardmäßig ist diese Eigenschaft auf "localhost" eingestellt, damit Sie nur auf die Datenbank zugreifen können, wenn sie vom lokalen Rechner aus zugreifen. Weitere Informationen über das Ändern dieser Einstellung finden Sie im MySQL-Handbuch.

F: Manchmal wird bei dem Versuch, eine Remote-MySQL-Datenbank mit einer lokalen MySQL-Datenbank zu synchronisieren oder Anwendungsredun-

danz zu verwenden, eine Verbindung mit der ADO.NET-Schnittstelle geöffnet und niemals geschlossen.

A: Gehen Sie zum Dialog Datenbankkonfiguration auf Seite 502 und deaktivieren Sie die Option **Automatisch erstellen**.

SYBASE

F: Ich habe meine Sybase-Datenbank mit der Schaltfläche Durchsuchen konfiguriert. Wenn ich auf die Test-Schaltfläche klicke, ist der Test erfolgreich. Wenn ich aber mein Projekt ausführen will, erhalte ich den folgenden Fehler:

Datenbank: Fehler: Parsing-Fehler: DSN 'MyDatabase' existiert nicht.

Was mache ich falsch?

A: Mehr Informationen über dieses Problem finden Sie in Datenbank Anhang F – Verwendung von Sybase auf Seite 525.

F: Warum erhalte ich bei dem Versuch, eine Verbindung zur Sybase-Datenbank aufzubauen, den Fehler **Fehler bei Erstellen der Verbindungsklasse [iAnywhere.Data.AsaClient.AsaConnection]**?

A: Der **ADO.NET**-Anbieter ist nicht auf Ihrem Rechner installiert. Das Setup-Programm der Datenbank verfügt über eine Option zum Installieren des Anbieters. Führen Sie das Setup-Programm erneut aus und stellen Sie sicher, dass diese Option angewählt wird.

SQL SERVER CE

F: Warum zeigt der Gateway **TypeLoad Fehler** an, wenn ich versuche, auf meine SQL-Server-CE-Datenbank zuzugreifen?

A: Dieses Problem tritt normalerweise auf, wenn der SQL-Server-CE.NET-Anbieter nicht auf ihrem CE-Gerät installiert ist.

F: Warum erhalte ich die Fehlermeldung **Störung im gemeinsamen Dateizugriff. Der Prozess wird evtl. von einer anderen Datei verwendet**?

A: Sie haben ein anderes Programm mit der SQL-Server-CE-Datenbank geöffnet. Dies passiert zum Beispiel bei Verwendung der Konfigurationssoftware für SQL Server CE.

17.8 Anhänge

17.8.1 Verwendung von ODBC-Datenbanken

Fast jede Datenbank stellt eine ODBC-Schnittstelle zur Verfügung, die für die Kommunikation mit ihr verwendet werden kann. Die von "WinStudio" zur Verfügung gestellten Datenbankfunktionen können über die ADO.NET-Schnittstelle für ODBC mit ODBC-Treibern verwendet werden. Um diese Möglichkeit nutzen zu können, müssen Sie Microsoft .NET Framework 1.1 oder höher verwenden.

Microsoft .NET Framework 2.0 wird automatisch mit "Rexroth WinStudio v7.2" oder höher installiert.

Mit dem Dialog Datenbankkonfiguration auf Seite 502 können Verbindungs-Strings für die Verbindung mit einem ODBC DSN zur Verfügung gestellt werden. Der Verbindungs-String kann durch Klicken auf die Schaltfläche Durchsuchen (**...**) automatisch erstellt werden. Wenn das Fenster "Datenverknüpfung" angezeigt wird, sollten Sie wie unten gezeigt die Option Microsoft OLE-DB-Anbieter für ODBC-Treiber auswählen:

Datenbankschnittstelle

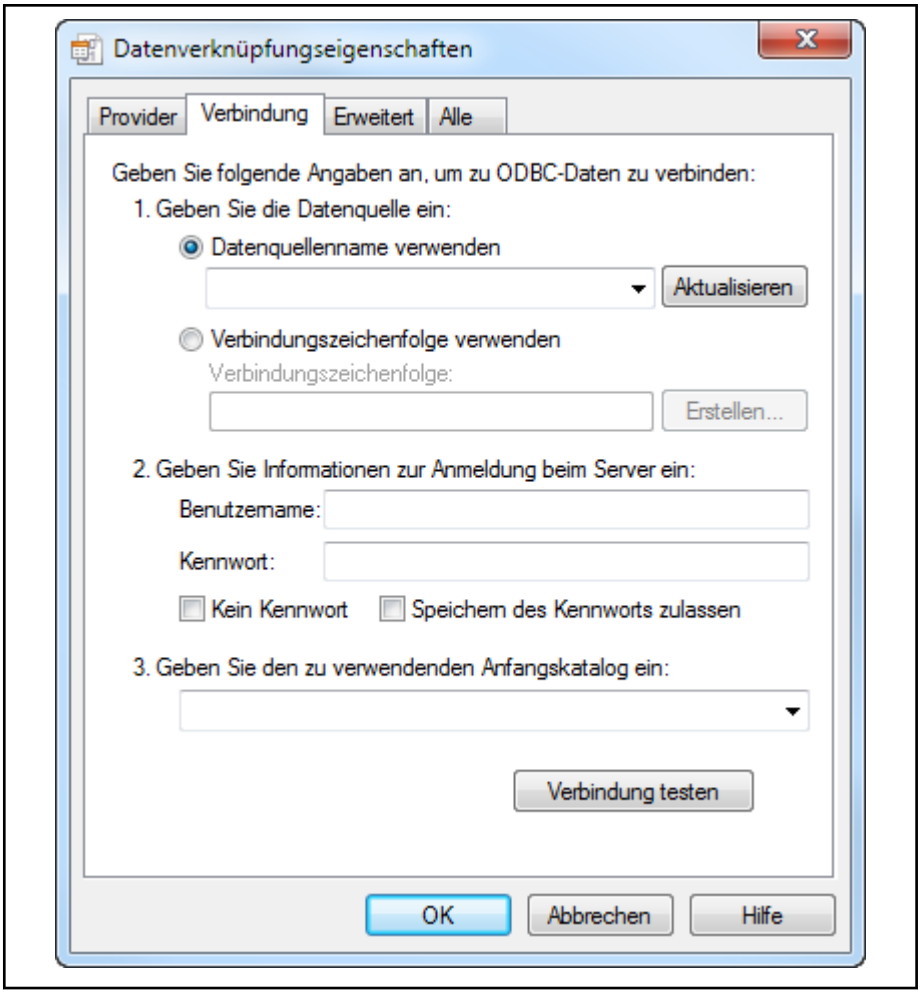


Abb.17-14: Eigenschaften Datenverknüpfung, Anbieter – ODBC

Durch Klicken auf **Weiter** wird das folgende Fenster angezeigt:

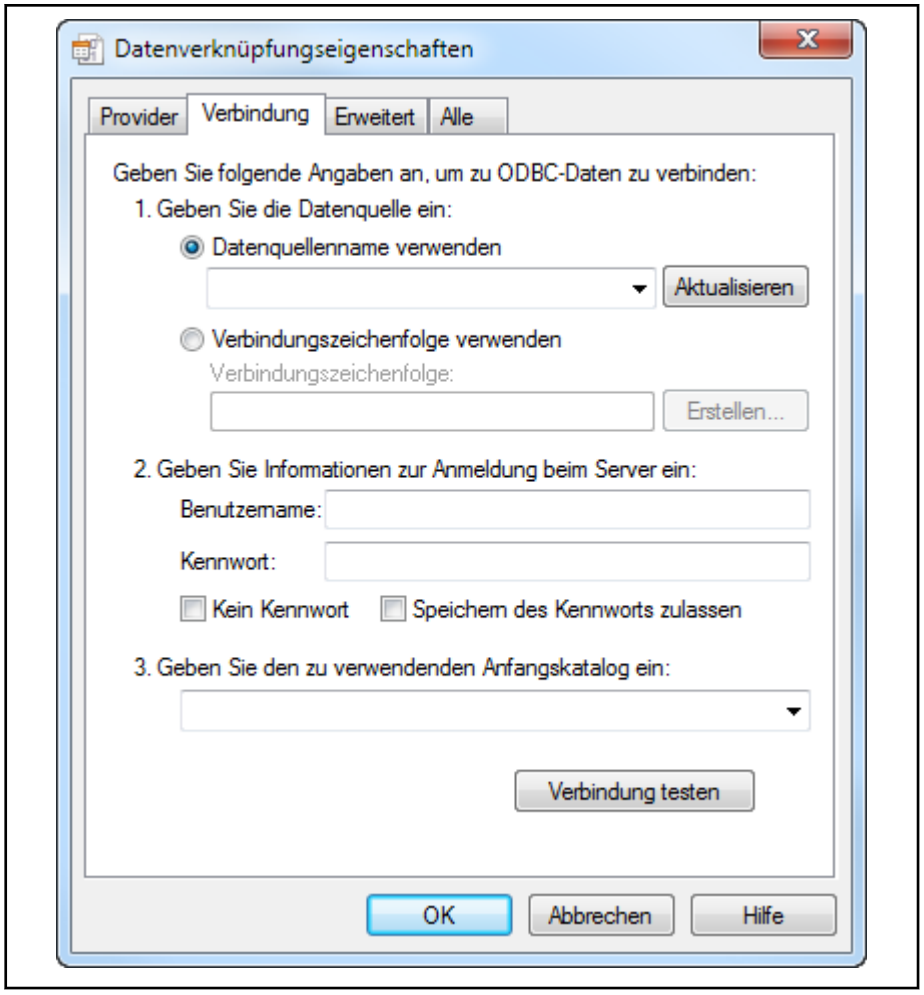


Abb.17-15: Eigenschaften Datenverknüpfung, Anbieter – Verbindung

Wählen Sie den DSN aus, zu dem Sie eine Verbindung herstellen wollen, und klicken Sie auf **OK**. Wenn Sie Benutzername und Passwort in diesem Fenster und nicht im Dialog "Objekteigenschaften" festlegen wollen, muss das Kontrollkästchen **Passwort speichern zulassen** aktiviert werden.

17.8.2 Verwendung von Microsoft SQL Server

Mit der Datenbankschnittstelle von "WinStudio" können Sie Daten von relationalen Microsoft-SQL-Server-Datenbanken abrufen bzw. Daten dort speichern. Zum Konfigurieren der SQL-Server-Datenbank sollten Sie die nachfolgenden Schritte in der angegebenen Reihenfolge ausführen:

1. Klicken Sie im Dialogfenster Datenbankkonfiguration auf Seite 502 auf die Schaltfläche **Suchen**. Das folgende Fenster wird angezeigt:

Datenbankschnittstelle

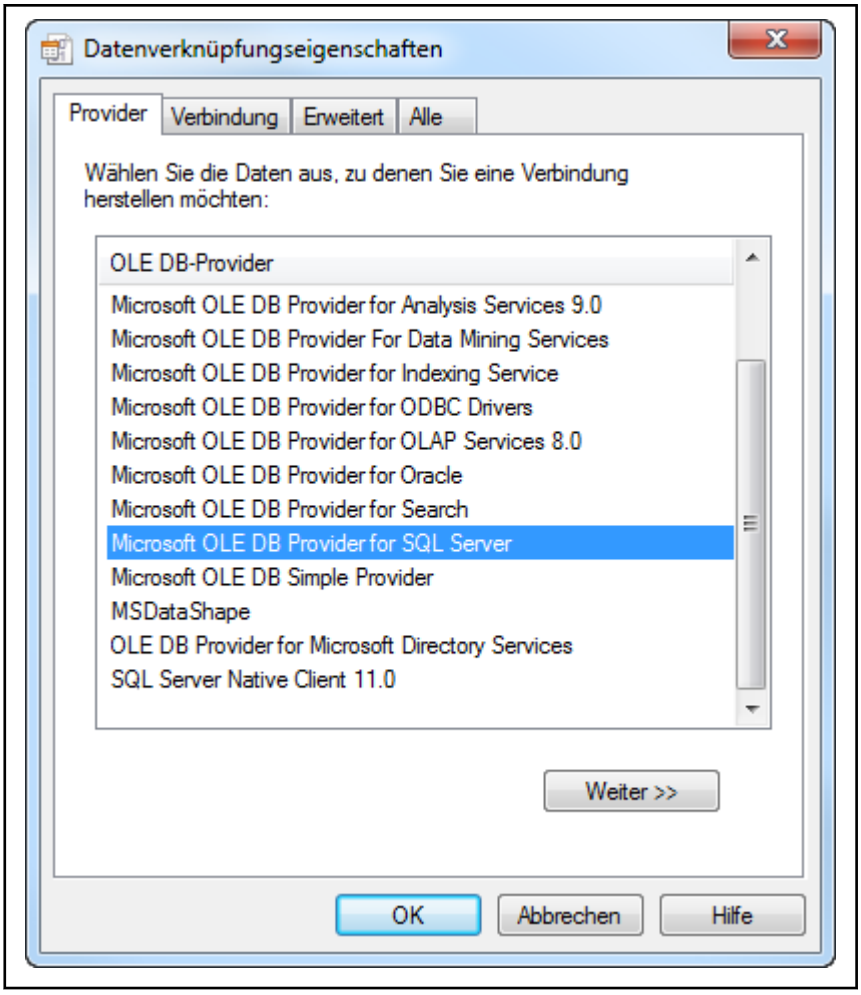


Abb.17-16: Eigenschaften Datenverknüpfung, Anbieter – SQL Server

2. Wählen Sie den Microsoft OLE Anbieter für SQL Server aus und klicken Sie auf **Weiter**. Das folgende Fenster wird angezeigt:

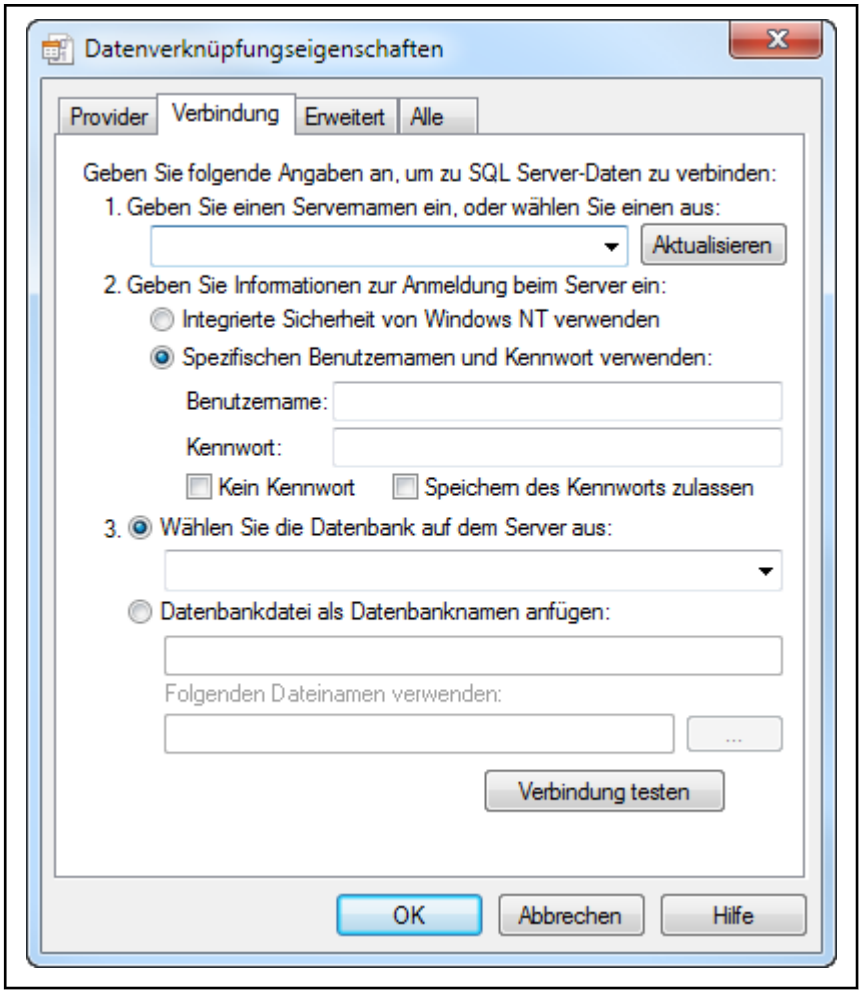


Abb.17-17: Eigenschaften Datenverknüpfung, Verbindung – SQL Server

3. Geben Sie in den Feldern in diesem Fenster Ihre Datenbankinformationen ein. Wenn Sie nicht die Windows NT Integrated Security verwenden, achten Sie darauf, das Kontrollkästchen **Passwort speichern zulassen** beim Schließen des Fensters "Eigenschaften Datenverknüpfung" zu aktivieren.
4. Zum Beenden der Konfiguration des "Verbindungsstrings" klicken Sie auf **OK**.

Ihr Verbindungs-String sollte ähnlich wie dieser aussehen:

Provider=SQLOLEDB.1; Integrated Security=SSPI; Initial Catalog=MyDatabase;

Data Source=192.168.23.200

☞ Diese Vorgehensweisen wurden mit Microsoft SQL Server 2000 getestet.

17.8.3 Verwendung von ORACLE-Datenbanken

Mit der Datenbankschnittstelle von "WinStudio" können Sie Daten von relationalen ORACLE-Datenbanken abrufen bzw. Daten dort speichern.

Zum Konfigurieren der ORACLE-Datenbank sollten Sie die nachfolgenden Schritte in der angegebenen Reihenfolge ausführen:

Datenbankschnittstelle

1. Klicken Sie im Dialogfenster Datenbankkonfiguration auf Seite 502 auf die Schaltfläche **Suchen**. Das folgende Fenster wird angezeigt:

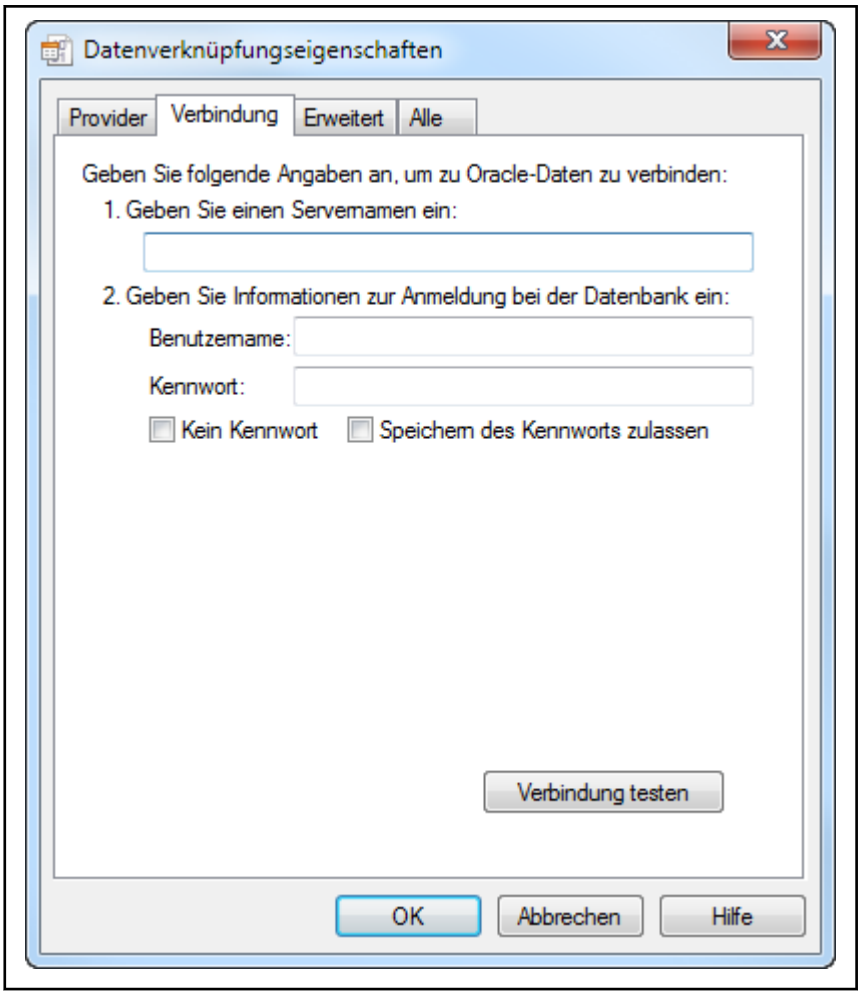


Abb. 17-18: Eigenschaften Datenverknüpfung, Anbieter – Oracle

2. Wählen Sie den Microsoft OLE Anbieter für Oracle aus und klicken Sie auf **Weiter**. Das folgende Fenster wird angezeigt:

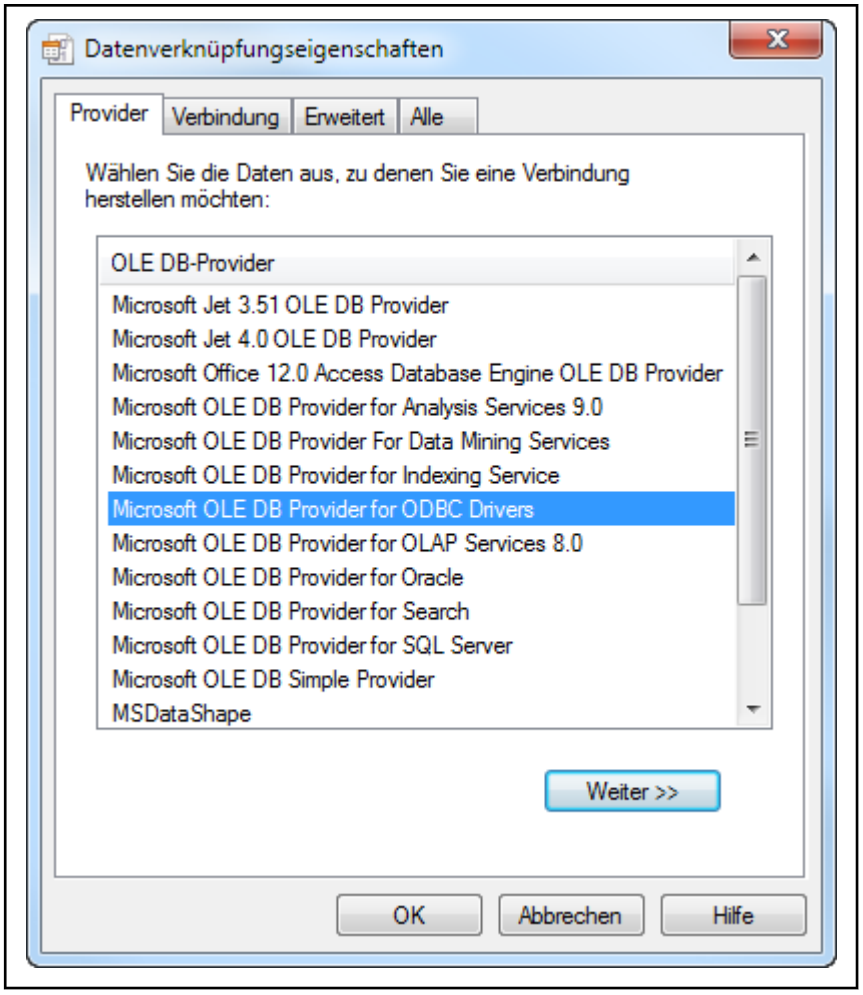


Abb.17-19: Eigenschaften Datenverknüpfung, Verbindung – Oracle

3. Geben Sie in den Feldern in diesem Fenster Ihre Datenbankinformationen ein. Vergessen Sie nicht, das Kontrollkästchen **Passwort speichern zulassen** zu aktivieren, um das Passwort beim Schließen des Fensters "Eigenschaften Datenverknüpfung" zu speichern. Die Information Servername hat das folgende Format:

   Server / TNS

   Wobei gilt:

   - **Server**: Rechner, auf dem Oracle Database läuft
   - **TNS**: Oracle TNS Name

HINWEIS	**Zum Zeitpunkt der Erstellung dieses Dokuments konnte das Feld Servername mit maximal 16 Buchstaben konfiguriert werden.**

Wenn mehr als 16 Buchstaben angegeben werden, erhalten Sie den folgenden Fehler:

ORA-00162: external dbid length 19 is greater than maximum (16),

wobei 19 die Anzahl der Buchstaben in **Servername** ist.

Weitere Tipps zur Umgehung dieses Problems finden Sie unter Fehlerbeseitigung Datenbank auf Seite 509.

4. Zum Beenden der Konfiguration des "Verbindungsstrings" klicken Sie auf **OK**.

☞ Diese Vorgehensweisen wurden mit ORACLE 10g Release 1 getestet.

17.8.4 Verwendung von Microsoft Access Datenbanken

Mit der "WinStudio"-Datenbankschnittstelle können Informationen in Microsoft-Access-Datenbankdateien (ACCDB) gespeichert bzw. von dort abgerufen werden.

Access muss nicht auf dem gleichen Rechner wie "WinStudio" installiert sein. "WinStudio" kann direkt aus einer bestehenden ACCDB-Datei lesen bzw. dorthin schreiben. Allerdings muss Access zum Erstellen der Ausgangsdatei verwendet werden – entweder eine leere Datenbank für neue Daten oder eine gefüllte Datenbank für Referenzdaten. Nach dem Erstellen der Datei können sie diese an eine beliebige Stelle verschieben und eine Verbindung dorthin herstellen.

☞ Diese Vorgehensweise wurde zuletzt mit Microsoft Access 2007 (12.0.6211.1000) getestet.

Zum Erstellen einer Verbindung zwischen Ihrem "WinStudio"-Projekt und Ihrer ACCDB-Datei:

1. Klicken Sie im Dialog Datenbankkonfiguration auf Seite 502 auf die Schaltfläche Suchen (**...**). Das Eigenschaftsblatt "Eigenschaften Datenverknüpfung" wird angezeigt:

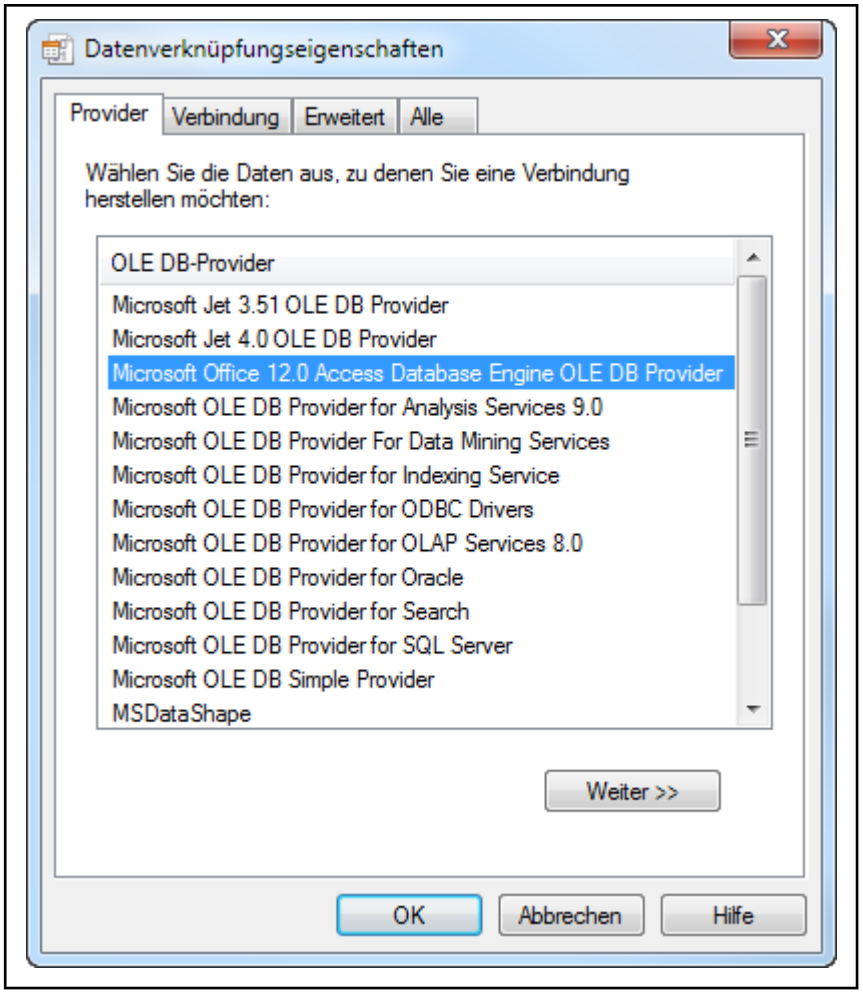


Abb.17-20: Auswahl des OLE Datenbankanbieters

2. Wählen Sie den passenden OLE-DB-Anbieter für Ihre Datenbank aus:
   - Für eine Microsoft-Access-2003-Datenbank-Datei wählen Sie **Microsoft Jet 4.0** aus.
   - Für eine Microsoft-Access-2007-Datenbank-Datei wählen Sie **Microsoft Office 12.0 Access Database Engine** aus.
3. Klicken Sie auf **Weiter**.

   Die Registerkarte "Verbindung" des Eigenschaftsblatts wird angezeigt:

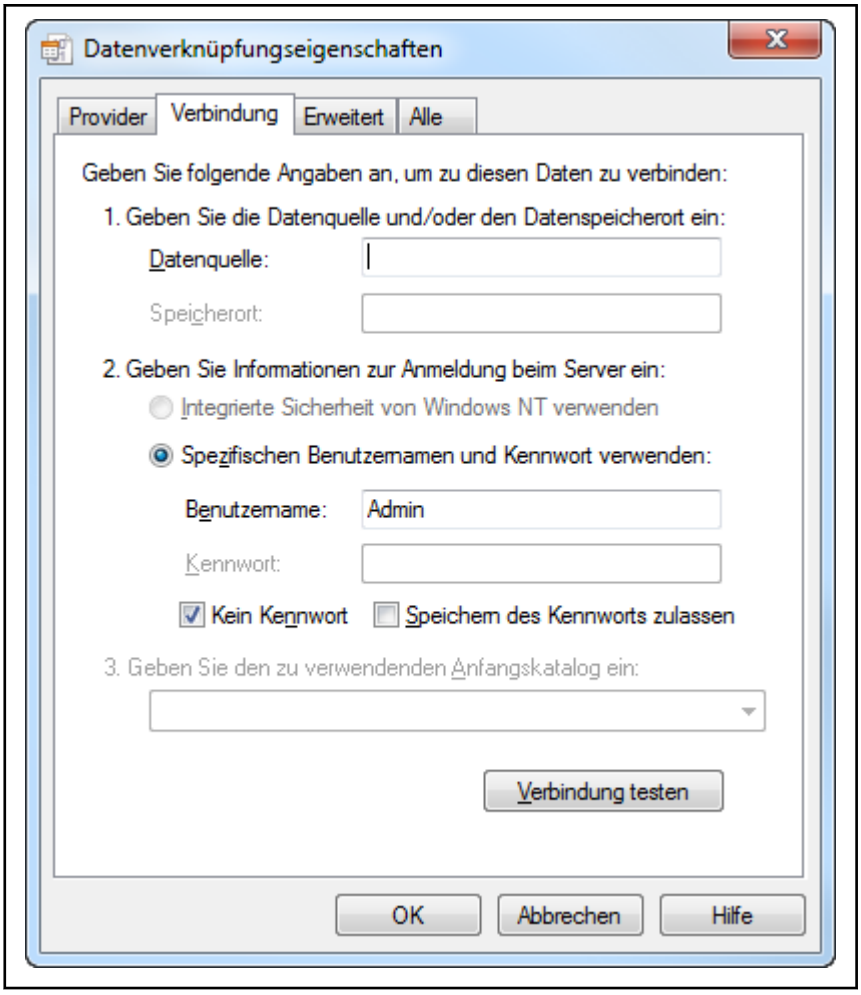


Abb.17-21: Speicherort der Datei festlegen

4. Geben Sie im Textfeld **Datenquelle** den vollständigen Dateipfad für Ihre ACCDB-Datei ein. Obwohl sich die Datei an einer beliebigen Stelle auf Ihrem Rechner bzw. in Ihrem Netzwerk befinden kann, ist es hilfreich, sie in Ihrem Projektordner zu behalten. Zum Beispiel:

   C:\Users\benutzername\Eigene Dokumente\WinStudio-Projekte\projekt_name\datenbank1.accdb

   Wenn Sie dies tun, müssen Sie aber bei jedem Verschieben des Projektordners den Dateipfad aktualisieren.
5. Geben Sie ggf. **Benutzername** und **Passwort** für die ACCDB-Datei ein.
6. Klicken Sie auf **Testverbindung**.

   Wenn eine Verbindung erfolgreich aufgebaut wurde, wird eine entsprechende Meldung angezeigt.
7. Klicken Sie zum Speichern Ihrer Änderungen auf **OK** und kehren Sie zum Eigenschaftsblatt "Datenbankkonfiguration" zurück.

☞ Stellen Sie sicher, im Dialog "Datenbankkonfiguration" die Option **Primärschlüssel deaktivieren** zu aktivieren (auszuwählen). Wenn nicht, kann "WinStudio" sich nicht mit Ihrer Access-Datenbank verbinden.

☞ Desktop-Office-Anwendungen wie z.B. Microsoft Access und Microsoft Excel können große Datenmengen nicht effizient verarbeiten. Wenn Sie versuchen, alle historischen Daten Ihres Projekts in einer Access-Datenbank oder einer Excel-Tabellenkalkulation zu speichern, werden die Abfragen langsamer und Sie erhalten unter Umständen unerwartete Ergebnisse. Daher empfehlen wir, Access oder Excel nur als sekundäre Datenbank mit aktivierter Option **Speichern und Weiter** zu verwenden oder Ihre Daten mit einer Fremdsoftware zu verwalten.

Zur Handhabung großer Mengen historischer Daten empfehlen wir, entweder das proprietäre Format von "WinStudio" oder eine dedizierte relationale Datenbank wie z.B. Microsoft SQL Server oder ORACLE zu verwenden.

17.8.5 Verwendung von SQL Server CE

Verwendung von SQL Server CE, Allgemeines

"WinStudio"-Projekte mit CEView können durch Verwendung von OLEDB Provider for SQL Server CE mit Microsoft-SQL-Server-CE-Datenbanken (SQL CE) auf dem gleichen Gerät kommunizieren. Dieser Anbieter muss auf Ihrem Windows-Embedded-Gerät installiert werden.

Da der Anbieter nur auf Ihrem Gerät und nicht auf Ihrer Entwicklungsarbeitsstation installiert ist, können Sie diesen bei der Entwicklung Ihres Projekts im Dialog Eigenschaften Datenverknüpfung auf Seite 502 nicht auswählen. Statt dessen muss der Verbindungs-String im folgenden Format manuell eingegeben werden:

Version	Verbindungs-String
SQL CE 2,0	Provider=SQLCE; Data Source=database_path
SQL CE 3,0	Provider=SQL_CE3.0; Data Source=database_path
SQL CE 3.5	Provider=SQL_CE3.5; Data Source=database_path

☞ Die Datenbankschnittstelle wurde mit diesen Versionen von SQL CE vollständig getestet. Andere Versionen könnten aber auch unterstützt werden; bitte erfragen Sie nähere Informationen bei Ihrem Kundendienst.

Beispiele:

- Zugriff auf eine Datenbankdatei an einem festen Speicherort, z.B. **\Harddisk\MyDatabase.sdf**:

  Provider=SQLCE; Data Source=\Harddisk\MyDatabase.sdf

  Der genaue Pfad hängt davon ab, wie der permanente Speicher Ihres Gerätes organisiert ist. Bitte ziehen Sie die Herstellerdokumentation zu Rate.

- Zugriff auf eine Datenbankdatei, deren Speicherort durch den Wert des String-Tags "DatabaseFile" festgelegt wird:

  Provider=SQLCE; Data Source={DatabaseFile}

  Die geschweiften Klammern ("{}") zeigen an, dass es sich um eine Tag-Referenz handelt.

Bitte bedenken Sie, dass SQL CE nur für schlanke Datenbanken und einfache Transaktionen bestimmt ist. Es kann große Datenmengen wie die Alarm-

und Ereignishistorien nicht effizient handhaben. In diesen Fällen empfehlen wir, entweder das proprietäre Historienformat oder eine leistungsstärkere relationale Datenbank wie die Vollversion von Microsoft SQL Server zu verwenden. Wenn Sie SQL CE verwenden "müssen", empfehlen wir die Nutzung nur als sekundäre Datenbank unter Verwendung der Option **Speichern und Weiter**. Weitere Informationen finden Sie unter Speichern von Alarm-Historie/ Ereignisprotokoll in einer externen Datenbank auf Seite 281.

HINWEIS

Bedenken Sie beim Verwenden eines Datenbankarbeitsblatts auf Seite 371 oder der DB/ ERP-Funktionen auf Seite 545 für den Zugriff auf die SQL-CE-Datenbank ...

- Spalten-/Feldnamen müssen genau passen, sonst schlagen die Datenbankbefehle fehl; und
- WinStudio-Datentypen auf Seite 124

  werden in möglicherweise unerwartete SQL-Datentypen konvertiert.

Die folgende Tabelle zeigt die Konvertierung:

WinStudio	SQL CE
Boolean	int
Integer	int
Real	real
String	nvarchar

Installieren von SQL Server auf einem Windows-Embedded-Gerät

Microsoft SQL Server wird von den Herstellern der meisten Windows-Embedded- und Windows-Mobile-Geräte nicht mitgeliefert. Wenn Ihr "WinStudio"-Projekt also SQL Server erfordert, müssen Sie diesen ggf. selbst herunterladen und auf dem Gerät installieren.

Bevor Sie fortfahren, überprüfen Sie bitte anhand der Herstellerdokumentation, ob Microsoft SQL Server in dem Gerät enthalten ist. Wenn ja, muss die Software nicht erneut installiert werden, sofern keine Probleme auftreten.

Wenn nicht, überprüfen Sie zumindest, ob das Gerät die Systemanforderungen auf Seite 39 für das Ausführen eines "WinStudio"-Projekts erfüllt und notieren Sie sich den Geräteprozessor.

Da nur der Gerätehersteller Software mit dem Plattform Builder „permanent“ auf dem Gerät installieren kann, kopieren Sie tatsächlich die CAB-Dateien in den permanenten Speicher des Gerätes und konfigurieren dann das Gerät zum Laden dieser Dateien bei jedem Hochfahren.

Zum Zeitpunkt der Erstellung dieser Dokumentation ist die letzte stabile Version von SQL Server **Microsoft SQL Server Compact Edition 3.5 Service Pack 2**.

1. Laden Sie auf Ihrem Desktop die Microsoft SQL Server Compact Edition von der folgenden Website herunter: http://www.microsoft.com/sqlserver/en/us/editions/compact.aspx
2. Extrahieren und installieren Sie die Software auf Ihrem Desktop. Es wird ein neues Verzeichnis mit den Microsoft-SQL-Server-Dateien erstellt,

einschließlich für verschiedene Geräteprozessoren angepasster Dateien.

3. Öffnen Sie das Verzeichnis: **C:\Program Files\Microsoft SQL Server Compact Edition\v3.5\Devices\wce500**
4. Wählen Sie in dem Verzeichnis das korrekte Unterverzeichnis für Ihren Geräteprozessor aus.

   Wenn Ihr Gerät zum Beispiel einen x86-Prozessor verwendet, wählen Sie das Unterverzeichnis "x86".
5. Wählen Sie im Unterverzeichnis die CAB-Dateien aus, die auf Ihr Gerät kopiert werden müssen. Für das Beispiel x86 wählen Sie die folgenden Dateien:
   - `sqlce.dev.ENU.wce5.x86.CAB`
   - `sqlce.repl.wce5.x86.CAB`
   - `sqlce.wce5.x86.CAB`
6. Kopieren Sie die CAB-Dateien in den permanenten Speicher Ihres Gerätes und notieren Sie sich den Speicherort. Die genaue Vorgehensweise zum Kopieren von Dateien in das Gerät hängt von Ihrer Installation ab. Evtl. können Sie Desktop-Synchronisation, eine Netzwerkverbindung oder einen USB-Stick verwenden. Die verfügbaren Vorgehensweisen finden Sie in der Herstellerdokumentation.
7. Erstellen Sie auf Ihrem Desktop eine neue Textdatei mit dem Namen **install.bat**.
8. Bearbeiten Sie **install.bat** zum Erstellen eines Batch-Skripts, das die CAB-Dateien beim Hochfahren lädt.

   Für das Beispiel x86 fügen Sie den folgenden Code in die Datei ein:

   wceload /noui /delete 0 "dateipfad\sqlce.dev.ENU.wce5.x86.CAB"

   wceload /noui /delete 0 "dateipfad\sqlce.repl.wce5.x86.CAB"

   wceload /noui /delete 0 "dateipfad\sqlce.wce5.x86.CAB"

   ...wobei **dateipfad** der Speicherort der kopierten CAB-Dateien ist.

   Weitere Informationen im MSDN-Artikel über das Werkzeug Wceload.
9. Kopieren Sie **install.bat** in das Verzeichnis **\Windows\Startup**. Wenn das Gerät kein Verzeichnis **\Windows\Startup** hat, befolgen Sie die Anweisungen der Herstellerdokumentation für die Konfiguration von Startup-Elementen.
10. Führen Sie einen Neustart des Gerätes aus. Wenn die Dateien korrekt ausgewählt und installiert wurden, startet der SQL-Server beim Hochfahren und Sie sollten mit Ihrem "WinStudio"-Projekt darauf zugreifen können.

17.8.6 Verwendung von Sybase

Sie müssen den AsaClient-Anbieter auf Ihrem Rechner installieren. Die Tests mit "WinStudio" wurden mit der im Abschnitt Verknüpfen der Datenbank über einen Remote-Datenbankanbieter auf Seite 495 erläuterten Architektur durchgeführt.

Wenn Sie die Schaltfläche "Durchsuchen" zum automatischen Erzeugen des Verbindungs-Strings verwenden, hat der resultierende String das folgende Format:

Provider=ASAProv.90; Data Source=Test

Für dieses Formats müssen Sie eine ODBC DSN mit dem gleichen Namen wie die **Datenquelle** erstellen (in diesem Fall "Test"), um mit der Datenbank kommunizieren zu können. Wenn die DSN nicht erstellt wird, wird beim Verbinden mit der Datenbank der folgende Fehler im LogWin angezeigt:

Datenbank: Fehler: Parsing-Fehler: DSN 'Test' existiert nicht.

Um einen ODBC DSN-Fehler zu vermeiden, können Sie wie im nachfolgenden Beispiel gezeigt manuell mit dem Verbindungs-String zugreifen:

Provider=ASAProv.90; DBF=C:\ Test.db

☞ Diese Vorgehensweisen wurden mit Sybase Server Anywhere 9.0.1.1751 getestet.

17.8.7 Verwendung von Microsoft Excel

Mit der "WinStudio"-Datenbankschnittstelle können Informationen in Microsoft-Excel-Tabellenkalkulationen (ACCDB) gespeichert bzw. von dort abgerufen werden.

Excel muss nicht auf dem gleichen Rechner wie "WinStudio" installiert sein. "WinStudio" kann direkt aus einer bestehenden XLS/XLSX-Datei lesen bzw. dorthin schreiben. Allerdings muss Excel und zum Erstellen der Ausgangsdatei verwendet werden – entweder eine leere Tabelle für neue Daten oder eine gefüllte Tabelle für Referenzdaten. Nach dem Erstellen der Datei können sie diese an eine beliebige Stelle verschieben und eine Verbindung dorthin herstellen.

☞ Diese Vorgehensweise wurde zuletzt mit Microsoft Excel 2007 (12.0.6331.5000) getestet.

Zum Erstellen einer Verbindung zwischen Ihrem "WinStudio"-Projekt und Ihrer XLS/XLSX-Datei:

1. Klicken Sie im Dialog Datenbankkonfiguration auf Seite 502 auf die Schaltfläche Suchen (**...**). Das Eigenschaftsblatt "Eigenschaften Datenverknüpfung" wird angezeigt:

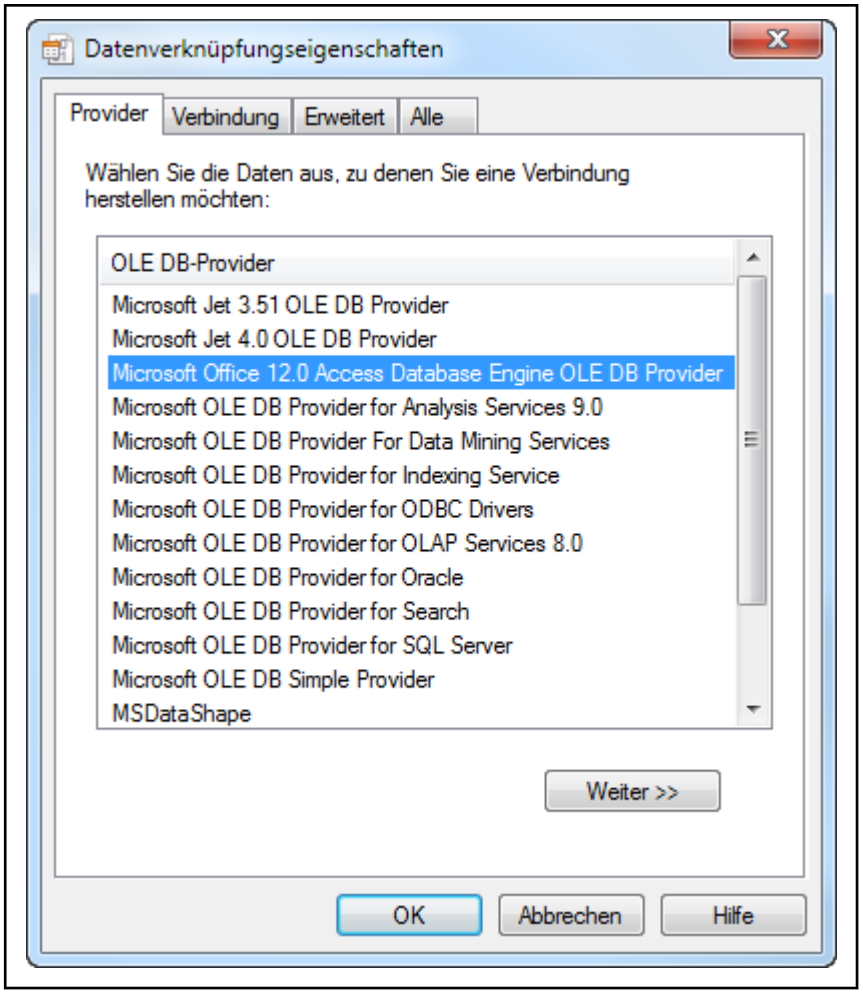


Abb.17-22: Auswahl des OLE Datenbankanbieters

2. Wählen Sie den passenden OLE-DB-Anbieter für Ihre Datenbank aus:
   - Für eine Microsoft-Excel-2003-Tabellenkalkulationsdatei wählen Sie **Microsoft Jet 4.0** aus.
   - Für eine Microsoft-Access-2007-Tabellenkalkulationsdatei wählen Sie **Microsoft Office 12.0 Access Database Engine** aus.
3. Klicken Sie auf **Weiter**.

   Die Registerkarte "Verbindung" des Eigenschaftsblatts wird angezeigt:

Datenbankschnittstelle

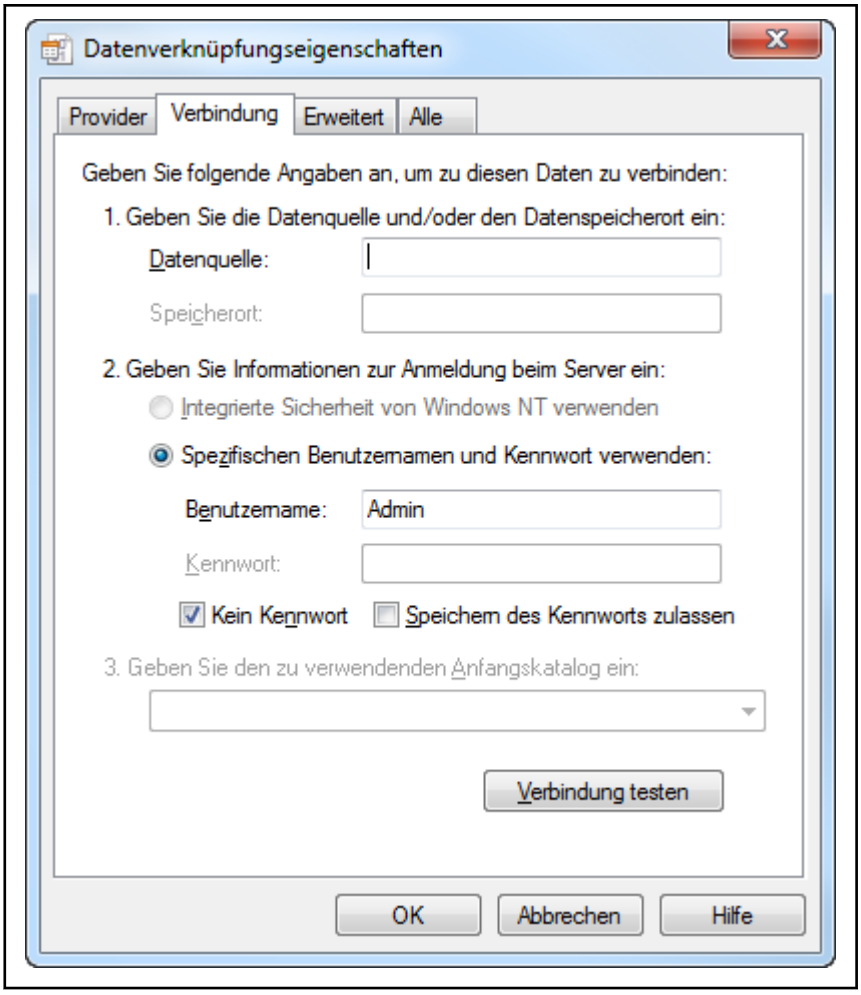


Abb.17-23: Speicherort der Datei festlegen

4. Geben Sie im Textfeld **Datenquelle** den vollständigen Dateipfad für Ihre XLS/XLSX-Datei ein. Obwohl sich die Datei an einer beliebigen Stelle auf Ihrem Rechner bzw. in Ihrem Netzwerk befinden kann, ist es hilfreich, sie in Ihrem Projektordner zu behalten. Zum Beispiel:

   C:\Users\benutzername\Eigene Dokumente\WinStudio-Projekte\projekt_name\arbeitsmappe1.xls

   Wenn Sie dies tun, müssen Sie aber bei jedem Verschieben des Projektordners den Dateipfad aktualisieren.
5. Klicken Sie auf die Registerkarte **Alle**.
6. Wählen Sie **Erweiterte Eigenschaften** und klicken Sie dann auf **Wert bearbeiten**. Der Dialog "Eigenschaftswert bearbeiten" wird angezeigt.
7. Geben Sie im Feld **Eigenschaftswert** einen der folgenden Werte ein:
   - Für eine Microsoft-Excel-2003-Tabellenkalkulationsdatei geben Sie "Excel 11.0" ein.
   - Für eine Microsoft-Excel-2007-Tabellenkalkulationsdatei geben Sie "Excel 12.0" ein.
8. Schließen Sie den Dialog "Eigenschaftswert bearbeiten" durch Klicken auf **OK**.

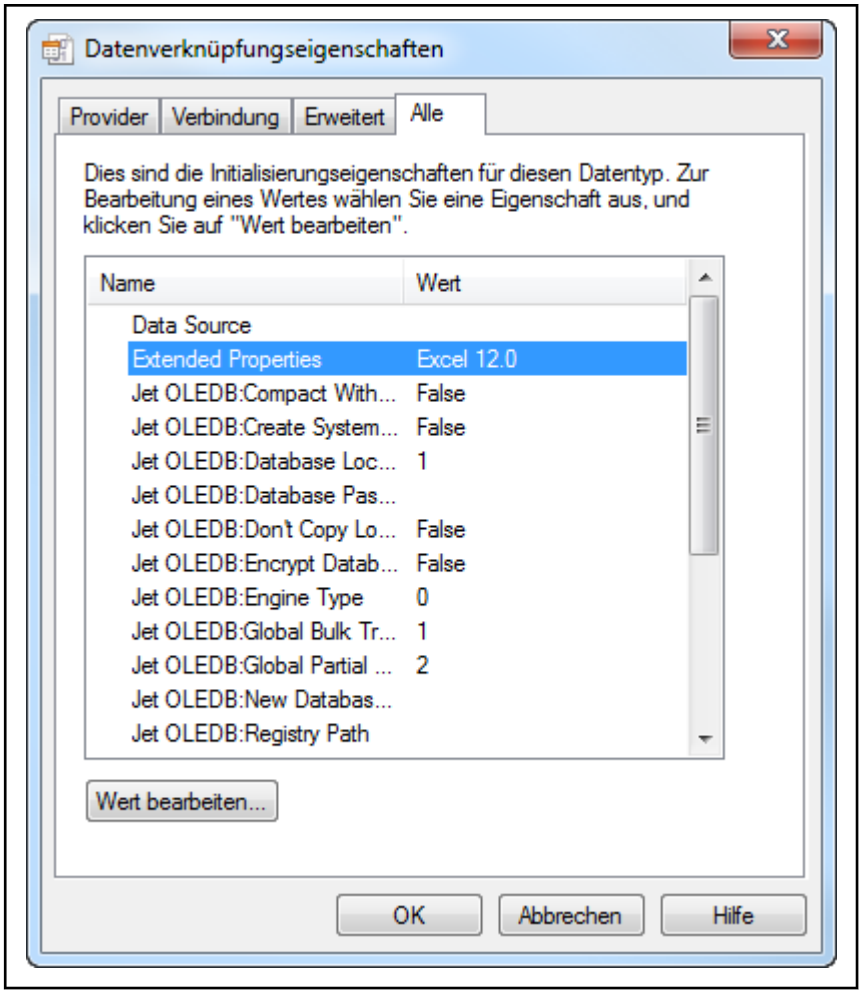


Abb.17-24: Bearbeiten des Werts für Erweiterte Eigenschaften

9. Klicken Sie auf die Registerkarte **Verbindung**.
10. Klicken Sie auf **Testverbindung**. Wenn eine Verbindung erfolgreich aufgebaut wurde, wird eine entsprechende Meldung angezeigt.
11. Klicken Sie zum Speichern Ihrer Änderungen auf **OK** und kehren Sie zum Dialog "Datenbankkonfiguration" zurück.

☞ Stellen Sie sicher, im Dialog "Datenbankkonfiguration" die Option **Primärschlüssel deaktivieren** zu aktivieren (auszuwählen). Wenn nicht, kann "WinStudio" sich nicht mit Ihrer Excel-Tabelle verbinden.

☞ Desktop-Office-Anwendungen wie z.B. Microsoft Access und Microsoft Excel können große Datenmengen nicht effizient verarbeiten. Wenn Sie versuchen, alle historischen Daten Ihres Projekts in einer Access-Datenbank oder einer Excel-Tabellenkalkulation zu speichern, werden die Abfragen langsamer und Sie erhalten unter Umständen unerwartete Ergebnisse. Daher empfehlen wir, Access oder Excel nur als sekundäre Datenbank zu verwenden, mit aktivierter Option **Speichern und Weiter**, oder Daten an eine Fremdsoftware weiterzuleiten.

Zur Handhabung großer Mengen historischer Daten empfehlen wir, entweder das proprietäre Format von "WinStudio" oder eine dedizierte relationale Datenbank wie z.B. Microsoft SQL Server oder ORACLE zu verwenden.

17.8.8 Verwendung von MySQL

"WinStudio" kann mit MySQL-Datenbanken kommunizieren, dafür muss jedoch der ADO.Net-Anbieter für MySQL installiert werden.

Der für "WinStudio" erforderliche Anbieter ist MySQL Connector/Net und zum Zeitpunkt der Erstellung dieser Dokumentation konnte die notwendige Software von der offiziellen MySQL-Seite herunter geladen werden. (Bitte beachten Sie, dass die verlinkte Seite sich unserem Einfluss entzieht und ohne vorherige Ankündigung geändert werden kann.)

Nach der Installation des Anbieters können Sie das Eigenschaftsblatt Datenbankkonfiguration auf Seite 502 zum Konfigurieren einer MySQL-Datenbankverbindung verwenden. Anders als bei anderen Datenbanktypen kann allerdings der Dialog "Eigenschaften Datenverknüpfung" (Zugriff durch Klicken auf **...** rechts neben dem Feld **Verbindungs-String**) nicht zum Erstellen des Verbindungs-Strings verwendet werden. Statt dessen muss der Verbindungs-String im folgenden Format direkt eingegeben werden:

Provider=MYSQLCLIENT; Server=myServerAddress; Database=myDataBase; Uid=myUsername; Pwd=myPassword;

Die folgenden optionalen Parameter können an den Verbindungs-String angehängt werden:

Optionale Parameter für den Verbindungs-String MySQL Connector/Net

Parameter	Beschreibung
Port=number;	Gibt an, welcher Port für die Verbindung verwendet werden soll. Der Standard-Port ist 3306, es kann aber ein beliebiger Port festgelegt werden, solange dieser zur Serverkonfiguration passt. Wenn ein Port mit -1 angegeben ist, verwendet die Verbindung das Named-Pipes-Netzwerkprotokoll (siehe "Protokoll" weiter unten).
Server=myServerAddress1 & myServerAddress2 & … & myServerAddressN;	Verwenden eines beliebigen Servers in einer replizierten Serverkonfiguration.
Encryption=true;	Aktiviert SSL-Verschlüsselung für alle zwischen Client und Server gesendete Daten. Auf dem Server muss ein gültiges Zertifikat installiert sein.
Encrypt=true;	Eine Alternative zu "Encryption" oben, falls Fehler auftreten.
Default Command Timeout=milliseconds;	Legt eine Standard-Zeitüberschreitung für Befehle für die Verbindung fest. Diese überlagert nicht die Zeitüberschreitungseigenschaften von individuellen Befehlen.

Parameter	Beschreibung
Connection Timeout=seconds;	Gibt an, wie lange der Client auf eine Serververbindung wartet, bevor der Versuch abgebrochen wird.
Ignore Prepare=true;	Weist den Datenbankanbieter an, **Command.Prepare**-Anweisungen zu ignorieren, um Korruption durch vom Server erteilte Befehle zu vermeiden.
Protocol=myProtocol;	Gibt an, welches Netzwerkprotokoll verwendet werden soll. Der Standardwert ist "socket" oder "tcp", es können aber auch "pipe" zur Verwendung einer Named-Pipes-Connection oder "memory" zur Verwendung eines gemeinsamen verwendeten Speicherobjekts angegeben werden.
Shared Memory Name=MySQL;	Gibt den Namen des gemeinsamen verwendeten Speicherobjekts für die Kommunikation an. (Dieser Parameter gilt nur, wenn der "Protokoll"-Parameter oben auf "memory" eingestellt ist.)
CharSet=UTF8;	Gibt an, welcher Zeichensatz für die Verschlüsselung von Anfragen an den Server verwendet werden soll. Bitte beachten Sie, dass Abfrageergebnisse mit dem gleichen Zeichensatz verschlüsselt werden, mit dem die Daten selbst erfasst werden.

☞ Diese Vorgehensweisen wurden mit MySQL v5.1.11 und MySQL Connector/Net v6.2.0 getestet.

18 Fehlerbeseitigung und häufig gestellte Fragen

18.1 Allgemeine Fehlerbeseitigung

Wenn Sie technische Unterstützung benötigen, müssen Sie über einige Informationen verfügen, bevor Sie den technischen Support kontaktieren. Bei allen Problemen sind Informationen zur Ereignissen und Aktion zur Entdeckung des Problems äußerst hilfreich. Die Beschreibung muss so genau wie möglich sein und Sie sollten keinesfalls improvisieren, da dies die zur Fehlerbeseitigung benötigte Zeit und die Vorgehensweise stark beeinflussen kann.

Vor Kontaktaufnahme mit dem technischen Support

Die folgenden Schritte sollten vor dem Kontaktieren des technischen Supports durchgeführt werden:

- **Ziehen Sie die Dokumentation zu Rate** Die Hilfefunktion ist in allen WinStudio-Editoren verfügbar. Drücken Sie "F1", um die WinStudio-Hilfe zu öffnen.
- **Ziehen Sie kürzlich an Ihrem System vorgenommene Änderungen in Betracht** Wenn etwas vorher funktioniert hat, überlegen Sie, was sich geändert hat. Die Installation neuer Software oder allgemeine Systemänderungen können die Leistung und allgemeine Funktionalität anderer Software auf Ihrem System beeinflussen.
- **Versuchen Sie, das Problem in einer neuen Date zu reproduzieren** Wenn das Problem in einer neuen Testdatei nicht reproduziert werden kann, vergleichen Sie die neue Datei mit Ihrer Originaldatei, um die Unterschiede zu finden und zu eliminieren. Dadurch kann die Fehlerursache eingegrenzt werden.
- **Versuchen Sie, das Problem auf einem anderen Rechner zu reproduzieren** Wenn das Problem auf einem anderen Rechner nicht auftritt, betrachten Sie die Unterschiede zwischen den beiden Systemen. Ist dies der Fall, besteht wahrscheinlich ein Systemkonflikt.

Überprüfen Ihres Projekts

Wenn Tags in einer Tag-Datenbank geändert, neu sortiert oder gelöscht werden oder wenn Web-Einstellungen neu konfiguriert werden, muss das Projekt erneut verifiziert werden, um alle Bilder und Arbeitsblätter auf den aktuellen Stand der Datenbank zu bringen. Wählen Sie dazu im Kontextmenü von WinStudio den Eintrag **Anwendung verifizieren**.

Verwandte Veröffentlichungen

Die **WinStudio Kurzanleitung** ist für neue Benutzer bestimmt. Diese Veröffentlichung enthält Informationen zu den grundlegenden Funktionen von "Rexroth WinStudio" und ist im Ordner **Dokumentation** auf der Installations-CD von "Rexroth WinStudio" zu finden.

Die "Benutzerhandbücher" für die Treiber erläutern das Konfigurieren der Direktkommunikationstreiber gemäß ihrer spezifischen Protokollmerkmale. Für jeden Treiber wird eine Dokumentation zur Verfügung gestellt. Sie können die Dokumentation eines Treibers anzeigen, indem Sie im Ordner "Kommunikation" den entsprechenden Treiber-Ordner anwählen und "F1" drücken. Alternativ können Sie die Treiberbeschreibung aus dem Verzeichnis

[…]\WinStudio\Drv öffnen.

Kontaktaufnahme mit dem technischen Support

Wenn Sie in der Produktdokumentation bzw. im Hilfesystem keine Antwort auf Ihre technische Frage finden können, stehen unsere Spezialisten vom technischen Support Ihnen zur Unterstützung bei der Wartung aktueller Produkte zur Verfügung.

Bitte versuchen Sie, das Problem vor Kontaktaufnahme mit dem technischen Support zu definieren, damit Sie die Schritte, die zu dem Problem geführt haben, wiederholen können und genau erläutern können, wann und wie das

Problem auftrat. Um Ihnen helfen zu können, muss der Supportmitarbeiter wissen, wo genau das Problem liegt. Diese Schritte helfen uns dabei, Ihr Problem schneller einzugrenzen und zu lösen.

Bitte halten Sie die folgenden Informationen bereit:

- Hardware-Umgebung: verfügbarer Speicherplatz, Prozessortyp, Ausgabegerät
- Software-Umgebung: Betriebssystem, Windows®-Version, Netzwerkplattform
- Produktname, Versionsnummer und Produktseriennummer
- Größe des in Ihrem System installierten Speichers
- Größe des freien Festplattenspeicherplatzes in Ihrem System
- Bildschirmauflösung (Bildschirmgröße in Pixel, z.B. 1024 x 768)
- Farbtiefe des Bildschirms (Anzahl der Farben oder Bits, z.B. 256 Farben oder 8-Bit-Farbe)
- Hersteller der Grafikkarte, Modellname und Treiberversionsnummer
- Hersteller und Modellname der Soundkarte
- eine Liste der an den Rechner angeschlossenen externen Geräte
- kurze Beschreibung des Problems bzw. Fehlers und der genaue Text eventuell ausgegebener Fehlermeldungen
- eine Beschreibung der Schritte, die Sie bereits zur Fehlerbehebung unternommen haben, z.B. auf wie vielen Geräten Sie Tests durchgeführt haben und ob das Problem sich ein einer neuen Datei wiederholen lässt
- Schritte zum Wiederholen des Problems, wenn es reproduzierbar ist. Wenn das Problem nicht reproduzierbar ist, könnte es sich eher um ein Entwicklungsproblem als um ein Produktproblem handeln.

Stürzt das Projekt während der Laufzeit vollständig ab, wird ein Fehlerbeseitigungsbericht erstellt und unter

[…]\projekt_name\Web\Dump\WindDump.dmp gespeichert

Bitte halten Sie diese Datei bereit, um sie ggf. dem technischen Support zur Analyse zuzusenden.

Wenn Sie uns kontaktieren, halten Sie bitte Ihre Systeminformationen bereit.

18.2 Häufig gestellte Fragen

Datenbank und Sicherheitssystem

F. Was wird im Ordner "Verteilte Variablen" gespeichert?

A. Im Ordner "Verteilte Variablen" werden die Tags gespeichert, die aus dem mit dem "WinStudio"-Projekt verknüpften Steuerungen importiert werden.

F. Wie kann gezählt werden, wie viele Tags in der Projekt-Datenbank konfiguriert sind?

A. Die Anzahl der aktuell im Projekt verwendeten Tags wird in der Statuszeile am unteren Rand der Entwicklungsumgebung angezeigt. Jedes Array-Element und jedes Klassenelement der in der Tag-Datenbank von WinStudio konfigurierten Tags wird gezählt.

F. Wie kann ich die Liste der „Benutzer“ einsehen, die ich meinem Projekt während der Laufzeit mit der Funktion CreateUser auf Seite CreateUser hinzugefügt habe?

A. Führen Sie den folgenden Befehl aus:

"WinStudio\BIN\Studio Manager.exe" "WinStudio\BIN\ExtUser.dll"

(zum Beispiel:

"E:\Rexroth\IndraWorks\WinStudio\BIN\Studio Manager.exe" "E:\Rexroth \IndraWorks\WinStudio\BIN\ExtUser.dll").

Dieser Befehl öffnet einen Dialog. Sie können die mit der Funktion **CreateUser()** erstellten Benutzer sehen und dann Benutzer erstellen oder löschen.

Grafiken

F. Wie kann ich ein ActiveX-Objekt in ein Projekt einfügen und konfigurieren.

A. Zum Einfügen eines ActiveX-Objekts in ein WinStudio-Bild:

1. Öffnen Sie ein WinStudio-Bild im Bildeditor und klicken Sie im Werkzeugfenster "WinStudio Screen Tools" auf **ActiveX-Steuerelement**.
2. Wählen Sie das einzufügende ActiveX-Steuerelement aus der Liste und klicken Sie dann auf **OK**. Das ActiveX-Objekt erscheint im Bild. (Nicht registrierte ActiveX-Objekte sind in diesem Listenfeld nicht verfügbar.)
3. Führen Sie einen Doppelklick auf das ActiveX-Objekt aus und weisen Sie ihm einen Namen zu (Eingabe eines Werts im Feld **Name**). Die Liste der Animationen und Methoden können durch Auswahl der Schaltfläche **Methoden** angezeigt werden. Die statischen Eigenschaften können mithilfe der Schaltfläche **Eigenschaften** eingestellt werden (eine detaillierte Beschreibung der Objekteigenschaften befindet sich in der vom Komponentenentwickler bereitgestellten Komponentendokumentation).

Zur Bewertung des ActiveX-Komponente während der Laufzeit stehen drei Funktionen zur Verfügung:

- **XGet(strName,strProperties)**: Gibt den Wert der Eigenschaften **strProperties** des Objekts **strName** zurück.

  Die Liste der Eigenschaften, die verwendet werden können, sind im Dialog "Objekteigenschaften" des Objekts aufgeführt, mit der Syntax

  <Eigenschaften-Name>(PropGet)

  (zum Beispiel **Color(PropGet)**).

- **XSet(strName,strProperties,Value)**: Schreibt den Wert **Value** in die Eigenschaft **strProperties** des Objekts **strName**. Die Liste der Eigenschaften, die im Projekt eingestellt werden können, sind im Dialog "Objekteigenschaften" des Objekts aufgeführt, mit der Syntax

  <Eigenschaften-Name>(PropPut)

  (zum Beispiel **Color(PropPut)**).

- **XRun(strName, strMethod, Parameter1, Parameter2, …, ParameterN)**: Führt die Methode **strMethod** aus dem Objekt **strName** aus, gemäß der Parameter **Parameter1, Parameter2, …, ParameterN**.

  Die Liste der im Objekt verfügbaren Methoden sind im Dialog "Objekteigenschaften" des Objekts aufgeführt, mit der Syntax

  <Methodenname>(Method)

  (zum Beispiel **OpenFile(Method)**).



Stellen Sie vor dem Einfügen eines ActiveX-Steuerelements (üblicherweise eine OCX-Datei) in das Projekt sicher, dass es ordnungsgemäß im Rechner registriert wurde.



Die Anzahl der der in der Funktion "XRun auf Seite 764" gesetzten Parameter kann zwischen 0 und 255 variieren und hängt von der jeweiligen ActiveX-Komponente ab. Zum Setzen der Parameter können Tags verwendet werden. Allerdings muss der Tag-Typ der Art des Komponentenparameters entsprechen (Boolean, Integer, String oder Real).

F. Wie kann ich ein Bild festlegen, das sich bei jedem Starten des Projekts öffnet?

A. Öffnen Sie den WinStudio-Eigenschaftendialog, klicken Sie in das Register **Viewer** und geben Sie dann den Namen des Startbildschirms im Feld **Startbildschirm** ein.

F. Wie kann ich ein Hintergrundbild ins Bild einfügen?

A. Führen Sie einen Rechtsklick auf das Bild aus und wählen Sie die Option **Bildattribute** aus dem Kontextmenü. Aktivieren Sie das Kontrollkästchen **Hintergrund aktivieren** und wählen Sie aus der Combobox neben der Bezeichnung das Bildformat aus. Kopieren Sie die Bilddatei in den Unterordner "Bild" des Projekts und ändern Sie den Namen in den Namen des Bildes (**BildName.scr**). Durch Verwendung der Option **Gemeinsam verwendetes Bild** kann eine Bitmap-Datei in den Unterordner "Bild" kopiert und somit das Bild von mehr als einem Bild verwendet werden. In diesem Fall ist es notwendig, den Namen der Bitmap im Feld **Gemeinsam verwendetes Bild** einzugeben.

Tasks

F. Wie kann der Historientrend in eine ASCII-Datei konvertiert werden?

A. Zum Konvertieren einer Historien-Trenddatei in das ASCII-Format wird die Datei **"..\WinStudio\bin\hst2txt.exe"** in den Pfad **"\<Projekt-Verzeichnis>\hst\"** kopiert. Alternativ kann die Funktion "HST2TXT" in einem "Mathematik-Arbeitsblatt" zum automatischen Konvertieren von binären Dateien in Textformat verwendet werden, ohne Verwendung eines DOS-Fensters.

F. Wie können Daten über ein ODBC-Protokoll mit Fox Pro ausgetauscht werden?

A. Beim Austausch von Daten mit einer FOX-Pro-Datenbank ist es notwendig, den Parameter "UseQuote=0" aus dem Abschnitt [ODBC] in der Datei **projekt_name.app** festzulegen.

F. Wie kann ein Feld **DATUM** für eine ODBC-Schnittstelle mit einem Oracle-Paket eingestellt werden?

A. Konfigurieren Sie die „Spalten“-Zellen im ODBC-Arbeitsblatt mit der Syntax **SpaltenName.ts** (zum Beispiel: **MeinDatum.ts**).

F. Wie kann ich ein Arbeitsblatt "Mathematik" beim Hochfahren des Projekts ausführen und ein anderes Arbeitsblatt "Mathematik" beim Herunterfahren?

A.

- **Hochfahren**: Führen Sie ein Arbeitsblatt "Mathematik" beim Hochfahren aus, indem Sie ein Arbeitsblatt "Mathematik" erstellen und den Ausdruck **<TagName>=0** (zum Beispiel **StartTag=0**) im Feld **Ausführung** eingeben. Legen Sie in der letzten Zeile des Arbeitsblatts "Mathematik" den Wert "1" im Tag **<TagName>** fest. Der Tag **<TagName>** sollte vom Typ **Boolean** sein.
- **Herunterfahren**: Anstatt die Funktion **ShutDown()** direkt auszuführen, führen Sie ein Arbeitsblatt "Mathematik" aus und konfigurieren die Funktion **ShutDown()** in der letzten Zeile des Arbeitsblatts "Mathematik".

Kommunikation

F. Wie wird ein Alarm „Kommunikationsfehler“ festgelegt?

A. Konfigurieren Sie im Feld **Status Schreiben** oder **Status Lesen** des Treiberarbeitsblatts einen Tag und konfigurieren Sie einen Alarm für den Fall, wenn dieser Tag nicht "0" (null) ist.

F. Wie kann ich mit einer Siemens S7-200 SPS kommunizieren, ohne Prodave-Software zu verwenden?

A. Die Siemens S7-200 SPS verfügt über einen Freeport, der jedes beliebige Protokoll per SPS-Programmierung implementieren kann. Es gibt von Siemens vertriebene, kostenlose SPS-Software, die das Modbus-Protokoll in dem SPS-Freeport implementiert (weitere Details erhalten Sie vom Siemens Support). Durch Verwendung dieser Software in der SPS mit dem Modbus-Treiber von "WinStudio" (MODBU) können Informationen zwischen den beiden Elementen ausgetauscht werden.

F. Wie kann ich während der Laufzeit Kommunikationstreiber starten und anhalten?

A. Zur Handhabung der Ausführung der Kommunikationstreiber während der Laufzeit stehen drei Funktionen zur Verfügung:

- Starten aller im Projekt konfigurierten Treiber: **StartTask("Treiber")**

  Zum Beispiel: **StartTask("Treiber")**

- Starten eines bestimmten im Projekt konfigurierten Treibers:

  WinExec("..\WinStudio\bin\Studio Manager.exe" + " + "..\WinStudio\bin \Treiber.dll" + " + " TreiberName")

  Zum Beispiel:

  WinExec(Asc2Str(34) + "Bin\Studio Manager.exe" + Asc2Str(34) + " " + Asc2Str(34) + "Bin\Treiber.dll" + Asc2Str(34) + " " + Asc2Str(34) + "MODBU" + Asc2Str(34))

☞ Die Funktion **Asc2Str(34)** wird verwendet, um Anführungszeichen für Pfade hinzuzufügen, in denen Leerzeichen enthalten sind.

- Anhalten eines bestimmten im Projekt konfigurierten Treibers:

  EndTask("DriverDriverName")

  Zum Beispiel:

  EndTask("DriverMODBU")

Mit den Funktionen **StartTask(TaskName)** und **EndTask(TaskName)** können andere Tasks gestartet oder angehalten werden.

Zum Beispiel **StartTask("Viewer"), Endtask("Viewer")**.

F. Wie lauten die Parameter des "WinStudio"-DDE-Servers?

A. Die "WinStudio"-DDE-Server- und NetDDE-Serverparameter sind in der folgenden Tabelle aufgeführt:

Komm. Typ	Anwendung	Topic	Element
Network DDE	\\<Computer Name>\NDDE$	UNISOFT$	<TagName>
Local DDE	UNIDDE	DB	<TagName>
Network DDE	\\<Computer Name>\NDDE$	UNISOFT$	<TagName>
Local DDE	UNIDDE	DB	<TagName>

F. Wie können Daten mit Excel über NetDDE ausgetauscht werden?

Fehlerbeseitigung und häufig gestellte Fragen

A. Net DDE kann zum Datenaustausch zwischen vernetzten Stationen über das DDE-Protokoll verwendet werden.

1. Starten Sie das DDE-Server-Modul aus der Entwicklungsanwendung in der Taskkonfiguration (Eintrag **Status** im Kontextmenü von WinStudio).
2. Führen Sie Excel auf der Remote-Station aus.
3. Öffnen Sie ein Excel-Arbeitsblatt und füllen Sie die Zellen, die mit "WinStudio" Daten austauschen müssen, mit der folgenden Syntax aus:

   ='\ computer name \NDDE$'|'UNISOFT$'! tag name

   Zum Beispiel:

   ='\PC\NDDE$'|'UNISOFT$'!second

- Beim Ausführen unter Windows NT oder Windows2000 muss sichergestellt werden, dass die Dienste Network DDE und Network DDE DSDM ausgeführt werden. (Zum Starten dieser Dienste verwenden Sie den Dienste-Verknüpfung in der Systemsteuerung.)
- Beim Ausführen unter Windows 98 muss das Programm **<Windows-Verzeichnis>\etdde.exe** auf beiden Rechnern ausgeführt werden (zum Beispiel **c:\Windows\etdde.exe**).

F. Ist die "WinStudio"-OPC-Schnittstelle zur OPC-Spezifikation v1.0a oder v2.0 konform?

A. Die "WinStudio"-Module OPC Client und OPC Server sind mit beiden OPC-Spezifikationen v1.0a und v2.0 konform.

F. Wie erhalte ich Fehler von den Anwendungen Intellution / GE Fanuc iFIX?

A. Wenn Ihr Projekt über TCP/IP auf Seite 401 mit einer iFIX-Anwendung kommuniziert, sollten Sie den folgenden Schlüssel in Ihre Projektdatei aufnehmen (d.h. **projekt_name.APP**):

[TCP]

SetQualityToBadOnError=1

Wenn die iFIX-Anwendung danach während der Laufzeit einen Fehler erzeugt, wird die Qualität der betroffenen Tags in Ihrem Projekt auf BAD gesetzt. Diese Information erhalten Sie durch Auslesen des Tag-Felds Qualität auf Seite 142 (d.h. **tagname->Quality**).

Allgemeines

F. Welche Betriebssysteme sind mit "Rexroth WinStudio" und CEView kompatibel?

A. siehe Tabelle unten. „N“ bedeutet, dass das Betriebssystem nicht unterstützt wird und „J“ bedeutet, dass das Betriebssystem unterstützt wird.

Betriebssystem		"Rexroth WinStudio"				CEView			
Name	Version	v2.x	v3.x bis v4.1	v4.2 bis v6.0	v6.1 oder höher	v3.x	v4.x	v5.x	v6.x
Windows Vista	alle	N	N	N	J	N	N	N	N
Windows XP	alle	N	J	J	J	N	N	N	N
Windows 2000	alle	N	J	J	J	N	N	N	N
Windows NT	v4.0+SP4 oder höher	J	J	J	N	N	N	N	N

Betriebssystem		"Rexroth WinStudio"				CEView			
Name	Version	v2.x	v3.x bis v4.1	v4.2 bis v6.0	v6.1 oder höher	v3.x	v4.x	v5.x	v6.x
Windows ME	alle	N	J	N	N	N	N	N	N
Windows 98	alle	N	J	N	N	N	N	N	N
Windows 95	alle	J	J	N	N	N	N	N	N
Windows CE	v2.12	N	N	N	N	J	N	N	N
	v3.x	N	N	N	N	J	J	J	J
Windows CE.Net	v4.0	N	N	N	N	N	N	J	J
	v4.1	N	N	N	N	N	N	J	J
	v4.2	N	N	N	N	N	N	J	J
	v5.0	N	N	N	N	N	N	N	J
	v6.0	N	N	N	N	N	N	N	J

F. Wie kann ich "WinStudio" automatisch beim Einschalten des Rechners starten?

A. Erstellen Sie eine Verknüpfung zu **[…]\WinStudio\Bin\RunStudio.exe** und legen Sie diese in Ihren Autostart-Ordner (**C:\WINDOWS\Dokumente und Einstellungen\All Users\Startmenü\Programme\Autostart**).

F. Wie kann ich Dr. Watson deaktivieren?

A. Nachstehend finden Sie eine Schritt-für-Schritt-Anleitung zum Deaktivieren von Dr. Watson unter Windows NT:

1. Führen Sie das Programm **<Windows-Verzeichnis>\RegEdit.exe** aus

   (zum Beispiel

   C:\WinNT\Regedit.exe)
2. Wählen Sie den Pfad **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft \Windows NT\CurrentVersion\AeDebug**
3. Stellen Sie im Parameter „Auto“ aus dem gewählten Pfad den Wert 0 (null) ein.
4. Schließen Sie das Fenster "Registry Editor".

HINWEIS

Beim Bearbeiten von Parametern im Programm "Registry Editor" muss mit besonderer Vorsicht vorgegangen werden, weil einige davon das Gesamtverhalten des Betriebssystems verändern können.

Das Datumsformat in "WinStudio" kann auf zwei Arten verändert werden:

- **Offline**: Setzen Sie die Parameter **Order=<Datumsformat>**

  (z.B. "TMJ") und **Separator=Trennzeichen**

  (z.B. "/" oder .) aus dem Abschnitt [International] der Datei **[…]\WinStudio\Bin\Program Settings.ini**.

Fehlerbeseitigung und häufig gestellte Fragen

☞ Sie müssen Ihr Projekt verifizieren, um die Datumseinstellungen auf zuvor gespeicherte Webseiten anzuwenden.

- **Online**: Verwenden Sie die Funktion **SetDateFormat(strSeparator,strDateFormat)**.

  Zum Beispiel **SetDateFormat("/","MDY")**.

F. Welche Merkmale werden von CEView nicht unterstützt?

A. Die von CEView nicht unterstützten Hauptmerkmale sind: DDE, NetDDE, ODBC, PasteLink und eine Reihe von Funktionen.

F. Wie kann ich beim Starten eines Projekts in CEView einen Startbildschirm anzeigen?

A. Um in Ihrem Projekt das Anzeigen eines Startbildschirms beim Hochfahren zu aktivieren, fügen Sie der Datei **CEView.ini** des Geräts den folgenden Schlüssel hinzu:

Datei CEView.ini

```
 [OEM] SpashWnd = Path to bitmap File // default is Splash.bmp
SplashWndTime = Time in milliseconds // default is 1000
```

F. Wie kann ich Laufzeitmeldungen und -Fehler auf einem bildschirmlosen Windows-Embedded-Gerät, also einem Gerät ohne Display, sehen?

A. Wenn das Gerät über eine Netzwerkverbindung verfügt, kann das Werkzeug Remote LogWin auf Seite 448 zum Anzeigen der vom Gerät erzeugten Meldungen verwendet werden.

Sie können die Meldungen auch in einer Speicherauszugsdatei (**dump.txt**) speichern und bei Bedarf darauf zugreifen. Zum Erstellen der Datei fügen Sie der Datei **CEView.ini** des Geräts den folgenden Schlüssel hinzu**: [OEM] BlindDevice = 1**

F. Was sind die wichtigsten Schritte zum Erstellen einer web-basierten Anwendung?

A. Befolgen Sie die nachstehenden Schritte:

1. Entwickeln Sie das Projekt lokal. Verwenden Sie für die Bilder, die im HTML-Format gespeichert werden sollen, keine Merkmale, die nicht von Thin Clients unterstützt werden.
2. Nach dem Speichern der Bilder im Standardformat (**Speichern** im Kontextmenü) speichern Sie die Bilder, die für den Thin Client verfügbar sein müssen, im HTML-Format (**Speichern als HTML** im Kontextmenü).
3. Öffnen Sie den Dialog "Web-Einstellungen" aus den Kontextmenü des Ordners "WinStudio".
4. Geben Sie im Feld **Datenserver IP** die IP-Adresse der Serverstation ein (d.h. der Station, auf der das Projekt läuft).
5. Öffnen Sie die Taskkonfiguration (Eintrag "Status" im Kontextmenü von WinStudio)und stellen Sie dann das TCP/IP-Servermodul auf **Startup=Automatisch** ein.
6. Öffnen Sie den Editor "Anwendungsvariablen" und stellen Sie in der Spalte **Array** für alle Tags, die zwischen dem Server und der Thin-Client-Station Daten austauschen müssen, die Option **Server** anstelle von **Lokal** ein.

7. Überprüfen Sie das Projekt (**Anwendung verifizieren** im Kontextmenü von WinStudio), um die HTML-Dateien mit den neuen Einstellungen zu aktualisieren.
8. Wenn kein Web-Server auf dem Rechner läuft, kopieren Sie das Programm "NTWebServer.exe" aus dem Unterordner "Bin" des Programmverzeichnisses in das Web-Verzeichnis (z.B. **\projektname\Web**) und führen Sie es aus. Das Web-Server-Programm wird im Stammverzeichnis des Servers ausgeführt. Der Web Server wird zum Exportieren von Daten (Web-Dateien) im HTTP-Protokoll zu den Thin Clients benötigt.
9. Führen Sie das Projekt auf der Serverstation aus.
10. Geben Sie unter Verwendung eines Browsers (z.B. Internet Explorer) in der Thin-Client-Station die URL-Adresse ein, um das im HTML-Format gespeicherte Bild herunter zuladen.

    (z.B. **http://ServerIPAdresse/BildName.html**).

☞ Der Thin Client erfordert eine ActiveX-Komponente (**ISSymbol.ocx**) zum Handhaben von Bildern im Browser. Wenn der Thin Client mit dem Internet verbunden ist, wird die Komponente automatisch herunter geladen und registriert. Andernfalls ist es notwendig, diese in das Verzeichnis **\OSPfad\System32** des Thin Clients zu kopieren und mit dem Befehl **regsvr32 ISSymbol.ocx** zu registrieren. Diese Datei befindet sich im Ordner **\BIN** des Installationsverzeichnisses von "WinStudio".

F. Wie erhalte ich die Kommunikation zwischen einem über Proxy verbundenen Thin Client und einer Web-Gateway-Anwendung mit Microsoft IIS aufrecht?

A. Microsoft Internet Information Services (IIS) verfügt über eine Konfigurationsoption zum Aufrechterhalten von HTTP-Verbindungen. Wird diese Option aktiviert, kann sie evtl. mit über Proxy verbundenen Thin Clients in Konflikt kommen. Zum Deaktivieren dieser Option:

1. Starten Sie die Verwaltung für Internetdienste.
2. Öffnen Sie im Fenster "Internet-Informationsdienste" den lokalen Server ("* "server name"").
3. Führen Sie einen Rechtsklick auf **Standard-Websites** aus und wählen Sie den Eintrag **Eigenschaften** aus dem Kontextmenü. Der Dialog "Eigenschaften Standard-Website" wird angezeigt.
4. Wählen Sie die Registerkarte **Website** im Dialog "Eigenschaften Standard-Website".
5. Deaktivieren Sie im Bereich "Verbindungen" der Registerkarte **Website** die Option **HTTP Keep-Alives aktiviert**.
6. Klicken Sie auf **OK**, um die Änderung zu speichern und den Dialog zu schließen.

F. Wie sende ich eine E-Mail vom "WinStudio"-Projekt?

A. Befolgen Sie die nachstehenden Schritte:

- Führen Sie die Funktion **CNFEMail(strSMTP,strFrom,strPOP3,strUser, strPassword,numTimeOut)** aus, um die allgemeinen Parameter zum Versenden von E-Mails zu konfigurieren. Nach einmaligem Ausführen dieser Funktion werden die so eingestellten Parameter im System beibehalten, bis das Projekt geschlossen wird. Daher führen die meisten Projekte diese Funktion nur ein Mal aus, nach dem Starten des Projekts:

- Führen Sie die Funktion

  SendEMail(strSubject,strMessage,strTO) and/or SendEMailExt(strSubject,strMessage,strTO,strCC,strBCC, strFile1,...,strFileN)

  jedes Mal aus, wenn eine E-Mail-Nachricht gesendet werden soll. Der wichtigste Unterschied zwischen den beiden Funktionen ist in der nachfolgenden Tabelle aufgeführt:

Eigenschaft	SendEmail	SendEmailExt
Ausführung	synchron	asynchron
Unterstützt Betreff-Text	J	J
Unterstützt Nachrichtentext	J	J
Unterstützt AN-Adressen	J	J
Unterstützt CC-Adressen	N	J
Unterstützt BCC-Adressen	N	J
Unterstützt Dateianhänge	N	J

F. Der Laufzeittask (TCP/IP, OPC, DDE, ODBC, etc.) funktioniert nicht.

A. Stellen Sie sicher, dass der Laufzeittask in der Taskkonfiguration auf **Automatisch** eingestellt ist (Eintrag **Status** im Kontextemenü des Ordners WinStudio). Wählen Sie einen auszuführenden Laufzeittask (z.B. TCP/IP-Server), klicken Sie auf **Start** und stellen Sie diesen dann auf **Automatisch** ein.

F. Der Browser vom Thin Client zeigt das Bild nicht an und gibt eine Warnmeldung zu ISSymbol.ocx aus.

A. Stellen Sie sicher, dass der Laufzeittask im Dialog "Ausführungs-Tasks" auf **Automatisch** eingestellt ist (Eintrag **Status** im Kontextemenü des Ordners WinStudio). Wählen Sie einen auszuführenden Laufzeittask (z.B. TCP/IP-Server), klicken Sie auf **Start** und stellen Sie diesen dann auf **Automatisch** ein.

F. Der Browser des Thin Clients erzeugt eine Fehlermeldung zum Fehlen von **ISSymbol.ocx** und zeigt die Bilder vom Server nicht an.

A.ISSymbol.ocx ist das vom Browser des Thin Clients verwendete ActiveX-Objekt zum Betrachten der Webseiten. Wenn der Thin Client mit dem Internet verbunden ist, wird das Steuerelement **ISSymbol.ocx** automatisch herunter geladen und in der Thin-Client-Station registriert. Andernfalls muss dieses in den Ordner **\WinNT\System32** der Thin-Client-Station kopiert und manuell registriert werden. Nach der Registrierung kann der Browser die Seiten sehen.

☞ Verwenden Sie den Befehl **regsvr32 ISSymbol32.ocx** zum Registrieren der ActiveX-Komponente im Thin Client.

F. Die Bilder werden auf dem Thin Client angezeigt (Browser), aber die Daten (Tag-Werte) werden nicht vom Server gelesen.

A. Stellen Sie sicher, dass die Parameter in der Spalte **Bereich** im Editor "Anwendungsvariablen" auf **Server** und nicht auf **Lokal** eingestellt ist. Die auf **Server** eingestellten Tags behalten den gleichen Wert im Server und im Thin Client (Browser). Die auf **Lokal** eingestellten Tags haben unabhängige Werte im Server und im Thin Client (Browser).

HINWEIS	**Nach dem Verändern der Tag-Einstellungen muss das Projekt verifiziert werden. Andernfalls werden die Änderungen in den Webseiten nicht aktualisiert.**

F. Die in der Befehlsanimation konfigurierten Ausdrücke „Bei Lösen“ werden nicht nicht ausgeführt.

A. Die Ausdrücke „Bei Lösen“ aus der Befehlsanimation werden nicht ausgeführt, wenn der Mauszeiger vor dem Lösen aus dem Objektbereich herausgezogen wird. Wenn das Kontrollkästchen **Lösen aus den Befehlsobjekteigenschaften** aktiviert ist, werden die „Bei Lösen“-Ausdrücke auch dann ausgeführt, wenn der Mauszeiger vor dem Lösen aus dem Objektbereich heraus bewegt wird.

F. Die Trendhistorie funktioniert nach dem Hinzufügen oder Entfernen von Tags im Arbeitsblatt "Trend" nicht mehr.

A. Wird ein Tag in ein Arbeitsblatt "Trend" eingefügt oder daraus entfernt wird, wird das Format der Historiendateien (***.hst**) verändert. Die gleiche ***.hst**-Datei kann nicht zwei verschiedene Formate haben, andernfalls werden die Daten nicht ordnungsgemäß vom Trendobjekt abgerufen. Für das Hinzufügen oder Entfernen von Tags für Historiendateien gibt es zwei gültige Vorgehensweisen: Erstellen Sie ein neues Arbeitsblatt "Trend" oder löschen Sie die alten ***.hst**-Dateien.

F. Der Wert von indirekten Tags (**@TagName**) wird im Thin-Client-Programm nicht angezeigt.

A. Wenn ein Bild als HTML gespeichert wird, speichert "WinStudio" eine Datei **<BildName>.tagl** im Unterordner "\WEB". Diese Datei enthält eine Liste aller im Bild konfigurierten Tags (Objekte und Animationen). Wenn ein Bild im Thin-Client-Browser geöffnet wird, werden die in **<BildName>.tagl** aufgeführten Tags für die TCP/IP-Kommunikation mit der Serverstation „aktiviert“. So wird eine optimierte Kommunikation zwischen der Serverstation und den Thin-Client-Stationen zur Verfügung gestellt.

Bei der Verwendung von indirekten Tags auf diese Weise (**@indirekter Tag**) tauschen die gezeigten Tags keine Daten mit dem Server aus, wenn sie nicht im Bild konfiguriert wurden. Anders ausgedrückt müssen die Tags, auf die im Bild gezeigt werden, auf einem Bildobjekt konfiguriert werden, um die TCP/IP-Kommunikation für diese Tags mit der Serverstation zu aktivieren.



Fügen Sie ein transparentes Rechteck (ohne Füllung und ohne Linie) in der Ecke des Bildes ein. Wenden Sie die Befehlsanimation auf dieses Rechteck an und konfigurieren Sie die Tags (auf die in der Thin-Client-Station während der Laufzeit mit indirekten Tags gezeigt werden kann) in den Feldern **Ausdruck** (lassen Sie die Felder **Variablenname** leer). Diese Tags werden zur Datei **<BildName>.TAGL** hinzugefügt und stehen zur TCIP/IP-Kommunikation mit der Serverstation zur Verfügung.

F. Welche Funktionalitäten werden von PocketPC-Plattformen nicht unterstützt (z.B. IPaq, Cassiopeia, Jornada)?

A. PocketPC-Geräte mit Windows-CE unterstützen einige Funktionalitäten nicht, die nicht von Windows-CE-Geräten unterstützt werden, die mit der „Standard“-Version von Windows CE betrieben werden:

Fehlerbeseitigung und häufig gestellte Fragen

Funktionalität nicht von PocketPC-Geräten unterstützt
DCOM (Distributed Component Object Model): Bedeutet, dass alle auf DCOM basierenden Merkmale (z.B. Remote-OPC-Kommunikation) nicht von PocketPC-Geräten unterstützt werden.
Die Funktion DialGetClientIP auf Seite 750 funktioniert nicht mit PocketPC-Geräten.

19 Anhang: WinStudio-interne Skriptsprache

19.1 Anhang: Interne Skriptsprache, Allgemeines

Dieser Abschnitt beschreibt alle Funktionen der internen Skriptsprache von Winstudio. Jede Funktionsbeschreibung enthält die vollständige Syntax, mögliche Ausgabewerte und Anwendungsbeispiele.

19.2 Liste der verfügbaren Funktionen

Die im folgenden beschriebenen Funktionen sind in "Rexroth WinStudio" verwendbar.

Protokollmeldung

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
Trace, Seite 557	synchron	unterstützt	unterstützt	nicht unterstützt

Arithmetisch

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
Abs, Seite 559	synchron	unterstützt	unterstützt	unterstützt
Div, Seite 560	synchron	unterstützt	unterstützt	unterstützt
Format, Seite 560	synchron	unterstützt	unterstützt	unterstützt
GetBit, Seite 564	synchron	unterstützt	unterstützt	unterstützt
Mod, Seite 564	synchron	unterstützt	unterstützt	unterstützt
Pow, Seite 565	synchron	unterstützt	unterstützt	unterstützt
ResetBit, Seite 565	synchron	unterstützt	unterstützt	unterstützt
Round, Seite 566	synchron	unterstützt	unterstützt	unterstützt
SetBit, Seite 567	synchron	unterstützt	unterstützt	unterstützt
SQRT, Seite 568	synchron	unterstützt	unterstützt	unterstützt
Swap16, Seite 568	synchron	unterstützt	unterstützt	unterstützt
Swap32, Seite 568	synchron	unterstützt	unterstützt	unterstützt
Trunc, Seite 569	synchron	unterstützt	unterstützt	unterstützt

Statistisch

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
Avg, Seite 570	synchron	unterstützt	unterstützt	unterstützt
Max, Seite 571	synchron	unterstützt	unterstützt	unterstützt
Min, Seite 571	synchron	unterstützt	unterstützt	unterstützt
Rand, Seite 572	synchron	unterstützt	unterstützt	unterstützt

Anhang: WinStudio-interne Skriptsprache

Logarithmisch

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
Exp, Seite 573	synchron	unterstützt	unterstützt	unterstützt
Log, Seite 573	synchron	unterstützt	unterstützt	unterstützt
Log10, Seite 574	synchron	unterstützt	unterstützt	unterstützt

Logisch

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
False, Seite 576	synchron	unterstützt	unterstützt	unterstützt
If, Seite 576	synchron	unterstützt	unterstützt	unterstützt
Toggle, Seite 577	synchron	unterstützt	unterstützt	unterstützt
True, Seite 577	synchron	unterstützt	unterstützt	unterstützt

Zeichenkette

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
Asc2Str, Seite 579	synchron	unterstützt	unterstützt	unterstützt
CharToValue, Seite 579	synchron	unterstützt	unterstützt	unterstützt
CharToValueW, Seite 580	synchron	unterstützt	unterstützt	unterstützt
ClassMembersToStrVector, Seite 581	synchron	unterstützt	unterstützt	unterstützt
NCopy, Seite 582	synchron	unterstützt	unterstützt	unterstützt
Num, Seite 583	synchron	unterstützt	unterstützt	unterstützt
Str, Seite 583	synchron	unterstützt	unterstützt	unterstützt
Str2Asc, Seite 583	synchron	unterstützt	unterstützt	unterstützt
StrCompare, Seite 584	synchron	unterstützt	unterstützt	unterstützt
StrCompareNoCase, Seite 585	synchron	unterstützt	unterstützt	unterstützt
StrFromInt, Seite 585	synchron	unterstützt	unterstützt	unterstützt
StrFromReal, Seite 586	synchron	unterstützt	unterstützt	unterstützt
StrFromTime, Seite 587	synchron	unterstützt	unterstützt	unterstützt
StrGetElement, Seite 588	synchron	unterstützt	unterstützt	unterstützt
StrLeft, Seite 588	synchron	unterstützt	unterstützt	unterstützt
StrLen, Seite 589	synchron	unterstützt	unterstützt	unterstützt
StrLower, Seite 589	synchron	unterstützt	unterstützt	unterstützt
StrRChr, Seite 589	synchron	unterstützt	unterstützt	unterstützt
StrRight, Seite 590	synchron	unterstützt	unterstützt	unterstützt
StrSetElement, Seite 590	synchron	unterstützt	unterstützt	unterstützt
StrStr, Seite 591	synchron	unterstützt	unterstützt	unterstützt
StrStrPos, Seite 591	synchron	unterstützt	unterstützt	unterstützt
StrTrim, Seite 592	synchron	unterstützt	unterstützt	unterstützt

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
StrTrimAll, Seite 593	synchron	unterstützt	unterstützt	unterstützt
StrUpper, Seite 593	synchron	unterstützt	unterstützt	unterstützt
ValueToChar, Seite 594	synchron	unterstützt	unterstützt	unterstützt
ValueWToChar, Seite 594	synchron	unterstützt	unterstützt	unterstützt

Datum & Uhrzeit

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
ClockGetDate, Seite 595	synchron	unterstützt	unterstützt	unterstützt
ClockGetDayOfWeek, Seite 596	synchron	unterstützt	unterstützt	unterstützt
ClockGetTime, Seite 597	synchron	unterstützt	unterstützt	unterstützt
DateTime2Clock, Seite 597	synchron	unterstützt	unterstützt	unterstützt
GetClock, Seite 598	synchron	unterstützt	unterstützt	unterstützt
Hour2Clock, Seite 598	synchron	unterstützt	unterstützt	unterstützt
SetSystemDate, Seite 599	synchron	unterstützt	unterstützt	unterstützt
SetSystemTime, Seite 599	synchron	unterstützt	unterstützt	unterstützt

Trigonometrisch

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
ACos, Seite 600	synchron	unterstützt	unterstützt	unterstützt
ASin, Seite 600	synchron	unterstützt	unterstützt	unterstützt
ATan, Seite 601	synchron	unterstützt	unterstützt	unterstützt
Cos, Seite 601	synchron	unterstützt	unterstützt	unterstützt
Cot, Seite 601	synchron	unterstützt	unterstützt	unterstützt
Pi, Seite 602	synchron	unterstützt	unterstützt	unterstützt
Sin, Seite 602	synchron	unterstützt	unterstützt	unterstützt
Tan, Seite 602	synchron	unterstützt	unterstützt	unterstützt

Öffnen und Schließen von Fenstern

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
Close, Seite 603	asynchron	unterstützt	unterstützt	nicht unterstützt
Open, Seite 604	asynchron	unterstützt	unterstützt	unterstützt
OpenPrevious, Seite 606	asynchron	unterstützt	unterstützt	unterstützt
ShowInplaceInput, Seite 607	asynchron	unterstützt	unterstützt	unterstützt
ShowMessageBox, Seite 608	asynchron	unterstützt	unterstützt	unterstützt

Anhang: WinStudio-interne Skriptsprache

Sicherheit

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
BlockUser, Seite 610	synchron	unterstützt	unterstützt	unterstützt
CheckESign, Seite 611	synchron	unterstützt	unterstützt	unterstützt
CreateUser, Seite 611	synchron	unterstützt	unterstützt	unterstützt
ExportSecuritySystem, Seite 612	synchron	unterstützt	unterstützt	unterstützt
GetSecuritySystemStatus, Seite 613	synchron	unterstützt	unterstützt	unterstützt
GetUserFullName, Seite 615	synchron	unterstützt	unterstützt	unterstützt
GetUserNames, Seite 615	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
GetUserPWDAging, Seite 616	synchron	unterstützt	unterstützt	unterstützt
GetUserState, Seite 617	synchron	unterstützt	unterstützt	unterstützt
RemoveUser, Seite 619	synchron	unterstützt	unterstützt	unterstützt
ImportSecuritySystem, Seite 617	synchron	unterstützt	unterstützt	unterstützt
SetPassword, Seite 619	synchron	unterstützt	unterstützt	unterstützt
UnblockUser, Seite 620	synchron	unterstützt	unterstützt	unterstützt

Modulaktivität

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
AppActivate, Seite 622	asynchron	unterstützt	unterstützt	unterstützt
AppIsRunning, Seite 624	synchron	unterstützt	unterstützt	unterstützt
AppPostMessage, Seite 624	synchron	unterstützt	unterstützt	unterstützt
AppSendKeys, Seite 625	synchron	unterstützt	nicht unterstützt	unterstützt
CleanReadQueue, Seite 626	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
CloseSplashWindow, Seite 626	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
DisableMath, Seite 627	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt
EnableMath, Seite 627	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt
EndTask, Seite 627	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt
ExitWindows, Seite 628	asynchron	unterstützt	nicht unterstützt	unterstützt
IsScreenOpen, Seite 629	synchron	unterstützt	unterstützt	nicht unterstützt
IsTaskRunning, Seite 629	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
IsViewerInFocus, Seite 630	synchron	unterstützt	nicht unterstützt	nicht unterstützt
KeyPad, Seite 630	asynchron	unterstützt	unterstützt	unterstützt
LogOff, Seite 632	asynchron	unterstützt	unterstützt	unterstützt

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
LogOn, Seite 632	asynchron	unterstützt	unterstützt	unterstützt
Math, Seite 633	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
PostKey, Seite 633	synchron	unterstützt	unterstützt	unterstützt
Recipe, Seite 634	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
Report, Seite 635	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
RunGlobalProcedureOnServer, Seite 640	synchron	unterstützt	unterstützt	unterstützt
RunVBScript, Seite 643	synchron	unterstützt	unterstützt	unterstützt
SecureViewerReload, Seite 644	synchron	nicht unterstützt	nicht unterstützt	unterstützt (nur Secure Viewer)
SendKeyObject, Seite 644	synchron	unterstützt	nicht unterstützt	nicht unterstützt
SetAppPath, Seite 646	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
SetViewerInFocus, Seite 647	synchron	unterstützt	nicht unterstützt	unterstützt
SetViewerPos, Seite 647	synchron	unterstützt	nicht unterstützt	unterstützt
ShutDown, Seite 648	synchron	unterstützt	unterstützt	nicht unterstützt
StartTask, Seite 648	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt
ViewerPostMessage, Seite 649	asynchron	unterstützt	unterstützt	unterstützt
Wait, Seite 650	synchron	unterstützt	unterstützt	unterstützt
WinExec, Seite 651	asynchron	unterstützt	unterstützt	unterstützt
WinExecIsRunning, Seite 653	synchron	unterstützt	unterstützt	unterstützt

Datei

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
DeleteOlderFiles, Seite 654	synchron	unterstützt	unterstützt	unterstützt
DirCreate, Seite 655	synchron	unterstützt	unterstützt	unterstützt
DirDelete, Seite 655	synchron	unterstützt	unterstützt	unterstützt
DirLength, Seite 656	synchron	unterstützt	unterstützt	unterstützt
DirRename, Seite 657	synchron	unterstützt	unterstützt	unterstützt
FileCopy, Seite 657	synchron	unterstützt	unterstützt	unterstützt
FileDelete, Seite 658	synchron	unterstützt	unterstützt	unterstützt
FileLength, Seite 659	synchron	unterstützt	unterstützt	unterstützt
FileRename, Seite 659	synchron	unterstützt	unterstützt	unterstützt
FindFile, Seite 661	synchron	unterstützt	unterstützt	unterstützt
FileWrite, Seite 659	synchron	unterstützt	unterstützt	unterstützt

Anhang: WinStudio-interne Skriptsprache

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
FindPath, Seite 662	synchron	unterstützt	unterstützt	unterstützt
GetFileAttributes, Seite 662	synchron	unterstützt	unterstützt	unterstützt
GetFileTime, Seite 663	synchron	unterstützt	unterstützt	unterstützt
GetHstInfo, Seite 663	synchron	unterstützt	unterstützt	unterstützt
GetLine, Seite 664	synchron	unterstützt	unterstützt	unterstützt
HST2TXT, Seite 667	asynchron	unterstützt	unterstützt	unterstützt
HST2TXTIsRunning, Seite 668	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
PDFCreate, Seite 671	synchron	unterstützt	nicht unterstützt	unterstützt
Print, Seite 672	asynchron	unterstützt	unterstützt	unterstützt
RDFileN, Seite 673	synchron	unterstützt	unterstützt	unterstützt
WebGetFile, Seite 673	synchron	unterstützt	unterstützt	unterstützt

Grafisch

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
AutoFormat, Seite 674	synchron	unterstützt	unterstützt	unterstützt
GetScrInfo, Seite 675	synchron	unterstützt	unterstützt	unterstützt
PrintSetup, Seite 676	asynchron	unterstützt	unterstützt	unterstützt
PrintWindow, Seite 677	asynchron	unterstützt	unterstützt	nicht unterstützt
ResetDecimalPointsTable, Seite 678	synchron	unterstützt	unterstützt	unterstützt
RGBColor, Seite 679	synchron	unterstützt	unterstützt	unterstützt
RGBComponent, Seite 679	synchron	unterstützt	unterstützt	unterstützt
SetDecimalPoints, Seite 681	synchron	unterstützt	unterstützt	unterstützt
SetDisplayUnit, Seite 682	synchron	unterstützt	unterstützt	unterstützt
SetTagDisplayUnit, Seite 683	synchron	unterstützt	unterstützt	unterstützt

Übersetzung

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
Ext, Seite 684	synchron	unterstützt	unterstützt	unterstützt
SetLanguage, Seite 685	synchron	unterstützt	unterstützt	unterstützt
SetTranslationFile, Seite 685	synchron	unterstützt	unterstützt	unterstützt

Multimedia

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
Play, Seite 686	Synchron/asynchron	unterstützt	unterstützt	unterstützt

Systeminfo

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
DbVersion, Seite 688	synchron	unterstützt	unterstützt	unterstützt
GetAppHorizontalResolution, Seite 689	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
GetAppPath, Seite 689	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
GetAppVerticalResolution, Seite 690	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
GetComputerIP, Seite 690	synchron	unterstützt	unterstützt	unterstützt
GetComputerName, Seite 690	synchron	unterstützt	nicht unterstützt	unterstützt
GetCursorX, Seite 691	synchron	unterstützt	unterstützt	unterstützt
GetCursorY, Seite 691	synchron	unterstützt	unterstützt	unterstützt
GetDisplayHorizontalResolution, Seite 691	synchron	unterstützt	unterstützt	unterstützt
GetDisplayVerticalResolution, Seite 692	synchron	unterstützt	unterstützt	unterstützt
GetFreeMemoryCE, Seite 692	synchron	nicht unterstützt	unterstützt	unterstützt
GetHardKeyModel, Seite 693	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
GetHardKeySN, Seite 693	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
GetIPAll, Seite 694	synchron	unterstützt	unterstützt	unterstützt
GetMemoryCE, Seite 694	synchron	nicht unterstützt	unterstützt	unterstützt
GetNetMACID, Seite 695	synchron	unterstützt	unterstützt	unterstützt
GetOS, Seite 696	synchron	unterstützt	unterstützt	unterstützt
GetPrivateProfileString, Seite 696	synchron	unterstützt	unterstützt	unterstützt
GetProductPath, Seite 697	synchron	unterstützt	unterstützt	unterstützt
GetRegValue, Seite 697	synchron	unterstützt	unterstützt	nicht unterstützt
GetRegValueType, Seite 698	synchron	unterstützt	unterstützt	nicht unterstützt
GetServerHostName, Seite 699	synchron	nicht unterstützt	nicht unterstützt	unterstützt
GetTickCount, Seite 699	synchron	unterstützt	unterstützt	unterstützt
InfoAppAlrDir, Seite 700	synchron	unterstützt	unterstützt	unterstützt
InfoAppHST, Seite 700	synchron	unterstützt	unterstützt	unterstützt
InfoDiskFree, Seite 700	synchron	unterstützt	nicht unterstützt	unterstützt
InfoResources, Seite 701	synchron	unterstützt	unterstützt	unterstützt
IsActiveXReg, Seite 701	synchron	unterstützt	unterstützt	unterstützt
IsAppChangedOnServer, Seite 702	synchron	unterstützt	unterstützt	unterstützt
NoInputTime, Seite 703	synchron	unterstützt	unterstützt	unterstützt
ProductVersion, Seite 703	synchron	unterstützt	unterstützt	unterstützt

Anhang: WinStudio-interne Skriptsprache

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
RegSaveCE, Seite 704	synchron	nicht unterstützt	unterstützt	nicht unterstützt
ReloadAppFromServer, Seite 704	synchron	unterstützt	unterstützt	unterstützt
SaveAlarmFile, Seite 705	synchron	unterstützt	unterstützt	nicht unterstützt
SetAppAlarmPath, Seite 706	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
SetAppHSTPath, Seite 706	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
SetDateFormat, Seite 707	synchron	unterstützt	unterstützt	unterstützt
SetKeyboardLanguage, Seite 707	synchron	unterstützt	unterstützt	unterstützt
SetRegValue, Seite 708	synchron	unterstützt	unterstützt	nicht unterstützt
SetWebConfig, Seite 709	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
SNMPGet, Seite 711	synchron	unterstützt	nicht unterstützt	unterstützt
SNMPSet, Seite 712	synchron	unterstützt	nicht unterstützt	unterstützt
WritePrivateProfileString, Seite 714	synchron	unterstützt	unterstützt	unterstützt

Tag-Datenbank

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
ExecuteAlarmAck, Seite 715	synchron	unterstützt	unterstützt	unterstützt
ForceTagChange, Seite 716	synchron	unterstützt	unterstützt	unterstützt
GetTagValue, Seite 716	synchron	unterstützt	unterstützt	unterstützt
SetTagValue, Seite 717	synchron	unterstützt	unterstützt	unterstützt

Loop

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
For, Seite 718 & "Next"	entf.	unterstützt	unterstützt	nicht unterstützt

ODBC

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
ODBCBeginTrans, Seite 720	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCBindCol, Seite 721	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCCanAppend, Seite 722	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCCanTransact, Seite 722	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCCanUpdate, Seite 723	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
ODBCClose, Seite 723	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCCommitTrans, Seite 724	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCDelete, Seite 724	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCExecuteSQL, Seite 725	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCInsert, Seite 725	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCIsBOF, Seite 726	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCIsDeleted, Seite 727	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCIsEOF, Seite 727	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCIsFieldNULL, Seite 728	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCIsFieldNullable, Seite 728	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCMove, Seite 729	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCMoveFirst, Seite 729	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCMoveLast, Seite 730	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCMoveNext, Seite 730	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCMovePrev, Seite 731	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCOpen, Seite 732	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCQuery, Seite 733	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCRollback, Seite 733	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCSetFieldNULL, Seite 734	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCSetFilter, Seite 734	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCSetSort, Seite 735	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Anhang: WinStudio-interne Skriptsprache

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
ODBCUnbindCol, Seite 736	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCUpdate, Seite 736	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

E-Mail

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
CnfEmail, Seite 737	synchron	unterstützt	unterstützt	unterstützt
GetStatusSendEmailExt, Seite 740	synchron	unterstützt	unterstützt	unterstützt
SendEmail, Seite 740	synchron	unterstützt	unterstützt	unterstützt
SendEmailExt, Seite 741	asynchron	unterstützt	unterstützt	unterstützt

Einwahl

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
DialError, Seite 744	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt
DialGetClientIP, Seite 750	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt
DialGetServerIP, Seite 751	synchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt
DialStatus, Seite 753	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt
DialUp, Seite 755	asynchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt
DialUpToCE, Seite 756	asynchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt
FindAllDevices, Seite 758	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt
FindModem, Seite 758	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt
HangUp, Seite 759	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt
PhoneDialUp, Seite 759	asynchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt
PhoneDisableListen, Seite 760	synchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt
PhoneEnableListen, Seite 761	synchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt
PhoneHangUp, Seite 761	asynchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt
PhoneStatus, Seite 762	synchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt

Die Einwahlfunktionen werden unter Windows 7 nicht unterstützt.

ActiveX Control und .NET Control

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
XGet, Seite 763	asynchron	unterstützt	nur ActiveX Controls	unterstützt
XRun, Seite 764	asynchron	unterstützt	nur ActiveX Controls	unterstützt
XSet, Seite 765	asynchron	unterstützt	nur ActiveX Controls	unterstützt

Ereignisprotokollierung

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
SendEvent, Seite 766	synchron	unterstützt	unterstützt	unterstützt

FTP

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
CnfFTP, Seite 767	synchron	unterstützt	unterstützt	unterstützt
FTPGet, Seite 768	asynchron	unterstützt	unterstützt	unterstützt
FTPPut, Seite 770	asynchron	unterstützt	unterstützt	unterstützt
FTPStatus, Seite 771	synchron	unterstützt	unterstützt	unterstützt

DB/ERP

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
DBCursorClose, Seite 773	synchron	unterstützt	unterstützt	unterstützt
DBCursorCurrentRow, Seite 776	synchron	unterstützt	unterstützt	unterstützt
DBCursorGetValue, Seite 776	synchron	unterstützt	unterstützt	unterstützt
DBCursorMoveTo, Seite 777	synchron	unterstützt	unterstützt	unterstützt
DBCursorNext, Seite 778	synchron	unterstützt	unterstützt	unterstützt
DBCursorOpen, Seite 779	synchron	unterstützt	unterstützt	unterstützt
DBCursorOpenSQL, Seite 781	synchron	unterstützt	unterstützt	unterstützt
DBCursorPrevious, Seite 783	synchron	unterstützt	unterstützt	unterstützt
DBCursorRowCount, Seite 783	synchron	unterstützt	unterstützt	unterstützt
DBDelete, Seite 784	synchron	unterstützt	unterstützt	unterstützt
DBExecute, Seite 785	synchron	unterstützt	unterstützt	unterstützt
DBInsert, Seite 787	synchron	unterstützt	unterstützt	unterstützt
DBSelect, Seite 788	synchron	unterstützt	unterstützt	unterstützt
DBUpdate, Seite 790	synchron	unterstützt	unterstützt	unterstützt
SyncAlarm, Seite 791	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt
SyncAlarmStatus, Seite 792	synchron	unterstützt	unterstützt	auf dem Server ausgeführt

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
SyncEvent, Seite 793	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt
SyncEventStatus, Seite 793	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
SyncTrend, Seite 794	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt
SyncTrendStatus, Seite 795	synchron	unterstützt	unterstützt	auf dem Server ausgeführt

19.3 Aufbau der Funktionsbeschreibungen

Nachstehend finden Sie eine Erläuterung der Beschreibungen der internen Skriptfunktionen.

Die Funktionsbeschreibungen sind in mehrere Abschnitte unterteilt.

Funktionsattribute

Jede Funktion verfügt über bestimmte Attribute, die jeweils in einer einzeiligen Tabelle beschrieben sind:

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Funktionsname	Gruppenname	synchron oder asynchron	unterstützt oder nicht unterstützt	unterstützt oder nicht unterstützt	unterstützt oder nicht unterstützt

In der ersten Spalte wird der genaue Name der Funktion genannt, wie er in Ihrem Projekt verwendet werden sollte.

Die zweite Spalte enthält den Namen der **Gruppe**, der die Funktion zugeordnet ist. Jede Gruppe fasst Funktionen, anhand der Art der Berechnung oder des Teil des Projekts, auf das sie wirkt, zusammen. Der Gruppenname ist bei der Suche nach einer bestimmten Funktion im Funktionsbrowser oder der Dokumentation hilfreich.

Die Spalte **Ausführung** beschreibt, ob die Funktion synchron oder asynchron ausgeführt wird:

- **Synchron** bedeutet, dass die jeweilige Station bei Ausführung der Funktion auf dem Projektserver oder dem Projekt-Client eine Antwort bzw. Bestätigung von der Gegenstelle benötigt. Die Laufzeitumgebung wartet beim Aufruf einer synchronen Funktion an, bis die Funktion abgearbeitet wurde. Auf diese Weise bleibt die Ausführung auf Server und Client synchron. Bei Funktionen, die sehr schnell ausgeführt werden ist dies unproblematisch. Wenn aber ein Client sehr viele Funktionsaufrufe durchführt oder das Netzwerk langsam ist, kann die Leistungsfähigkeit der Anwendung darunter leiden.
- **Asynchron** bedeutet, dass die Funktion sowohl auf dem Projektserver als auch auf dem Projekt-Client ausgeführt werden kann, ohne dass die aufrufende Funktion auf das Ergebnis wartet. Das Projekt läuft somit ohne Unterbrechung weiter.

Die nachfolgenden Spalten enthalten Informationen, welche Zielsysteme die Funktion unterstützen:

- Die Spalte **Windows** gibt Auskunft darüber, ob die Funktion auf Client und Server Stationen ausgeführt werden können, die auf einer Desktop- oder Server-Vollversion von Microsoft Windows laufen.
- Die Spalte **Embedded** gibt Auskunft darüber, ob die Funktion auf Client und Serverstationen unter WinCE ausgeführt werden können.

- Die Spalte **Client** gibt Auskunft darüber, ob die Funktion auf Client Stationen mit dem Secure Viewer bzw. Web-Browser ausgeführt werden können.

Weitere Informationen zu den Zielsystemen erhalten Sie unter Systemanforderungen, Seite 39.

Syntaxdiagramm und -parameter

Ein Syntaxdiagramm zeigt, wie die Funktion eingegeben werde sollte und welche Parameter sie verwendet.

In den meisten Fällen kann ein Parameter entweder einen konstanten Wert oder eine Variable (Tag) enthalten. Der Datentyp des Parameters wird durch seinen Präfix angezeigt:

- "bool" bedeutet, dass der Parameter einen Booleschen Wert oder eine Boolesche Variable verlangt.

  Zum Beispiel entweder "0" oder "boolTag".
- "num" bedeutet, dass der Parameter einen Zahlenwert oder eine Integer bzw. Real Variable verlangt.

  Zum Beispiel entweder "45,6543" oder "numTag".
- "str" bedeutet, dass der Parameter eine Zeichenkette in Anführungszeichen oder eine Textvariable (String Tag) verlangt.

  Zum Beispiel entweder "„Mein string“" oder "strTag".

Der zusätzliche Präfix "opt" zeigt an, dass ein Parameter optional ist. Wenn Sie keinen Wert für den Parameter festlegen, nimmt die Funktion den in der Parameterbeschreibung angegebenen Standardwert an.

In den Fällen, in denen ein Parameter eine Variable oder eine andere spezielle Eingabe erwartet, wird dies in der Parameterbeschreibung genauer erläutert.

Rückgabewert

In diesem Abschnitt werden die ggf. von der Funktion zurückgegebenen Werte beschrieben.

Dabei kann es sich um ein Berechnungsergebnis oder um einen Fehlercode handeln.

Der Fehlercode signalisiert dem Anwender, ob eine Funktion fehlerfrei ausgeführt wurde oder hilft im Fehlerfall die Ursache des Problems zu ermittelt.

Hinweise

In diesem Abschnitt werden zusätzliche Hinweise oder Warnungen zur Verwendung der Funktion beschrieben.

Beispiele

In diesem Abschnitt finden Sie Kodebeispiele, wie die Funktion aufgerufen werden kann. Anhand verschiedener Beispiele wird gezeigt, wie die Funktion mit konstanten Werten, Variablen und optionalen Parametern aufgerufen werden kann.

19.4 Protokollmeldungsfunktionen

19.4.1 Protokollmeldungsfunktionen – Allgemeines

Diese Funktionen dienen der Anzeige von Status- und Fehlermeldungen im Fenster "Ausgabe" (für lokale Laufzeit) oder im Fenster "Remote LogWin".

19.4.2 Trace

"Trace" ist eine interne Skriptfunktion, die eine Textmeldung im Fenster "Ausgabe" ausgibt. Sie wird üblicherweise zur Nachverfolgung des Programmablaufs verwendet.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Trace	Protokollmeldung	synchron	unterstützt	unterstützt	nicht unterstützt

Syntax

```
Trace(strOutputMessage)
```

Eingaben

strOutputMessage

Der Text der anzuzeigenden Meldung.

Beispiele

Statischen Text anzeigen, der ein bestimmtes Ereignis berichtet

```
Trace("Beginning step 5.")
```

Anzeige eines Datums oder Zeitstempels durch Referenzieren des entsprechenden System-Tags

```
Trace(Date)
```

Verknüpfen von statischem Text, Tag-Referenzen und Funktionsaufrufen zur Ausgabe einer komplexen Meldung

```
Trace("Die aktuelle Sekunde der Minute ist " + Second + " und der Systemtick ist " +
 GetTickCount() + " ms.")
```

19.5 Arithmetische Funktionen

19.5.1 Arithmetische Funktionen – Allgemeines

Diese Funktionen dienen der Durchführung von arithmetischen Operationen und Bit-Manipulationen auf numerischen Werten.

arithmetisch

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
Abs, Seite 559	synchron	unterstützt	unterstützt	unterstützt
Div, Seite 560	synchron	unterstützt	unterstützt	unterstützt
Format, Seite 560	synchron	unterstützt	unterstützt	unterstützt
GetBit, Seite 564	synchron	unterstützt	unterstützt	unterstützt
Mod, Seite 564	synchron	unterstützt	unterstützt	unterstützt
Pow, Seite 565	synchron	unterstützt	unterstützt	unterstützt
ResetBit, Seite 565	synchron	unterstützt	unterstützt	unterstützt
Round, Seite 566	synchron	unterstützt	unterstützt	unterstützt
SetBit, Seite 567	synchron	unterstützt	unterstützt	unterstützt
SQRT, Seite 568	synchron	unterstützt	unterstützt	unterstützt
Swap16, Seite 568	synchron	unterstützt	unterstützt	unterstützt
Swap32, Seite 568	synchron	unterstützt	unterstützt	unterstützt
Trunc, Seite 569	synchron	unterstützt	unterstützt	unterstützt

19.5.2 Arithmetische Operatoren

Die interne Skriptsprache unterstützt die folgenden arithmetischen Operatoren.

Arithmetische Operatoren

Operator	Verwendung	Beschreibung
+	X + Y	addieren (plus)
-	X - Y	subtrahieren (minus)
*	X * Y	multiplizieren mit
/	X / Y	dividieren durch

Arithmetische Operatoren werden von links nach rechts gemäß der Standardrechenreihenfolge aufgelöst. Zum Ändern der Reihenfolge setzen Sie den zuerst aufzulösenden Teil in Klammern. Die folgende Gleichung zum Beispiel hat das Ergebnis 11, weil die Multiplikation vor der Addition durchgeführt wird; die Gleichung multipliziert 2 mit 3 und addiert dann 5 zum Ergebnis hinzu:

5+2*3

Wenn hingegen die Syntax durch Verwendung von Klammern geändert wird, werden 5 und 2 addiert und dann mit 3 multipliziert, was 21 ergibt:

(5+2)*3



Für komplexere mathematische Operationen verwenden Sie die Funktionen Arithmetisch, Seite 558, Statistisch, Seite 569, Logarithmisch, Seite 573 und Trigonometrisch, Seite 599.

19.5.3 Abs

"Abs" ist eine interne Skriptfunktion, mit der der absolute Wert eines bestimmten numerischen Werts ermittelt werden kann.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Abs	arithmetisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Abs(numValue)
```

Eingaben

numValue
Der numerische Wert, aus dem die Funktion den Absolutwert ermittelt.

Rückgabewert

Der absolute Wert des angegebenen numerischen Werts.

Beispiele

Tag-Name	Ausdruck
Tag	ABS("-54.9788") // Returned value = 54.9788
Tag	ABS(numValue) // Returned value = absolute value of the number in the numValue tag.

19.5.4 Div

"Div" ist eine interne Skriptfunktion, mit der der Quotient zweier bestimmter numerischer Werte erzielt wird.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Div	arithmetisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Div( numNumerator, numDenominator )
```

Eingaben

numNumerator

Der Zähler der Division (Dividend).

numDenominator

Der Nenner der Division (Divisor).

Rückgabewert

Diese Funktion gibt den Quotienten nur als ganze Zahl aus. Der Rest wird weggelassen.



Verwenden Sie die Funktion Mod, Seite 564, um den Rest anstelle des Quotienten zu erhalten.

Beispiele

Tag-Name	Ausdruck
numValue	Div(100, 8) // Returns the value 12
numValue	Div(16, 4) // Returns the value 4
numValue	Div(100, 12.5) // Returns the value 8

19.5.5 Format

"Format" ist eine interne Skriptfunktion, die einen numerischen Wert formatiert und ihn als Zeichenkette zurückgibt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Format	arithmetisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Format(strFlag,numValue)
```

Eingaben

strFlag

Eine Beschreibung, wie der gegebene numerische Wert gemäß der Syntax formatiert werden soll

%width.precisionFormat , wobei gilt:

- **width** ist die Mindestanzahl der Zeichen, die von der Funktion ausgegeben werden. Wenn der ausgegeben Wert kürzer ist, wird er entweder mit Leerzeichen (" ") oder mit Nullen ("0") aufgefüllt, siehe Beispiele unten.

  Der Wert wird nicht gekürzt, auch wenn das Ergebnis länger als die angegebene Breite ist.

  Gilt für die Flags **d, x, X, o, b, f, e, E, g, G, s, c** und **h**.
- **.precision** ist die Anzahl der Dezimalstellen für eine Zahl mit Gleitkomma.

  Gilt für die Flags **f, e, E, g** und **G**.
- **F** ist das spezifische Format:

Format	Beschreibung
d	dezimal
x	hexadezimal (Kleinbuchstaben)
X	hexadezimal (Großbuchstaben)
o	oktal
b	Binär
f	Gleitkomma
e	wissenschaftliche Schreibweise (Kleinbuchstabe e)
E	wissenschaftliche Schreibweise (Großbuchstabe E)
g	gerundet (Kleinbuchstabe e, falls zutreffend)
G	gerundet (Großbuchstabe E, falls zutreffend)
s	Zeichenkette
c	ASCII-Zeichen (d.h. der numerische Wert wird als ASCII-Zeichencode interpretiert)
h	Stunden (hh:mm:ss)

Alternativ dazu kann das Format mit der Syntax "##.###" eingestellt werden, wobei der numerische Wert auf die angegebenen Dezimalstellen gerundet wird.

numValue

Der zu formatierende numerische Wert.

Rückgabewert

Diese Funktion gibt einen String aus, der den formatierten numerischen Wert enthält.

☞ "Format" ist ähnlich der Funktion "printf" in anderen Programmiersprachen und lässt die meisten ähnlichen Formatierungsoptionen zu. Allerdings kann "Format" im Gegensatz zu "printf" nur zum Formatieren jeweils eines numerischen Werts verwendet werden.

Diese Funktion ist besonders nützlich beim Formatieren von Werten, die in Berichten ausgegeben werden sollen.

Beispiele

Tag-Name	Ausdruck
Tag	Format("%d", 12.34) // Returned value = "12"
Tag	Format("%04d", 12.34) // Returned value = "0012"
Tag	Format("%4d", 12.34) // Returned value = "12"

Tag-Name	Ausdruck
Tag	Format("%x", 26) // Returned value = "1a"
Tag	Format("%04x", 26) // Returned value = "001a"
Tag	Format("%4x", 26) // Returned value = "1a"

Tag-Name	Ausdruck
Tag	Format("%X", 26) // Returned value = "1A"
Tag	Format("%04X", 26) // Returned value = "001A"
Tag	Format("%4X", 26) // Returned value = "1A"

Tag-Name	Ausdruck
Tag	Format("%o", 16) // Returned value = "20"
Tag	Format("%04o", 16) // Returned value = "0020"
Tag	Format("%4o", 16) // Returned value = "20"

Tag-Name	Ausdruck
Tag	Format("%b", 2) // Returned value = "10"
Tag	Format("%4b", 2) // Returned value = "0010"
Tag	Format("%04b", 2) // Returned value = "0010"

Tag-Name	Ausdruck
Tag	Format("%0.1f", 12.34) // Returned value = "12.3"
Tag	Format("%06.1f", 12.34) // Returned value = "0012.3"
Tag	Format("%6.1f", 12.34) // Returned value = "12.3"

Tag-Name	Ausdruck
Tag	Format("%e", 12.34) // Returned value = "1.234000e+001"
Tag	Format("%0.1e", 12.34) // Returned value = "1.2e+001"

Tag-Name	Ausdruck
Tag	Format("%09.1e", 12.34) // Returned value = "01.2e+001"
Tag	Format("%9.1e", 12.34) // Returned value = " 1.2e+001"

Tag-Name	Ausdruck
Tag	Format("%E", 12.34) // Returned value = "1.234000E+001"
Tag	Format("%0.1E", 12.34) // Returned value = "1.2E+001"
Tag	Format("%09.1E", 12.34) // Returned value = "01.2E+001"
Tag	Format("%9.1E", 12.34) // Returned value = " 1.2E+001"

Tag-Name	Ausdruck
Tag	Format("%0.1g", 12.34) // Returned value = "1e+001"
Tag	Format("%0.2g", 12.34) // Returned value = "12"
Tag	Format("%0.3g", 12.34) // Returned value = "12.3"
Tag	Format("%05.3g", 12.34) // Returned value = "012.3"
Tag	Format("%5.3g", 12.34) // Returned value = " 12.3"

Tag-Name	Ausdruck
Tag	Format("%0.1G", 12.34) // Returned value = "1E+001"
Tag	Format("%0.2G", 12.34) // Returned value = "12"
Tag	Format("%0.3G", 12.34) // Returned value = "12.3"
Tag	Format("%05.3G", 12.34) // Returned value = "012.3"
Tag	Format("%5.3G", 12.34) // Returned value = "12.3"

Tag-Name	Ausdruck
Tag	Format("%s", 12.34) // Returned value = "12"
Tag	Format("%04s", 12.34) // Returned value = "0012"
Tag	Format("%4s", 12.34) // Returned value = "12"

Tag-Name	Ausdruck
Tag	Format("%c", 97) // Returned value = "a"
Tag	Format("%4c", 97) // Returned value = "a"
Tag	Format("%04c", 97) // Returned value = "000a"

Tag-Name	Ausdruck
Tag	Format("%h", 30) // Returned value = "00:00:30"
Tag	Format("%h", 60) // Returned value = "00:01:00"
Tag	Format("%h", 90) // Returned value = "00:01:30"
Tag	Format("%h", 3600) // Returned value = "01:00:00"

Tag-Name	Ausdruck
Tag	Format("##.#", 26.56789) // Returned value = "26.6"
Tag	Format("#.##", 26.56789) // Returned value = "26.57"
Tag	Format("##.##", 26.56789) // Returned value = "26.57"

19.5.6 GetBit

"GetBit" ist eine interne Skriptfunktion, mit der der Wert eines einzelnen Bits in einem numerischen Wert ermittelt wird.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetBit	arithmetisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetBit( tagName, numBitNumber )
```

Eingaben

tagName

Der Name eines Integer-Tags, aus dem der Bit-Wert abgerufen wird.



Um den Namen eines Tags direkt anzugeben, anstatt den Tag-Wert zu verwenden, muss der Tag-Name in Anführungszeichen gesetzt werden.

Zum Beispiel **GetBit("Second", 1)**.

numBitNumber

Ein numerischer Tag oder Wert zur Angabe der Position (0...31) des abzurufenden Bits.

Rückgabewert

Gibt den Wert ("0" oder "1") des angegebenen Bits zurück.



Lesen und Schreiben einzelner Bits in einem Integer Tag ist auch über das "Bit-Feld" (Variablen-Attribut) möglich.

Geben Sie zum Beispiel "Second->b0" ein, um Zugriff auf das LSB (Least Significant Bit des Sekunden-Tags) zu erhalten, und "Second->b31", um Zugriff auf das MSB (Most Significant Bit des Sekunden-Tags) zu erhalten.

Beispiele

Tag-Name	Ausdruck
Tag	GetBit("numSource", 4) // If the tag numSource holds a value of 15, then this function returns 0.
Tag	GetBit("numSource", 1) // If the tag numSource holds a value of 19, then this function returns 1.

19.5.7 Mod

"Mod" ist eine interne Skriptfunktion, die den Rest einer Division ausgibt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Mod	arithmetisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Mod( numNumerator, numDenominator )
```

Eingaben

numNumerator

Integer- oder Real-Tag, welcher den Zähler der Funktion beinhaltet (Dividend)

numDenominator

Integer- oder Real-Tag, welcher den Nenner der Funktion beinhaltet (Divisor)

Rückgabewert Gibt nach Division von "numNumerator" durch "numDenominator" den Rest (als reelle Zahl) aus.



Verwenden Sie die Funktion Div, Seite 560, um den ganzzahligen Quotienten der Operation zu erhalten.

Beispiele

Tag-Name	Ausdruck
Tag	Mod(50, 4) // Returned value = 2
Tag	Mod(16, 4) // Returned value = 0
Tag	Mod(100, 8,2) // Returned value = 1,600

19.5.8 Pow

Pow ist eine interne Skriptfunktion, die die Potenz einer Basis berechnet.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Pow	arithmetisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Pow( numBase, numExponent )
```

Eingaben

numBase

Integer- oder Real-Tag, welcher den Wert der Basis der Funktion enthält.

numExponent

Gibt das Ergebnis der Exponentialfunktion zurück

Rückgabewert Gibt das Ergebnis des Potenzierung der Basis mit dem Exponenten zurück.

Beispiele

Tag-Name	Ausdruck
Tag	Pow(2, 3) // Returned value = 8
Tag	Pow(10, 4) // Returned value = 10000

19.5.9 ResetBit

Setzt einen einzelnen Bit in einem Integer-Tag auf 0 zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ResetBit	arithmetisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ResetBit( tagName, numBitNumber )
```

Eingaben

tagName

Name eines Integer-Tags, in dem der Bit-Wert zurückgesetzt wird.

☞ Um den Namen eines Tags direkt anzugeben, anstatt den Tag-Wert zu verwenden, muss der Tag-Name in Anführungszeichen gesetzt werden.

Zum Beispiel **ResetBit("Second", 1)**.

numBitNumber

Ein numerischer Tag oder Wert zur Angabe der Position (0...31) des zurückzusetzenden Bits.

Rückgabewert

Wert	Beschreibung
0	Kein Fehler
1	Ungültiger Parameter
2	Tag existiert nicht

☞ Lesen und Schreiben einzelner Bits in einem Integer Tag ist auch über das "Bit-Feld" (Variablen-Attribut) möglich.

Geben Sie zum Beispiel **Second->b0** ein, um Zugriff auf das LSB (Least Significant Bit des Sekunden-Tags) zu erhalten, und **Second->b31**, um Zugriff auf das MSB (Most Significant Bit des Sekunden-Tags) zu erhalten.

Beispiele

Tag-Name	Ausdruck
Tag	ResetBit("numSource", 4) // If the tag numSource held a value of 16, then the function returns 0 and numSource holds a new value of 0.
Tag	ResetBit("numSource", 1) // If the tag numSource held the value 19, then the function returns 0 and numSource holds a new value of 17.

19.5.10 Round

Rundet "numValue" auf die nächste ganze Zahl.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Round	arithmetisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Round( numValue )
```

Eingaben

numValue

Ein Real-Tag, der den zu rundenden Wert enthält.

Rückgabewert Gibt das ganzzahlige Ergebnis der Rundungsfunktion aus.

Beispiele

Tag-Name	Ausdruck
Tag	Round("345.87") // Returned value = 346
Tag	Round("65,323") // Returned value = 65

19.5.11 SetBit

Setzt ein einzelnes Bit in einem Integer-Tag auf 1.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetBit	arithmetisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
SetBit( tagName, numBitNumber )
```

Eingaben

tagName

Der Name eines Integer-Tags, in dem der Bit-Wert gesetzt wird.

☞ Um den Namen eines Tags direkt anzugeben, anstatt den Tag-Wert zu verwenden, muss der Tag-Name in Anführungszeichen gesetzt werden.

Zum Beispiel **SetBit("Second", 1)**.

numBitNumber

Ein numerischer Tag oder Wert zur Angabe der Position (0...31) des zu setzenden Bits.

Rückgabewert

Wert	Beschreibung
0	Kein Fehler
1	Ungültiger Parameter
2	Tag existiert nicht

☞ Lesen und Schreiben einzelner Bits in einem Integer Tag ist auch über das "Bit-Feld" (Variablen-Attribut) möglich.

Geben Sie zum Beispiel "Second->b0" ein, um Zugriff auf das LSB (Least Significant Bit des Sekunden-Tags) zu erhalten, und "Second->b31", um Zugriff auf das MSB (Most Significant Bit des Sekunden-Tags) zu erhalten.

Beispiele

Tag-Name	Ausdruck
Tag	SetBit("numSource", 4) // If the tag numSource held a value of 0, then this function returns 0 and numSource holds a new value of 16.
Tag	SetBit("numSource", 1) // If the tag numSource held the value 17, then this function returns 0 and numSource holds a new value of 19.

Anhang: WinStudio-interne Skriptsprache

19.5.12 Sqrt

Berechnet die Quadratwurzel aus "numValue".

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Sqrt	arithmetisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Sqrt( numValue )
```

Eingaben

numValue

Integer- oder Real-Tag, deren Wurzel berechnet werden zoll.

Rückgabewert Gibt die Quadratwurzel der Werts im Tag "numValue" zurück.



Wenn "numValue" einen negativen Wert hat, gibt diese Funktion den Wert 0 aus und setzt die Qualität des zurückgegebenen Tags auf "BAD" (Qualitätsanzeige im Datenbank Spion).

Beispiele

Tag-Name	Ausdruck
Tag	SQRT(25) // Returns the value 5
Tag	SQRT(67) // Returns the value 8.185353

19.5.13 Swap16

Vertauscht die zwei unteren Bytes eines Tags.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Swap16	arithmetisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Swap16( numValue )
```

Eingaben

numValue

Integer-Tag, der den numerischen Wert der zu vertauschenden Bytes enthält.

Rückgabewert Gibt den numerischen Wert nach dem Vertauschen der Bytes aus.

Beispiele

Tag-Name	Ausdruck
Tag	Swap16(16) // 16 = 0000000000010000 in binary. Returned value = 4096 = 0001000000000000 in binary.
Tag	Swap16(43760) // 43760 = 1010010111110000 in binary. Returned value = 61610 = 1111000010100101 in binary.

19.5.14 Swap32

Vertauscht zwei Datenworte eines Tags.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Swap32	arithmetisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Swap32( numValue )
```

Eingaben

numValue
Integer-Tag, der den numerischen Wert der zu vertauschenden Datenworte enthält.

Rückgabewert Gibt den numerischen Wert nach dem Vertauschen der Datenworte aus.

Beispiele

Tag-Name	Ausdruck
Tag	Swap32(16) // 16 = 00000000000000000000000000010000 in binary. Returned value = 1048576 = 00000000000100000000000000000000 in binary.
Tag	Swap32(246333120) // 286333120 = 10101010101010101011111111100000000 in binary. Returned value = -1094709586= 11111111000000001010101010101010 in binary.

19.5.15 Trunc

Liefert den ganzzahligen Anteil eines reellen Zahlenwerts

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Trunc	arithmetisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Trunc( numValue )
```

Eingaben

numValue
Real-Tag, dessen ganzzahliger Anteil berechnet werden soll.

Rückgabewert Gibt den ganzzahligen Anteil des reellen Zahlenwerts von "numValue" aus, d. h. alle Dezimalstellen werden ohne Rundung abgeschnitten.

Beispiele

Tag-Name	Ausdruck
	Trunc(234,987) // Returned value = 234
	Trunc(-3465.9) // Returned value = -3465

19.6 Statistische Funktionen

19.6.1 Statistische Funktionen – Allgemeines

Diese Funktionen dienen der Berechnung bestimmter statistischer Werte – wie Durchschnitt, Maximum und Minimum – aus zwei oder mehr numerischen Werten.

Statistisch

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
Avg, Seite 570	synchron	unterstützt	unterstützt	unterstützt
Max, Seite 571	synchron	unterstützt	unterstützt	unterstützt

Anhang: WinStudio-interne Skriptsprache

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
Min, Seite 571	synchron	unterstützt	unterstützt	unterstützt
Rand, Seite 572	synchron	unterstützt	unterstützt	unterstützt

19.6.2 Avg

Gibt den Durchschnittswert einer Reihe von Zahlen aus.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Avg	Statistisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Avg( numValue1, numValue2, … , numValueN )
Avg( "tagArray", numSample, optNumIgnore )
```

Diese Funktion hat zwei Formate:

- Wenn der erste Parameter ein numerischer Tag oder Wert ist, wird das Format

  Avg(numValue1, numValue2, … , numValueN) verwendet.
- Wenn der erste Parameter ein Array-Tag in Anführungszeichen oder ein String-Tag ist, wird das Format

  Avg("tagArray", numSample, optNumIgnore) verwendet.

Eingaben

numValue (1…N)

Integer- oder Real-Tags, die die Zahlen für die Durchschnittsbildung enthalten.

tagArray

Name eines Array-Tags (Real oder Integer), der die Zahlen für die Durchschnittsbildung enthält.

numSample

Die Anzahl der Array-Elemente für die Durchschnittsbildung.

optNumIgnore

Optionaler Integer- oder Real-Tag mit dem Wert, der bei der Durchschnittsbildung ignoriert werden soll.

Rückgabewert

Gibt den Durchschnitt der Werte aus.

Beispiele

Tag-Name	Ausdruck
Tag	Avg(1, 2.34, 5, 7, 4, 8, 9.4) // Returned value = 5.248571
Tag	Avg(1, 5, -9, 0, 5, 3) // Returned value = 0.833333

Tag-Name	Ausdruck
Tag	Avg("tagArray[1]", 3) // If tagArray[1]=10, tagArray[2]=20 and tagArray[3]=60, then the Returned Value = 30
Tag	Avg("tagArray[1]", 3, 10) // If tagArray[1]=10, tagArray[2]=20 and tagArray[3]=60, then the Returned Value = 40

19.6.3 Max

Gibt den Maximalwert einer Reihe von Zahlen aus.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Max	Statistisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Max( numValue1, numValue2, … , numValueN )
Max( "tagArray", numSample, optNumIgnore )
```



Diese Funktion hat zwei Formate:

- Wenn der erste Parameter ein numerischer Tag oder Wert ist, wird das Format

  Max(numValue1, numValue2, … , numValueN) verwendet.
- Wenn der erste Parameter ein Array-Tag in Anführungszeichen oder ein String-Tag ist, wird das Format **Max("tagArray", numSample, optNumIgnore)** verwendet.

Eingaben

numValue (1…N)

Integer- oder Real-Tags, die die zu analysierenden Zahlen enthalten.

tagArray

Name eines Array-Tags (Real oder Integer), der die zu analysierenden Zahlen enthält.

numSample

Die Anzahl der zu analysierenden Arra-Elemente.

optNumIgnore

Optionaler Integer- oder Real-Tag mit dem Wert, der bei der Analyse ignoriert werden soll.

Rückgabewert

Gibt den Maximalwert der Reihe aus.

Beispiele

Tag-Name	Ausdruck
Tag	Max(1, 2.34, 5, 7, 4, 8, 9.4) // Returned value = 9.4
Tag	Max(1, 5, -9, 0, 5, 3) // Returned value = 5
Tag	Max("tagArray[1]", 3) // If tagArray[1]=10, tagArray[2]=20 and tagArray[3]=60, then the Returned Value = 60
Tag	Max("tagArray[1]", 3, 10) // If tagArray[1]=10, tagArray[2]=20 and tagArray[3]=60, then the Returned Value = 60

19.6.4 Min

Gibt den Minimalwert einer Reihe von Zahlen aus.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Min	Statistisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Min( numValue1, numValue2, … , numValueN )
Min( "tagArray", numSample, optNumIgnore )
```

☞ Diese Funktion hat zwei Formate:

- Wenn der erste Parameter ein numerischer Tag oder Wert ist, wird das Format

  Min(numValue1, numValue2, … , numValueN) verwendet.
- Wenn der erste Parameter ein Array-Tag in Anführungszeichen oder ein String-Tag ist, wird das Format

  Min("tagArray", numSample, optNumIgnore) verwendet.

Eingaben

numValue (1…N)

Integer- oder Real-Tags, die die zu analysierenden Zahlen enthalten.

tagArray

Name eines Array-Tags (Real oder Integer), der die zu analysierenden Zahlen enthält.

numSample

Die Anzahl der zu analysierenden Array-Elemente.

optNumIgnore

Optionaler Integer- oder Real-Tag mit einem Wert, der bei der Analyse ignoriert werden soll.

Rückgabewert Gibt den Minimalwert der Reihe aus.

Beispiele

Tag-Name	Ausdruck
Tag	Min(1, 2.34, 5, 7, 4, 8, 9.4) // Returned value = 1
Tag	Min(1, 5, -9, 0, 5, 3) // Returned value = -9
Tag	Min("tagArray[1]", 3) // If tagArray[1]=10, tagArray[2]=20 and tagArray[3]=60, then the Returned Value = 10
Tag	Min("tagArray[1]", 3, 10) // If tagArray[1]=10, tagArray[2]=20 and tagArray[3]=60, then the Returned Value = 20

19.6.5 Rand

Erzeugt eine Zufallszahl zwischen 0 und 1.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Rand	Statistisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Rand()
```

Diese Funktion hat keine Parameter.

Rückgabewert Gibt eine reelle Zahl zwischen 0 und 1 aus.

Beispiele

Tag-Name	Ausdruck
Tag	Rand() // Returned value = ?, Where: 0<?<1

19.7 Logarithmische Funktionen

19.7.1 Logarithmische Funktionen – Allgemeines

Diese Funktionen werden zum Ausführen von logarithmischen Operationen auf numerischen Werten verwendet.

Logarithmisch

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
Exp, Seite 573	synchron	unterstützt	unterstützt	unterstützt
Log, Seite 573	synchron	unterstützt	unterstützt	unterstützt
Log10, Seite 574	synchron	unterstützt	unterstützt	unterstützt

19.7.2 Exp

Errechnet den Wert der Eulerschen Zahl "e" (2,718282) potenziert mit der Zahl von "numValue".

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Exp	Logarithmisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Exp(numValue)
```

Eingaben

numValue

Integer- oder Real-Tag, welcher den Wert des Exponenten dieser Funktion beinhaltet..

Rückgabewert

Gibt den Wert von "e^(numValue)" aus.

Beispiele

Tag-Name	Ausdruck
Tag	Exp(1) // Returned value = 2,718282
Tag	Exp(5.25896) // Returned value = 192.281415

19.7.3 Log

Errechnet den natürlichen Logarithmus von "numValue".

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Log	Logarithmisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Log(numValue)
```

Eingaben

numValue

Integer- oder Real-Tag, aus dem der natürliche Logarithmus berechnet wird.

Rückgabewert

Gibt den Wert von "ln(numValue)" aus.



Wenn "numValue" einen negativen Wert hat, gibt diese Funktion den Wert 0 aus und setzt die Qualität des zurückgegebenen Tags auf "BAD" (Qualitätsanzeige im Datenbank Spion).

Beispiele

Tag-Name	Ausdruck
Tag	Log(2,718282) // Returned value = 1
Tag	Log(100) // Returned value = 4.605170

19.7.4 Log10

Errechnet den Logarithmus von "numValue" zur Basis 10.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Log10	Logarithmisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Log10(numValue)
```

Eingaben

numValue

Integer- oder Real-Tag, aus dem der Logarithmus zur Basis 10 berechnet wird.

Rückgabewert

Gibt den Wert von "ln(numValue)" aus.



Wenn "numValue" einen negativen Wert hat, gibt diese Funktion den Wert 0 aus und setzt die Qualität des zurückgegebenen Tags auf "BAD" (Qualitätsanzeige im Datenbank Spion).

Beispiele

Tag-Name	Ausdruck
Tag	Log10(1000) // Returned value = 3
Tag	Log10(43.05) // Returned value = 1.633973

19.8 Logische Funktionen

19.8.1 Logische Funktionen – Allgemeines

Diese Funktionen dienen der Durchführung von logischen Operationen (z.B. if/then, true/false) an Tags und Ausdrücken.

Logisch

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
False, Seite 576	synchron	unterstützt	unterstützt	unterstützt
If, Seite 576	synchron	unterstützt	unterstützt	unterstützt
Toggle, Seite 577	synchron	unterstützt	unterstützt	unterstützt
True, Seite 577	synchron	unterstützt	unterstützt	unterstützt

19.8.2 Logische Operatoren

Die interne Skriptsprache unterstützt die folgenden logischen Operatoren.

Logische Operatoren

Operator	Verwendung	Beschreibung
AND	A AND B	WAHR wenn sowohl A und B WAHR sind
OR	A OR B	WAHR wenn A WAHR ist oder B WAHR ist oder beides
XOR	A XOR B	WAHR wenn A WAHR ist oder B WAHR ist aber nicht beides
NOT	NOT A	WAHR wenn A FALSCH ist
=	X = Y	WAHR wenn X gleich Y ist
>	X > Y	WAHR wenn X größer als Y ist
>=	X >= Y	WAHR wenn X größer oder gleich Y ist
<	X > Y	WAHR wenn X kleiner als Y ist
<=	X <= Y	WAHR wenn X kleiner oder gleich Y ist
<>	X <> Y	WAHR wenn X ungleich Y ist

&	X & Y	Bitweise AND:
		`0101 (dezimal 5)`
		`AND 0011 (dezimal 3)`
		`= 0001 (dezimal 1)`

|	X | Y	Bitweise OR:
		`0101 (dezimal 5)`
		`OR 0011 (dezimal 3)`
		`= 0111 (dezimal 7)`

^	X ^ Y	Bitweise XOR:
		`0101 (dezimal 5)`
		`XOR 0011 (dezimal 3)`
		`= 0110 (dezimal 6)`

~	~ X	Bitweise NOT:
		`NOT 0101 (dezimal 5)`
		`= 1010 (dezimal 10)`

>> n	X >> Y	n Bits nach rechts drehen:
		`0110 (dezimal 6)`
		`ROTATE RIGHT`
		`= 0011 (dezimal 3)`

<< n	X << Y	n Bits nach links drehen:
		`0110 (dezimal 6)`
		`ROTATE LEFT`
		`= 1100 (dezimal 12)`



Für komplexere logische Operationen verwenden Sie die Logischen Funktionen, Seite 574 und Loop-Funktionen, Seite 718.

Anhang: WinStudio-interne Skriptsprache

19.8.3 False

Bestimmt, ob der angegebene Tag bzw. Ausdruck logisch falsch ist.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
False	Logisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
False(TagOrExpression)
```

Eingaben

TagOrExpression

In dieser Funktion zu verwendender Tag oder Ausdruck.

Rückgabewert

0	Wenn der Tag oder Ausdruck "nicht" logisch falsch ist.
1	Wenn der Tag oder Ausdruck logisch falsch ist.



Diese Funktion kann hilfreich sein, wenn Sie einen tatsächlichen Wert 0 ausgeben müssen, wenn der Ausdruck einen Wert ungleich 0 ausgibt.

Beispiele

Tag-Name	Ausdruck
Tag	False(1) // Returned value = 0
Tag	False(5 < 2) // Returned value = 1

19.8.4 If

Bestimmt, ob der Inhalt von "numExpression" logisch wahr ist und gibt dann den entsprechenden Wert von "numThen" oder "optNumElse" aus.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
If	Statistisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
If( numExpression, numThen, optNumElse )
```

Eingaben

numExpression

Als Bedingung in dieser Funktion verwendeter Tag oder Ausdruck.

numThen

Verwendeter Tag oder Ausdruck, wenn die Bedingung logisch wahr ist.

optNumElse

Verwendeter Tag oder Ausdruck, wenn die Bedingung logisch falsch ist.

Rückgabewert

numThen	Wenn numExpression logisch wahr ist.
optNumElse	Wenn numExpression logisch falsch ist.
Kein Wert ausgegeben	Wenn numExpression logisch falsch ist und kein optNumElse in der Funktion enthalten ist.

☞ Der Parameter "numExpression" kann eine Kombination aus logischen Operationen ("AND", "OR" und "NOT") sein.

Zum Beispiel **If(TagA>TagB AND TagA=10,1,0)**.

Der Parameter "numThen" kann eine andere Funktion enthalten, die wiederum die Funktion "If()" verwendet. Daher können "If()"-Funktionen verschachtelt verwendet werden.

Zum Beispiel **if(TagA>TagB,If(TagA<TagC,1, 2),3)**.

Beispiele

Tag-Name	Ausdruck
Tag	If(5 > 4,10, 6) // Returned value = 10
Tag	If(5 < 2, 0, 2) // Returned value = 0
Tag	If(3 = 9, 67) // No Returned value. (Tag retains previous value.)

19.8.5 Toggle

Gibt den umgeschalteten Wert aus dem Inhalt des Tags "numValue" zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Toggle	Logisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Toggle( numValue )
```

Eingaben

numValue

Boolean-Tag mit dem umzuschaltenden Wert.

Rückgabewert Numerisches Ergebnis (0 oder 1) des umzuschaltenden Werts.

☞ Diese Funktion ändert nicht tatsächlich den Wert das Tags. Sie können die Funktion jedoch in einem Ausdruck oder einem Kommando verwenden, das den Wert des Tags ändert.

Beispiele

Tag-Name	Ausdruck
Tag	Toggle(MyBoolTag) // Returned value = 1 if MyBoolTag value equals 0, or 0 if MyBoolTag value equals 1
Tag	Toggle(numValue) // Returned value = toggled value of the number in the numValue tag

19.8.6 True

Bestimmt, ob der angegebene Tag bzw. Ausdruck logisch wahr ist.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
True	Logisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
True(TagOrExpression)
```

Eingaben

TagOrExpression

In dieser Funktion zu verwendender Tag oder Ausdruck.

Rückgabewert

0	Wenn der Tag oder Ausdruck nicht logisch wahr ist.
1	Wenn der Tag oder Ausdruck logisch wahr ist.



Diese Funktion kann hilfreich sein, wenn Sie einen tatsächlichen Wert 1 ausgeben müssen, wenn der Ausdruck einen Wert ungleich 0 ausgibt.

Beispiele

Tag-Name	Ausdruck
Tag	True(1) // Returned value = 1
Tag	True(5 < 2) // Returned value = 0

19.9 String-Funktionen

19.9.1 Stringfunktionen – Allgemeines

Diese Funktionen dienen zum Verändern von Zeichenketten oder deren Konvertierung in numerische Werte.

Zeichenkette

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
Asc2Str, Seite 579	synchron	unterstützt	unterstützt	unterstützt
CharToValue, Seite 579	synchron	unterstützt	unterstützt	unterstützt
CharToValueW, Seite 580	synchron	unterstützt	unterstützt	unterstützt
ClassMembersToStrVector, Seite 581	synchron	unterstützt	unterstützt	unterstützt
NCopy, Seite 582	synchron	unterstützt	unterstützt	unterstützt
Num, Seite 583	synchron	unterstützt	unterstützt	unterstützt
Str, Seite 583	synchron	unterstützt	unterstützt	unterstützt
Str2Asc, Seite 583	synchron	unterstützt	unterstützt	unterstützt
StrCompare, Seite 584	synchron	unterstützt	unterstützt	unterstützt
StrCompareNoCase, Seite 585	synchron	unterstützt	unterstützt	unterstützt
StrFromInt, Seite 585	synchron	unterstützt	unterstützt	unterstützt
StrFromReal, Seite 586	synchron	unterstützt	unterstützt	unterstützt
StrFromTime, Seite 587	synchron	unterstützt	unterstützt	unterstützt
StrGetElement, Seite 588	synchron	unterstützt	unterstützt	unterstützt

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
StrLeft, Seite 588	synchron	unterstützt	unterstützt	unterstützt
StrLen, Seite 589	synchron	unterstützt	unterstützt	unterstützt
StrLower, Seite 589	synchron	unterstützt	unterstützt	unterstützt
StrRChr, Seite 589	synchron	unterstützt	unterstützt	unterstützt
StrRight, Seite 590	synchron	unterstützt	unterstützt	unterstützt
StrSetElement, Seite 590	synchron	unterstützt	unterstützt	unterstützt
StrStr, Seite 591	synchron	unterstützt	unterstützt	unterstützt
StrStrPos, Seite 591	synchron	unterstützt	unterstützt	unterstützt
StrTrim, Seite 592	synchron	unterstützt	unterstützt	unterstützt
StrTrimAll, Seite 593	synchron	unterstützt	unterstützt	unterstützt
StrUpper, Seite 593	synchron	unterstützt	unterstützt	unterstützt
ValueToChar, Seite 594	synchron	unterstützt	unterstützt	unterstützt
ValueWToChar, Seite 594	synchron	unterstützt	unterstützt	unterstützt

19.9.2 Asc2Str

Diese Funktion konvertiert eine oder mehrere Unicode-Zeichencodes in eine Zeichenkette.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Asc2Str	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Asc2Str( numChar1, numChar2, … , numCharN )
```

Eingaben

numChar (1-N)

Ein Unicode-Zeichencode (dezimal).

Rückgabewert

Gibt eine Zeichenkette mit den konvertierten Codes zurück.

☞ Obwohl der Name der Funktion anzeigt, dass sie nur ASCII-Zeichen unterstützt, kann sie für gesamten Unicode-Zeichensatz verwendet werden.

Beispiele

Tag-Name	Ausdruck
Tag	Asc2Str(65) // Returned value = "A"
Tag	Asc2Str(65, 66, 67) // Returned value = "ABC"
Tag	Asc2Str(Array[0], Array[1], Array[2]) // Returned value = "ABC"

19.9.3 CharToValue

Diese Funktion konvertiert eine Zeichenkette in Unicode-Zeichencodes und speichert diese Werte dann in einem Integer-Array.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
CharToValue	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
CharToValue("tagString","tagArray")
```

Eingaben

tagString

Der Name des String-Tags mit dem zu konvertierenden Wert.

tagArray

Der Name des Integer-Arrays, der die konvertierten Werte erhält. Wenn kein Array angegeben wird, ist der Standardwert 0.

☞ Die Variablennamen für tagString und tagArray müssenm wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, die Werte des angegebenen Tags abzurufen.

Rückgabewert Gibt die Anzahl der verwendeten Array-Elemente an, die gleich der Zeichenanzahl in der Zeichenkette sein sollte.

Beispiele Wenn StrTag = "ABC", dann Array[0] = 65, Array[1] = 66 und Array[2] = 67:

CharToValue("StrTag", "Array")

Wenn StrTag = "ABC", dann Array[10] = 65, Array[11] = 66 und Array[12] = 67:

CharToValue("StrTag", "Array[10]")

19.9.4 CharToValueW

Diese Funktion konvertiert eine Zeichenkette in Unicode-Zeichencodes, kombiniert zwei Codes zu einem Doppelbyte-Wort und speichert dann diese Werte in einem Integer-Array.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
CharToValueW	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
CharToValueW("tagString","tagArray")
```

Eingaben

tagString

Der Name des String-Tags mit dem zu konvertierenden Wert.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

tagArray

Der Name des Integer-Arrays, das die konvertierten Werte erhält. Wenn kein Array angegeben wird, ist der Standardwert 0.

Die Variablennamen für tagString und tagArray müssen wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Werte der angegebenen Tags abzurufen.

Rückgabewert

Gibt die Anzahl der verwendeten Array-Elemente an, die gleich der "Hälfte" der Zeichenanzahl in der Zeichenkette sein sollte.

Aufgrund der Art der Kombination von zwei Zeichencodes zu einem einzigen Wert unterstützt diese Funktion nur die Unicode-Zeichen 0 bis 255. Für Zeichencodes über 255 oder wenn Doppel-Byte-Worte nicht erforderlich sind, wird die Funktion CharToValue, Seite 579 verwendet.

Beispiele

Wenn StrTag = "Studio", dann Array[0] = 29779 ("St"), Array[1] = 25717 ("ud") und Array[2] = 28521 ("io"):

CharToValue("StrTag", "Array")

Wenn StrTag = "Studio", dann Array[10] = 29779 ("St"), Array[11] = 25717 ("ud") und Array[12] = 28521 ("io"):

CharToValue("StrTag", "Array[10]")

19.9.5 ClassMembersToStrVector

Überträgt Werte von einem Klassen-Tag in einen Array-Tag.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ClassMembers-ToStrVector	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ClassMembersToStrVector( "strClassTag" , numStartPos, numNumPos, "strArrayTag", optBooStartPosTarget )
```

Eingaben

strClassTag

String-Wert mit dem Klassen-Tag-Namen.

numStartPos

Startposition (Array-Index) von "strClassTag".

numNumPos

Anzahl der Positionen (Array-Indizes) zum Übertragen aus "strClassTag".

strArrayTag

String-Wert mit dem Array-Tag, das die Werte aus "strClassTag" enthält.

optBooStartPosTarget

Startposition (Array-Index) von "strArrayTag". Wird dies ausgelassen, wird der Standardwert 1 verwendet.

Rückgabewert

−6	Array-Größe von "strClassTag" ist nicht groß genug für "numStartPos"
−5	"strClassTag" ist kein Klassen-Tag

−4	"strClassTag" wurde nicht gefunden
−3	"strArrayTag" wurde nicht gefunden
−2	Ungültiger Datentyp der Parameter
−1	Ungültige Anzahl Parameter
0	Erfolgreich übertragen

☞ Wenn "strClassTag" mehr als ein Element besitzt, wird der Wert jedes Klassenelements an "strArrrayTag" übertragen. Daher muss sichergestellt werden, dass die Array-Größe von "strArray-Tag" groß genug ist, um alle Werte aus "strClassTag" aufzunehmen.

Beispiele

Tag-Name	Ausdruck
Tag	ClassMembersToStrVector("Classtag", 5, 3, "Arraytag")
Tag	ClassMembersToStrVector("Classtag", 5, 3, "Arraytag", 0)
Tag	ClassMembersToStrVector (TagName, 0, 1, ArrayName)

19.9.6 NCopy

Kopiert einen bestimmten Abschnitt einer längeren Zeichenkette.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
NCopy	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
NCopy( strSource, numStartChar, numQtdChar )
```

Eingaben

strSource

Der Quellstring.

numStartChar

Integer-Tag mit einer Zahl, die dem ersten zu kopierenden Zeichen entspricht.

numQtdChar

Die Anzahl der zu kopierenden Zeichen.

Rückgabewert Gibt eine Zeichenkette zurück, die Teil des Quellstrings ist (wie von der Funktion definiert).

Beispiele

Tag-Name	Ausdruck
Tag	NCopy("Studio version 7.0", 7, 7) // Returned value = "version"
Tag	NCopy("Technical Reference", 0, 9) // Returned value = "Technical"

☞ Dem ersten Zeichen in der Zeichenkette wird der Wert 0 zugewiesen.

19.9.7 Num

Konvertiert eine Zeichenkette in eine Gleitkommazahl.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Num	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Num( strValue )
```

Eingaben

strValue

Die Anzahl der in das Gleitkommazahlenformat zu konvertierenden Zeichen.

Rückgabewert Gibt die Zahl (vorher im Zeichenkettenformat) als Gleitkommazahl zurück.

Beispiele

Tag-Name	Ausdruck
Tag	Num("321654.987") // Returned value = 321654.987
Tag	Num("5.6589626246") // Returned value = 5.6589626246

☞ Die Zeichenkette für die Gleitkommazahl darf nur die Zahlen (0..9) und den Dezimalpunkt (.) enthalten, andernfalls wird der Wert 0.0 ausgegeben.

19.9.8 Str

Konvertiert eine Zahl in eine Zeichenkette.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Str	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Str( numValue )
```

Eingaben

numValue

Ganzzahl- oder Gleitkommazahl-Tag mit einer Zahl, die in eine Zeichenkette umgewandelt werden soll.

Rückgabewert Gibt die Zeichenkette im Gleitkommazahlenformat zurück.

Beispiele

Tag-Name	Ausdruck
Tag	Str(321654,987) // Returned value = "321654,987"
Tag	Str(5.65896246) // Returned value = "5.658962"

19.9.9 Str2Asc

Diese Funktion konvertiert ein Zeichen in dessen ursprünglichen Unicode-Zeichencode zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Str2Asc	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Str2Asc(strChar)
```

Eingaben

strChar

Das zu konvertierende Zeichen.

Rückgabewert

Gibt den Unicode-Zeichencode (dezimal) für das angegebene Zeichen zurück.

☞ Obwohl die Bezeichnung dieser Funktion anzeigt, dass sie nur ASCII-Zeichen unterstützt, kann sie für den gesamten Unicode-Zeichensatz verwendet werden.

Beispiele

Tag-Name	Ausdruck
Tag	Str2Asc("C") // Returned value = 67
Tag	Str2Asc("o") // Returned value = 111

19.9.10 StrCompare

Vergleicht zwei Zeichenketten und prüft, ob sie identisch sind.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrCompare	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrCompare( strValue1, strValue2 )
```

Eingaben

strValue1

Eine Zeichenkette oder ein Tag vom Typ String, Seite 124. Dies ist die erste zu vergleichende Zeichenkette.

strValue2

Eine Zeichenkette oder ein Tag vom Typ String, Seite 124. Dies ist die zweite zu vergleichende Zeichenkette.

Rückgabewert

−1	Der Wert von ""strValue1"" ist kleiner als der Wert von ""strValue2"".
0	""strValue1"" und ""strValue2"" sind identisch.
1	Der Wert von ""strValue1"" ist größer als der Wert von ""strValue2"".

Beispiele

Tag-Name	Ausdruck
Tag	StrCompare("Text1", "Text2") // Returned value = −1
Tag1="Text" Tag2="Text"	StrCompare(Tag1, Tag2) // Returned value = 0
Tag1="Text1" Tag2="Text2"	StrCompare(Tag1, Tag2) // Returned value = −1

19.9.11 StrCompareNoCase

Vergleicht zwei Zeichenketten und prüft, ob sie identisch sind. Dabei werden Groß- und Kleinschreibung ignoriert (d.h. der Kleinbuchstabe "a" hat den gleichen Wert wie der Großbuchstabe "A").

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrCompareNo-Case	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrCompareNoCase( strValue1, strValue2 )
```

Eingaben

strValue1

Eine Zeichenkette oder ein Tag vom Typ String, Seite 124. Dies ist die erste zu vergleichende Zeichenkette.

strValue2

Eine Zeichenkette oder ein Tag vom Typ String, Seite 124. Dies ist die zweite zu vergleichende Zeichenkette.

Rückgabewert

−1	Der Wert von strValue1 ist kleiner als der Wert von strValue2.
0	strValue1 und strValue2 sind identisch.
1	Der Wert von strValue1 ist größer als der Wert von strValue2.

Beispiele

Tag-Name	Ausdruck
Tag	StrCompareNoCase("Text1", "TEXT1") // Returned value = "0"
Tag1="Text1" Tag2="TEXT1"	StrCompareNoCase(Tag1, Tag2) // Returned value = "0"

19.9.12 StrFromInt

Konvertiert eine Ganzzahl in die entsprechende Zeichenkette in einem anderen Zahlensystem, z.B. binär (Basis 2) oder Oktal (Basis 8).

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrFromInt	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrFromInt( numValue, numBase )
```

Eingaben

numValue

Der in eine Zeichenkette zu konvertierende numerische Wert.

numBase

Das Zahlensystem, in das konvertiert werden soll.

Rückgabewert

Diese Funktion gibt eine Zeichenkette für die gegebene Ganzzahl im angegebenen Zahlensystem aus. Der zurückgegebene Wert kann in jedem Tag vom Typ String gespeichert werden.



Anstelle einer Ganzzahl kann eine reelle Zahl angegeben werden, es wird aber nur der ganzzahlige Anteil der Zahl konvertiert. Zum Konvertieren der gesamten reellen Zahl verwenden Sie stattdessen die Funktion StrFromReal, Seite 586.

Wenn die Basis nicht verändert werden soll, verwenden Sie die Funktion Str, Seite 583.

Beispiele

Tag-Name	Ausdruck
Tag	"StrFromInt(26, 2)" // Returned value = ""11010""
Tag	"StrFromInt(26, 8)" // Returned value = ""32""

19.9.13 StrFromReal

Konvertiert einen reellen numerischen Wert in eine Zeichenkette, entweder als Gleitkommazahl oder in exponentieller Schreibweise.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrFromReal	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrFromReal(numValue,numPrecision,{ strType | f | e | E })
```

Eingaben

numValue

Der zu konvertierende numerische Wert.

numPrecision

Die Anzahl der in der resultierenden Zeichenkette anzuzeigenden Dezimalstellen. Bitte beachten Sie, dass der Wert gerundet und nicht abgeschnitten wird.

strType

Ein aus einem einzelnen Zeichen bestehender Code, der angibt, wie die resultierende Zeichenkette formatiert werden soll, wie in der folgenden Tabelle beschrieben:

Wert von strType	Beschreibung
f	Als Gleitkommazahl formatiert.
e	In exponentieller Schreibweise mit dem Kleinbuchstaben "e" formatiert.
E	In exponentieller Schreibweise mit dem Großbuchstaben "E" formatiert.

Rückgabewert

Diese Funktion gibt eine Zeichenkette als Darstellung des gegebenen numerischen Werts mit der angegebenen Genauigkeit und Schreibweise zurück.

Beispiele

StrFromReal(263.355, 2, "f")

...gibt einen Zeichenkettenwert "263.36" zurück.

StrFromReal(263.355, 2, "e")

...gibt einen Zeichenkettenwert "2.63e+002" zurück.

StrFromReal(263.355, 2, "E")

...gibt einen Zeichenkettenwert "2.63E+002" zurück.

19.9.14 StrFromTime

Konvertiert einen Zeitstempel aus der UTC-Standardschreibweise in eine formatierte Zeichenkette, angepasst an die Einstellungen der Zeitzone in der Systemsteuerung des lokalen Rechners.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrFromTime	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrFromTime( numUTCTime, numType )
```

Eingaben

numUTCTime

Eine Ganzzahl oder ein Tag vom Typ Integer, Seite 124. Ein Zeitstempel in UTC-Standardschreibweise.

numType

Eine Ganzzahl oder ein Tag vom Typ Integer, Seite 124. Legt das Format der resultierenden Zeichenkette fest, wie in der folgenden Tabelle beschrieben:

Wert von numType	Beschreibung
1	Zeigt das "Datum" im gleichen Format wie in der Systemsteuerung des lokalen Rechners gewählt an.
2	Zeigt die "Uhrzeit" im gleichen Format wie in der Systemsteuerung des lokalen Rechners gewählt an.
3	Zeigt eine Standard-24-Zeichenkette mit Datum und Uhrzeit an.
4	Zeigt die Abkürzung für den Wochentag an.
5	Zeigt den vollständigen Wochentag an.

Rückgabewert

Diese Funktion gibt eine Zeichenkette als Darstellung des gegebenen Zeitstempels im festgelegten Format an. Der zurückgegebene Wert kann in jedem Tag vom Typ String, Seite 124 gespeichert werden.



Der Standard Coordinated Universal Time (UTC) zählt die seit 0:00 Uhr WEZ am 1. Januar 1970 abgelaufenen Sekunden. Jeder Tag besteht aus 86.400 Sekunden.

Beispiele

Die untenstehenden Beispiele gelten für einen auf Eastern Standard Time (oder UTC −05:00) eingestellten Rechner.

Tag-Name	Ausdruck
Tag	StrFromTime(86400, 1) // Returned value = "1/1/70"
Tag	StrFromTime(86400, 2) // Returned value = "07:00:00 PM"
Tag	StrFromTime(86400, 3) // Returned value = "Thu Jan 01 19:00:00 1970"

Tag-Name	Ausdruck
Tag	StrFromTime(86400, 4) // Returned value = "Thu"
Tag	StrFromTime(86400, 5) // Returned value = "Thursday"

19.9.15 StrGetElement

Ruft ein bestimmtes Element aus einer Zeichenkette ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrGetElement	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrGetElement( strSource, strDelimiter, numElementNumber )
```

Eingaben

strSource

Der Quellstring.

strDelimiter

Als Trennzeichen zwischen den Elementen verwendetes Zeichen.

numElementNumber

Anzahl der Elemente, die von der Funktion zurückgegeben werden. Das erste Element hat die Nummer 1. Das zweite Element hat die Nummer 2 usw.

Rückgabewert

Gibt das aus "strSource" erzielte Element (String-Wert) zurück.

Beispiele

Tag-Name	Ausdruck
Tag	StrGetElement("a|b|c", "|", 2) // returned value = "b"
Tag	StrGetElement("a,b,c", ",", 3) // returned value = "c"

19.9.16 StrLeft

Kopiert die ersten Zeichen einer längeren Zeichenkette.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrLeft	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrLeft( strSource, numQtdChar )
```

Eingaben

strSource

Der Quellstring.

numQtdChar

Die Anzahl der zu kopierenden Zeichen.

Rückgabewert

Gibt eine Zeichenkette mit den am weitesten links stehenden Zeichen im Quellstring an.

Beispiele

Tag-Name	Ausdruck
Tag	StrLeft("Studio version 7.0", 8) // Returned value = Studio v
Tag	StrLeft ("Technical Reference", 9) // Returned value = Technical

19.9.17 StrLen

Bestimmt die Länge einer Zeichenkette.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrLen	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrLen( strSource )
```

Eingaben

strSource

Die Zeichenkette.

Rückgabewert Gibt eine ganze Zahl zurück, die der Anzahl der Zeichen in der Zeichenkette entspricht.

Beispiele

Tag-Name	Ausdruck
Tag	StrLen("Studio version 7.0") // Returned value = 18
Tag	StrLen("Technical Reference") // Returned value = 19

19.9.18 StrLower

Konvertiert eine Zeichenkette in Kleinbuchstaben.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrLower	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrLower( strSource )
```

Eingaben

strSource

Die zu konvertierende Zeichenkette.

Rückgabewert Gibt die Zeichenkette aus, wobei alle Buchstaben Kleinbuchstaben sind.

Beispiele

Tag-Name	Ausdruck
Tag	StrLower("Studio version 7.0") // Returned value = "studio version 7.0"
Tag	StrLower("Technical Reference") // Returned value = "technical reference"

19.9.19 StrRChr

Isoliert das letzte Auftreten einer Zeichensequenz innerhalb einer Zeichenkette.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrRChr	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrRChr( strSource, strChrSequence )
```

Eingaben

strSource

Der Quellstring.

strCharSequence

Der Referenzstring.

Rückgabewert Gibt eine Zeichenkette aus, die auf das letzte Auftreten eines Zeichens innerhalb des Strings folgt.

Beispiele

Tag-Name	Ausdruck
Tag	StrRChr("Studio version 7.0", "i") // Returned value = "ion 7.0"
Tag	StrRChr("Technical Reference", "n") // Returned value ="nce"

19.9.20 StrRight

Kopiert die letzten Zeichen einer längeren Zeichenkette.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrRight	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrRight( strSource, numQtdChar )
```

Eingaben

strSource

Der Quellstring.

numQtdChar

Die Anzahl der zu kopierenden Zeichen.

Rückgabewert Gibt eine Zeichenkette mit den am weitesten rechts stehenden Zeichen im Quellstring an.

Beispiele

Tag-Name	Ausdruck
Tag	StrRight("Studio version 7.0", 8) // Returned value = "sion 7.0"
Tag	StrRight("Technical Reference", 9) // Returned value = "Reference"

19.9.21 StrSetElement

Setzt ein bestimmtes Element in einer Zeichenkette.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrSetElement	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrSetElement( strSource, strDelimiter, numElementNumber, strValue )
```

Eingaben

strSource

Der Quellstring.

strDelimiter

Als Trennzeichen zwischen den Elementen verwendetes Zeichen.

numElementNumber

Nummer des Elements, in das der Zeichenkettenwert durch die Funktion geschrieben wird. Die Numerierung beginnt mit dem Index 1.

strValue

Zeichenkettenwert, der in numElementNumber des String-Tags strSource geschrieben wird.

Rückgabewert Gibt den mit "strValue" aktualisierten Zeichenkettenwert zurück.

Beispiele

Tag-Name	Ausdruck
StringTag	StrSetElement(strSource, "|", 2, "abcd")
StringTag	StrSetElement(strSource, ",", 3, "defg")

19.9.22 StrStr

Isoliert das erste Auftreten einer Zeichensequenz innerhalb einer Zeichenkette.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrStr	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrStr( strSource, strCharSequence )
```

Eingaben

strSource

Der Quellstring.

strCharSequence

Der Referenzstring.

Rückgabewert Gibt eine Zeichenkette aus, die auf das erste Auftreten eines Zeichens innerhalb des Quellstrings folgt.

Beispiele

Tag-Name	Ausdruck
Tag	StrStr("Studio version 7.0", "i") // Returned value = "io version 7.0"
Tag	StrStr("Technical Reference", "n") // Returned value ="nical Reference"

19.9.23 StrStrPos

Sucht das erste Auftreten eines Zeichens innerhalb einer Zeichenkette.

Anhang: WinStudio-interne Skriptsprache

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrStrPos	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrStrPos( strSource, strCharSequence )
```

Eingaben

strSource

Der Quellstring.

strCharSequence

Der Referenzstring.

Rückgabewert

Gibt eine ganze Zahl aus, die dem ersten Auftreten eines Zeichens innerhalb des Strings entspricht.

Beispiele

Tag-Name	Ausdruck
Tag	StrStrPos("Studio version 7.0", "i") // Returned value = 4
Tag	StrStrPos("Technical Reference", "a") // Returned value = 7

☞ Dem ersten Zeichen in der Zeichenkette wird der Wert 0 zugewiesen.

19.9.24 StrTrim

Entfernt Leerzeichen aus einer Zeichenkette.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrTrim	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrTrim( strReference, optNumFlag )
```

Eingaben

strReference

Eine Zeichenkette oder ein Tag vom Typ String, Seite 124 mit dem Quellstring.

optNumFlag

Eine "optionale" Ganzzahl oder ein Tag vom Typ Integer, Seite 124:

Wert von optNumFlag	Beschreibung
0	Entfernt alle Leerzeichen am Anfang und am Ende der Zeichenkette.
1	Entfernt alle Leerzeichen nur am Anfang der Zeichenkette.
2	Entfernt alle Leerzeichen nur am Ende der Zeichenkette.
3	Entfernt alle Leerzeichen außer einzelnen Leerzeichen zwischen Wörtern.

Wird für "optNumFlag" kein Wert angegeben, ist "0" der Standardwert.

Rückgabewert

Diese Funktion gibt eine Zeichenkette zurück, die gleich "strReference" abzüglich der angegebenen Leerzeichen ist. Der zurückgegebene Wert kann in jedem Tag vom Typ String, Seite 124 gespeichert werden.

Beispiele

Tag-Name	Ausdruck
Tag	StrTrim(" Studio version 7.0 ") // Returned value = "Studio version 7.0"
Tag	StrTrim(" Studio version 7.0 ", 0) // Returned value = "Studio version 7.0"
Tag	StrTrim(" Studio version 7.0 ", 1) // Returned value = "Studio version 7.0"
Tag	StrTrim(" Studio version 7.0 ", 2) // Returned value = "Studio version 7.0"
Tag	StrTrim(" Studio version 7.0 ", 3) // Returned value = "Studio version 7.0"

19.9.25 StrTrimAll

Entfernt ein bestimmtes Zeichen aus der gesamten Zeichenkette.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrTrimAll	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrTrimAll( strReference, optStrTrimChar )
```

Eingaben

strReference

Der Quellstring.

optStrTrimChar

Das Zeichen, das aus der Zeichenkette entfernt wird. Wird dieser Parameter weggelassen, werden standardmäßig die Leerzeichen aus der Zeichenkette entfernt.

Rückgabewert

Gibt eine Zeichenkette zurück, die gleich "strReference" abzüglich der durch die Funktion entfernten Zeichen ist.

Beispiele

Tag-Name	Ausdruck
Tag	StrTrimAll("Studio version 7.0 ", " ") // Returned value = "Studioversion7.0"

19.9.26 StrUpper

Konvertiert eine Zeichenkette in Großbuchstaben.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StrUpper	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
StrUpper( strSource )
```

Eingaben

strSource

Die Zeichenkette.

Rückgabewert Gibt die Zeichenkette mit allen Zeichen als Großbuchstaben zurück.

Beispiele

Tag-Name	Ausdruck
Tag	StrUpper("Studio version 7.0") // Returned value = "STUDIO VERSION 7.0"
Tag	StrUpper("Technical Reference") // Returned value = "TECHNICAL REFERENCE"

19.9.27 ValueToChar

Diese Funktion konvertiert ein Array mit Unicode-Zeichen in eine Zeichenkette.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ValueToChar	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ValueToChar("tagArray",numQtdChars)
```

Eingaben

tagArray

Der Name des Integer-Arrays, das die zu konvertierenden Werte enthält. Ist kein Array-Index angegeben, ist der Standard 0. Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

numQtdChars

Die Anzahl der zu konvertierenden Werte (mindestens 1), beginnend mit dem festgelegten Array-Index.

Rückgabewert Gibt eine Zeichenkette mit den konvertierten Werten zurück.

Beispiele Wenn Array[0] = 65, Array[1] = 66 und Array[2] = 67 dann ist der zurückgegebene Wert "ABC":

ValueToChar("Array", 3)

Wenn Array[10] = 65, Array[11] = 66 und Array[12] = 67 dann ist der zurückgegebene Wert "ABC":

ValueToChar("Array[10]", 3)

19.9.28 ValueWToChar

Diese Funktion konvertiert einen Integer-Array mit Unicode-Zeichen in einer Zeichenkette, wobei jeder Wert im Array ein Doppelbyte-Wort ist.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ValueWToChar	Zeichenkette	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ValueWToChar("tagArray",numQtdChars)
```

Eingaben

tagArray

Der Name des Integer-Arrays, das die zu konvertierenden Doppelbyte-Werte erhält. Ist kein Array-Index angegeben, ist der Standardwert 0. Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

numQtdChars

Die Anzahl der zu konvertierenden Werte (mindestens 1), beginnend mit dem festgelegten Array-Index.

Rückgabewert

Gibt eine Zeichenkette mit den konvertierten Werten zurück.

☞ Jeder Wert im Array muss ein Doppelbyte-Wort sein und jedes Wort wird für die Konvertierung in 2 Bytes aufgeteilt. Als solches unterstützt diese Funktion nur die Unicode-Zeichen 0 bis 255. Für Zeichencodes über 255 oder wenn Doppelbyte-Worte nicht benötigt werden, sollte die Funktion ValueToChar, Seite 594 verwendet werden.

Beispiele

Wenn Array[0] = 29779, Array[1] = 25717 und Array[2] = 28521, dann ist der zurückgegebene Wert "Studio":

ValueWToChar("Array", 3)

Wenn Array[10] = 29779, Array[11] = 25717 und Array[12] = 28521, dann ist der zurückgegebene Wert "Studio":

ValueWToChar("Array[10]", 3)

19.10 Datums- und Zeitfunktionen

19.10.1 Datums- und Zeitfunktionen – Allgemeines

Diese Funktionen dienen der Interaktion mit der Systemuhr oder dem Ändern von Zeitstempeln.

Datum & Uhrzeit

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
ClockGetDate, Seite 595	synchron	unterstützt	unterstützt	unterstützt
ClockGetDayOfWeek, Seite 596	synchron	unterstützt	unterstützt	unterstützt
ClockGetTime, Seite 597	synchron	unterstützt	unterstützt	unterstützt
DateTime2Clock, Seite 597	synchron	unterstützt	unterstützt	unterstützt
GetClock, Seite 598	synchron	unterstützt	unterstützt	unterstützt
Hour2Clock, Seite 598	synchron	unterstützt	unterstützt	unterstützt
SetSystemDate, Seite 599	synchron	unterstützt	unterstützt	unterstützt
SetSystemTime, Seite 599	synchron	unterstützt	unterstützt	unterstützt

19.10.2 ClockGetDate

Berechnet das Datum, basierend auf der Anzahl der seit 0:00 Uhr MEZ am 1. Januar 1970 vergangenen Sekunden (unter Berücksichtigung der aktuellen Zeitzone des Rechners).

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ClockGetDate	Datum & Uhrzeit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ClockGetDate( numSeconds )
```

Eingaben

numSeconds

Die Anzahl der seit 0:00 Uhr MEZ am 1. Januar 1970 vergangenen Sekunden.

Rückgabewert Gibt das berechnete Datum im String-Format zurück.

Beispiele

Tag-Name	Ausdruck
Tag	ClockGetDate(0) // If the computer is in the Central time zone. Returned value = 12/31/1969
Tag	ClockGetDate(1018886359) // If the computer is in the Central time zone. Returned value = 04/15/2002

☞ Diese Funktion berücksichtigt die aktuelle Zeitzone, die in der Systemsteuerung des lokalen Rechners angegeben ist.

19.10.3 ClockGetDayOfWeek

Berechnet den Wochentag, basierend auf der Anzahl der seit 12:00 Uhr MEZ am 1. Januar 1970 vergangenen Sekunden (unter Berücksichtigung der aktuellen Zeitzone des lokalen Rechners).

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ClockGetDayOf-Week	Datum & Uhrzeit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ClockGetDayOfWeek( numSeconds )
```

Eingaben

numSeconds

Die Anzahl der seit 0:00 Uhr MEZ am 1. Januar 1970 vergangenen Sekunden

Rückgabewert Gibt den Wochentag (berechnet Ganzzahlformat) wie folgt zurück:

0	Sonntag
1	Montag
2	Dienstag
3	Mittwoch
4	Donnerstag
5	Freitag
6	Samstag

Beispiele

Tag-Name	Ausdruck
Tag	ClockGetDayOfWeek(0) // If the computer is in the Central time zone. Returned value = 3
Tag	ClockGetDayOfWeek(1018886359) // If the computer is in the Central time zone. Returned value = 1

Diese Funktion berücksichtigt die aktuelle Zeitzone, die in der Systemsteuerung des lokalen Rechners angegeben ist.

19.10.4 ClockGetTime

Berechnet die Uhrzeit, basierend auf der Anzahl der seit 12:00 Uhr MEZ am 1. Januar 1970 vergangenen Sekunden (unter Berücksichtigung der aktuellen Zeitzone des lokalen Rechners).

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ClockGetTime	Datum & Uhrzeit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ClockGetTime( numSeconds )
```

Eingaben

numSeconds

Die Anzahl der seit 0:00 Uhr MEZ am 1. Januar 1970 vergangenen Sekunden.

Rückgabewert

Gibt die berechnete Uhrzeit im String-Format zurück.

Beispiele

Tag-Name	Ausdruck
Tag	ClockGetTime(0) // If the computer is in the Central time zone. Returned value = 18:00:00
Tag	ClockGetTime(1018886359) // If the computer is in the Central time zone. Returned value = 10:59:19

Diese Funktion berücksichtigt die aktuelle Zeitzone, die in der Systemsteuerung des lokalen Rechners angegeben ist.

Zum ausschließlichen Konvertieren der Anzahl der Sekunden in das Format "HH:MM:SS" muss die Funktion Format(), Seite 560 anstelle von "ClockGetTime()" verwendet werden.

19.10.5 DateTime2Clock

Berechnet, wie viele Sekunden seit 12:00 Uhr MEZ am 1. Januar 1970 vergangen sind (unter Berücksichtigung der aktuellen Zeitzone des lokalen Rechners).

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DateTime2Clock	Datum & Uhrzeit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DateTime2Clock( strDate, strTime )
```

Eingaben

strDate

Das für die Berechnung zu verwendende Datum.

strTime

Die für die Berechnung zu verwendende Uhrzeit.

Rückgabewert Gibt die Anzahl der seit 0:00 Uhr MEZ am 1. Januar 1970 vergangenen Sekunden zurück.

Beispiele

Tag-Name	Ausdruck
Tag	DateTime2Clock("12/31/1969", "18:00:00") // If the computer is in the Central time zone. Returned value = 0
Tag	DateTime2Clock("04/15/2002", "10:59:19") // If the computer is in the Central time zone. Returned value = 01018886359

☞ Diese Funktion berücksichtigt die aktuelle Zeitzone, die in der Systemsteuerung des lokalen Rechners angegeben ist.

19.10.6 GetClock

Berechnet, wie viele Sekunden zum Zeitpunkt der Ausführung der Funktion seit 12:00 Uhr MEZ am 1. Januar 1970 vergangen sind (unter Berücksichtigung der aktuellen Zeitzone des lokalen Rechners).

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetClock	Datum & Uhrzeit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetClock( )
```

Diese Funktion hat keine Parameter.

Rückgabewert Gibt die Anzahl der Sekunden zurück, die zum Zeitpunkt der Ausführung der Funktion seit 0:00 Uhr MEZ am 1. Januar 1970 vergangen sind.

Beispiele

Tag-Name	Ausdruck
Tag	GetClock() // If executed at 10:59:19 AM April 15th 2002 CST. Returned value = 101886359
Tag	GetClock() // If executed at 00:00:00 January 1st 1970 GMT. Returned value = 0

☞ Diese Funktion berücksichtigt die aktuelle Zeitzone, die in der Systemsteuerung des lokalen Rechners angegeben ist.

19.10.7 Hour2Clock

Konvertiert die Zeit im Format "HH:MM:SS" in Sekunden.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Hour2Clock	Datum & Uhrzeit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Hour2Clock( strTime )
```

Eingaben

strTime
Die Anzahl der Stunden, Minuten und Sekunden im Format "HH:MM:SS".

Rückgabewert Gibt die Anzahl der Sekunden zurück, die der festgelegten Gesamtzahl der Stunden, Minuten und Sekunden entspricht.

Beispiele

Tag-Name	Ausdruck
Tag	Hour2Clock("01:00:00") // Returned value = 3600
Tag	Hour2Clock("10:01:01") // Returned value = 36061

19.10.8 SetSystemDate

Stellt das Datum in der Uhr des Betriebssystems.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetSystemDate	Datum & Uhrzeit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
SetsystemDate( strDate )
```

Eingaben

strDate
Das Datum im Format "MM/TT/JJJJ" zum Stellen der Uhr.

Rückgabewert Gibt keine Werte zurück.

Beispiele

Tag-Name	Ausdruck
	SetSystemDate("04/15/2002") // Sets the system clock to April 15th 2002.

19.10.9 SetSystemTime

Stellt die Uhrzeit in der Uhr des Betriebssystems.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetSystemTime	Datum & Uhrzeit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
SetSystemTime( strTime )
```

Eingaben

strTime
Die Zeit im Format "HH:MM:SS" zum Stellen der Uhr.

Rückgabewert Kein Rückgabewert .

Beispiele

Tag-Name	Ausdruck
	SetSystemTime("15:45:18") // Sets the system clock to 3:45:18 PM.

19.11 Trigonometrische Funktionen

19.11.1 Trigonometrische Funktionen – Allgemeines

Diese Funktionen dienen der Berechnung von trigonometrischen Werten (z.B. Sinus, Cosinus, Tangens).

Anhang: WinStudio-interne Skriptsprache

trigonometrisch

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
ACos, Seite 600	synchron	unterstützt	unterstützt	unterstützt
ASin, Seite 600	synchron	unterstützt	unterstützt	unterstützt
ATan, Seite 601	synchron	unterstützt	unterstützt	unterstützt
Cos, Seite 601	synchron	unterstützt	unterstützt	unterstützt
Cot, Seite 601	synchron	unterstützt	unterstützt	unterstützt
Pi, Seite 602	synchron	unterstützt	unterstützt	unterstützt
Sin, Seite 602	synchron	unterstützt	unterstützt	unterstützt
Tan, Seite 602	synchron	unterstützt	unterstützt	unterstützt

19.11.2 ACos

Berechnet den Arkuskosinus eines Werts.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ACos	trigonometrisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ACos( numValue )
```

Eingaben

numValue
Numerischer Tag, für dessen Wert der Arkuskosinus berechnet wird.

Rückgabewert Gibt den Arkuskosinus von "numValue" als Bogenmaß zurück.

Beispiele

Tag-Name	Ausdruck
Tag	ACos(1) // Returned value = 0,000000
Tag	ACos(0) // Returned value = 1.570796

19.11.3 ASin

Berechnet den Arkussinus eines Werts.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ASin	trigonometrisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ASin( numValue )
```

Eingaben

numValue
Numerischer Tag, für dessen Wert der Arkussinus berechnet wird.

Rückgabewert Gibt den Arkussinus von "numValue" als Bogenmaß zurück.

Beispiele

Tag-Name	Ausdruck
Tag	ASin(1) // Returned value = 1,570796
Tag	ASin(0) // Returned value = 0.000000

19.11.4 ATan

Berechnet den Arkustangens eines Werts.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ATan	trigonometrisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ATan( numValue )
```

Eingaben

numValue

Numerischer Tag, für dessen Wert der Arkustangens berechnet wird.

Rückgabewert Gibt den Arkustangens von "numValue" als Bogenmaß zurück.

Beispiele

Tag-Name	Ausdruck
Tag	ATan(1) // Returned value = 0,785398
Tag	ATan(0) // Returned value = 1.570796

19.11.5 Cos

Berechnet den Kosinus eines Werts.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Cos	trigonometrisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Cos( numAngle )
```

Eingaben

numAngle

Der Winkel (als Bogenmaß), dessen Kosinus berechnet wird.

Rückgabewert Gibt den Kosinus von "numAngle" zurück.

Beispiele

Tag-Name	Ausdruck
Tag	Cos(1,570796) // Returned value = 0,000000
Tag	Cos(0) // Returned value = 1.000000

19.11.6 Cot

Berechnet den Kotangens eines Werts.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Cot	trigonometrisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Cot( numAngle )
```

Eingaben

numAngle

Der Winkel (als Bogenmaß), dessen Kotangens berechnet wird.

Rückgabewert Gibt den Kotangens von "numAngle" zurück.

Obwohl der Kotangens von 0 mathematisch gesehen unendlich ist, gibt "WinStudio" nur die größtmögliche Zahl zurück.

Beispiele

Tag-Name	Ausdruck
Tag	Cot(0,785398) // Returned value = 1,000000
Tag	Cot(0) // Returned value = 0.00000

19.11.7 Pi

Berechnet den Wert von "pi".

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Pi	trigonometrisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Pi()
```

Diese Funktion hat keine Parameter. Die Klammern müssen aber trotzdem angegeben werden, weil die Funktion sonst als Tag und nicht als Funktion erkannt wird.

Rückgabewert

Gibt den Wert von "pi" zurück.

Beispiele

Tag-Name	Ausdruck
Tag	Pi() // Returned value = 3.141593

19.11.8 Sin

Berechnet den Sinus eines Werts.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Sin	trigonometrisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Sin( numAngle )
```

Eingaben

numAngle
Der Winkel (als Bogenmaß), aus dem der Sinus berechnet wird.

Rückgabewert

Gibt den Sinus von "numAngle" zurück.

Beispiele

Tag-Name	Ausdruck
Tag	Sin(0) // Returned value = 0,000000
Tag	Sin(1.570796) // Returned value = 1.000000

19.11.9 Tan

Berechnet den Tangens eines Werts.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Tan	trigonometrisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Tan( numAngle )
```

Eingaben

numAngle
Der Winkel (als Bogenmaß), aus dem der Tangens berechnet wird.

Rückgabewert

Gibt den Tangens von "numAngle" zurück.

Obwohl der Tangens von Pi(), Seite 602 mathematisch gesehen unendlich ist, gibt "WinStudio" nur die größtmögliche Zahl zurück.

Beispiele

Tag-Name	Ausdruck
Tag	Tan(0) // Returned value = 0,00000
Tag	Tan(0.785398) // Returned value = 1.00000

19.12 Bildfunktionen

19.12.1 Bildfunktionen – Allgemeines

Diese Funktionen dienen dem Öffnen und Schließen von WinStudio-Bildern

Öffnen und Schließen von Fenstern

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
Close, Seite 603	asynchron	unterstützt	unterstützt	unterstützt
Open, Seite 604	asynchron	unterstützt	unterstützt	unterstützt
OpenPrevious, Seite 606	asynchron	unterstützt	unterstützt	unterstützt
ShowInplaceInput, Seite 607	asynchron	unterstützt	unterstützt	unterstützt
ShowMessageBox, Seite 608	synchron	unterstützt	unterstützt	unterstützt

19.12.2 Close

"Close" ist eine interne Skript-Funktion, die eine geöffnetes WinStudio-Bild schließt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Schließen	Bilder	asynchron	unterstützt	unterstützt	unterstützt

Syntax

```
Close(strScreen{ | ,optNumID })
```

Eingaben

strScreen

☞ Der Name des zu schließenden Bildes (ohne die Erweiterung ".scr").

Bei einigen Web-Servern muss auf Groß-/Kleinschreibung geachtet werden. Wenn Sie Ihr Projekt als Web-Anwendung bereitstellen wollen, sollten Sie für den Bildnamen nur Kleinbuchstaben verwenden.

optNumID

Die spezifische ID oder Instanznummer des zu schließenden Bildes, wenn mehr als ein Bild mit dem gleichen Namen geöffnet ist. (Die ID wird zugewiesen, wenn das Bild mit der Funktion Open, Seite 604 geöffnet wird.)

Dieser Parameter ist optional. Wenn kein Wert angegeben wird, ist die Standard-ID 0.

Rückgabewert Diese Funktion gibt keinen Wert zurück.

☞ Diese Funktion kann nicht mit Tasks, Seite 61 oder im Skript Globale Prozeduren, Seite 799 verwendet werden.

Außerdem brauchen Sie diese Funktion nicht zum Schließen eines Bildes aufrufen, wenn Sie ein anderes Bild öffnen, da das letzte Bild automatisch geschlossen wird. Weitere Informationen unter Bildeigenschaften, Seite 170.

Beispiele Schließen des WinStudio-Bildes mit dem Namen „main“:

Close("main")

Schließen des WinStudio-Bildes mit dem Namen „alarms“:

Close("alarms")

Schließen des WinStudio-Bildes mit dem Namen „main“ mit ID 10:

Close("main", 10)

19.12.3 Open

"Open" ist eine interne Skript-Funktion, die ein WinStudio-Bild öffnet.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Öffnen	Bilder	asynchron	unterstützt	unterstützt	unterstützt

Syntax

```
Open(strScreen{ | ,optNumX1{ | ,optNumY1,optNumX2,optNumY2,optNumResizeFlag{ | ,optNumID{ |
               ,optStrMnemonicList } } } })
```

Eingaben

strScreen

Der Name des zu öffnenden Bildes (ohne die Erweiterung ".scr").

☞ Bei einigen Web-Servern muss auf Groß-/Kleinschreibung geachtet werden. Wenn Sie Ihr Projekt als Web-Anwendung bereitstellen wollen, sollten Sie für den BIldnamen nur Kleinbuchstaben verwenden.

optNumX1, optNumY1, optNumX2, optNumY2

Die Koordinaten in Pixeln für die obere linke (X1,Y1) und untere rechte (X2,Y2) Ecke des Bildes.

Diese Parameter sind optional. Wenn keine Werte angegeben sind, werden die Standardwerte für Größe und Position des Bildes verwendet. Weitere Informationen unter Bildeigenschaften, Seite 170.

Bitte beachten Sie die folgenden Sonderfälle:

- Das Bild kann durch Verwendung von **Open("screen", 1)** oder **Open("screen", 1, −1, −1, −1, ...)** an der aktuellen Position des Mauszeigers geöffnet werden, wenn die Parameter am Ende gebraucht werden.
- Wenn **optNumX1** gleich **optNumX2** und **optNumY1** gleich **optNumY2**, dann wird der Standardwert für die Bildgröße verwendet, aber die linke obere Ecke des Bildes wird auf (X1,Y1) positioniert.
- Wenn **optNumX2** kleiner ist als **optNumX1** und/oder **optNumY2** kleiner ist als **optNumY1** oder wenn alle vier Parameter auf -1 gesetzt sind, werden die Parameter ignoriert und die Standardwerte für Bildgröße und -position werden verwendet.

optNumResizeFlag

Legt fest, ob die Größe der Bildobjekte beim Öffnen des Bildes angepasst wird:

Wert	Beschreibung
0	Die Größe der Bildobjekte wird nicht angepasst.
1	Die Größe der Bildobjekte wird automatisch gemäß den oben beschriebenen Koordinaten der neuen Bildgröße angepasst. Die Anpassung wird beim Öffnen des Bildes vorgenommen. Wenn der Benutzer die BIldgröße nach dem Öffnen ändert, werden die Objekte nicht nochmals angepasst.

Dieser Parameter wird benötigt, wenn alle vier Koordinaten angegeben sind.

optNumID

Dem Bild wird eine ID- oder Instanznummer zugewiesen, da mehrere Instanzen der gleichenBilddatei geöffnet werden können. (Diese Kennung wird benötigt, wenn ein Bild mithilfe der Funktion Close, Seite 603 geschlossen wird.)

Dieser Parameter ist optional. Wenn der Wert fehlt, wird der Standardwert 0 verwendet.

optStrMnemonicList

Eine Zeichenkette, die beschreibt, wie die benutzerdefinierten Eigenschaften von generischen Objekten oder verknüpften Symbolen, Seite 220im Bild beim Öffnen des Bildes vervollständigt werden. Diese Zeichenkette hat folgende Syntax:

 # Label: Value

"Label" ist der Name der Eigenschaft. "Value" ist der Tag, Ausdruck oder direkte Wert, den die Eigenschaft erhält. Es kann mehr als ein Mnemomic deklariert werden, solange die Mnemonics durch Leerzeichen getrennt sind. Weitere Informationen nachfolgend unter „Beispiele“.

Rückgabewert

Diese Funktion gibt einen der folgenden Werte zurück:

Wert	Beschreibung
0	Erfolgreich
1	Fehler

Diese Funktion kann nicht mit Tasks, Seite 61 oder im Skript Globale Prozeduren, Seite 799 verwendet werden.

Beispiele

Öffnen des Bildes „main“ unter Verwendung der Standardwerte für Bildgröße und -position:

Open("main")

Öffnen des Bildes an der aktuellen Position des Mauszeigers:

Open("main", 1)

Öffnen des Bildes an der aktuellen Position des Mauszeigers und Zuweisen der Kennung 10:

Open("main", 1, −1, −1, −1, 0, 10)

Öffnen des Bildes unter Verwendung der Standard-Bildgröße, aber mit der linken oberen Ecke positioniert auf die Koordinaten (500,250) und Zuweisen der Kennung 10:

Open("main", 500, 250, 500, 250, 0, 10)

Öffnen des Bildes unter Verwendung der Standard-Bildgröße und -position und Ersetzen der Standardeigenschaften "Mne1" und "Mne2" durch "Tag1" bzw. "Tag2":

Open("main", −1, −1, −1, −1, 1, 0, "#Mne1:Tag1 #Mne2:Tag2")

19.12.4 OpenPrevious

"OpenPrevious" ist eine interne Skript-Funktion, die das zuletzt geschlossene WInStudio-Bild erneut öffnet.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
OpenPrevious	Bilder	asynchron	unterstützt	unterstützt	unterstützt

Syntax

```
OpenPrevious({ | optNumX1,optNumY1,optNumX2,optNumY2 })
```

Eingaben

optNumX1, optNumY1, optNumX2, optNumY2

Die Koordinaten in Pixeln für die obere linke (X1,Y1) und untere rechte (X2,Y2) Ecke des Bildes.

Diese Parameter sind optional. Wenn keine Werte angegeben sind, werden die Standardwerte für Größe und Position des Bildes verwendet. Weitere Informationen unter Bildeigenschaften, Seite 170.

Rückgabewert

Diese Funktion gibt einen der folgenden Werte zurück:

Wert	Beschreibung
0	Erfolgreich
1	Fehler

☞ Diese Funktion kann nicht mit Tasks, Seite 61 oder im Skript Globale Prozeduren, Seite 799 verwendet werden.

Beispiele

Öffnen des vorherigen Bildes unter Verwendung der Standardgröße und -position:

OpenPrevious()

Öffnen der vorheriges Bildes in der oberen linken Ecke des Bildschirms in der Größe 800x600:

OpenPrevious(0, 0, 800, 600)

19.12.5 ShowInplaceInput

Diese Funktion zeigt einen einfachen Texteingabedialog an einer festgelegten Position im Projekt-Client/-Viewer an.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ShowInplaceInput	Bilder	asynchron	unterstützt	unterstützt	unterstützt

Syntax

```
ShowInplaceInput("tagOutput",numStartXPos,numStartYPos{ | ,optNumMin,optNumMax{ | ,
               {optNumEnablePasswordMode | 0 | 1 }{ | ,{ optNumShowOSVK | 0 | 1 } } } })
```

Eingaben

tagOutput

Der Name eines Tags, der die Eingabe erhält.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen eingeschlossen werden. Anderenfalls versucht die Runtime, den Wert des angegebenen Tags abzurufen.

numStartXPos

Die Startposition X in der oberen linken Ecke des Eingabedialogs – d.h. die Anzahl der Pixel zwischen dieser Ecke und dem linken Rand des Bildschirms.

numStartYPos

Die Startposition Y in der oberen linken Ecke des Eingabedialogs – d.h. die Anzahl der Pixel zwischen dieser Ecke und dem oberen Rand des Bildschirms.

optNumMin

Der minimale numerische Wert, der vom Eingabedialog akzeptiert wird.

Dieser Parameter ist optional. Wenn kein Wert angegeben wird, akzeptiert der Dialog jeden beliebigen Wert.

optNumMax

Der maximale numerische Wert, der vom Eingabedialog akzeptiert wird.

Dieser Parameter ist optional. Wenn kein Wert angegeben wird, akzeptiert der Dialog jeden beliebigen Wert.

optNumEnablePasswordMode

Option zum Aktivieren des Passwortmodus, der die Benutzereingaben wie bei Eingabe eines Passworts verbirgt:

Wert	Beschreibung
0	Eingabe als Klartext anzeigen.
1	Eingabe verbergen.

Dieser Parameter ist optional. Wenn kein Wert angegeben wird, ist der Standardwert 0.

optNumShowOSVK

Option zur Anzeige der virtuellen Standardtastatur, die in den Projekteinstellungen konfiguriert, Seite 75 ist:

Wert	Beschreibung
0	Virtuelle Tastatur nicht anzeigen.
1	Virtuelle Tastatur anzeigen.

Dieser Parameter ist optional. Wenn kein Wert angegeben wird, ist der Standardwert 0.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte aus:

Wert	Beschreibung
0	erfolgreich
-1	Ungültiger Tag für tagOutput angegeben.
-2	Ungültige Anzahl Parameter.
-3	Viewer wird nicht ausgeführt.

Beispiele

ShowInplaceInput - Beispiele

```
ShowInplaceInput( "OperatorInput", 50, 50 )
ShowInplaceInput( "OperatorInput", 50, 50, 1, 100 )
ShowInplaceInput( "OperatorInput", 50, 50, 1, 100, 0, 1 )
```

19.12.6 ShowMessageBox

Diese Funktion zeigt ein einfaches Meldungsfenster mit einer **OK**-Schaltfläche oder mit **Ja**/**Nein**-Schaltflächen an.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ShowMessage-Box	Bilder	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ShowMessageBox(strMessage{ | ,{ optNumButtons | 0 | 4 }{ | ,optStrTitle } })
ShowMessageBox( strMessage )
ShowMessageBox( strMessage, optNumButtons )
ShowMessageBox( strMessage, optNumButtons, optStrTitle )
```

Eingaben

strMessage

Der Meldungstext, der im Fenster angezeigt wird.

optNumButtons

Ein numerischer Merker zur Anzeige, welche Art von Bestätigungsschaltflächen im Meldungsfenster angezeigt werden sollen:

Wert	Beschreibung
0	Schaltfläche **OK**
4	Schaltflächen **Ja** / **Nein**



Um dem Fenster ein Ausrufezeichen hinzuzufügen – um Warnungen oder Alarme von normalen Meldungen zu unterscheiden – fügen Sie diesem Parameter 48 (vbExclamation) hinzu. Weitere Informationen nachfolgend unter „Beispiele“.

Dieser Parameter ist optional. Wenn kein Wert angegeben wird, ist der Standardwert 0.

optStrTitle

Der Meldungstitel.

Dieser Parameter ist optional. Ist kein Wert angegeben, wird kein Titel angezeigt.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte aus:

Wert	Beschreibung
1	Benutzer hat auf **OK** geklickt.
6	Benutzer hat auf **Ja** geklickt.
7	Benutzer hat auf **Nein** geklickt.

Beispiele

ShowMessageBox - Beispiele

```
ShowMessageBox( "The action could not be completed." )

ShowMessageBox( "Continue with action?", 4 )

ShowMessageBox( "The action could not be completed.", 0+48, "Alert" )
```

Anhang: WinStudio-interne Skriptsprache

19.13 Sicherheitsfunktionen

19.13.1 Sicherheitsfunktionen – Allgemeines

Diese Funktionen dienen der Verwaltung von Benutzern und Gruppen im Sicherheitssystem des Projekts.

Sicherheit

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
BlockUser, Seite 610	synchron	unterstützt	unterstützt	unterstützt
CheckESign, Seite 611	synchron	unterstützt	unterstützt	unterstützt
CreateUser, Seite 611	synchron	unterstützt	unterstützt	unterstützt
ExportSecuritySystem, Seite 612	synchron	unterstützt	unterstützt	unterstützt
GetSecuritySystemStatus, Seite 613	synchron	unterstützt	unterstützt	unterstützt
GetUserFullName, Seite 615	synchron	unterstützt	unterstützt	unterstützt
GetUserNames, Seite 615	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
GetUserPWDAging, Seite 616	synchron	unterstützt	unterstützt	unterstützt
GetUserState, Seite 617	synchron	unterstützt	unterstützt	unterstützt
RemoveUser, Seite 619	synchron	unterstützt	unterstützt	unterstützt
ImportSecuritySystem, Seite 617	synchron	unterstützt	unterstützt	unterstützt
SetPassword, Seite 619	synchron	unterstützt	unterstützt	unterstützt
UnblockUser, Seite 620	synchron	unterstützt	unterstützt	unterstützt

19.13.2 BlockUser

"BlockUser" ist eine interne Skript-Funktion, die einen bestehenden Benutzer für das Anmelden an einem Projekt sperrt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
BlockUser	Sicherheit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
BlockUser(strUserName)
```

Eingaben

strUserName

Der Name des zu sperrenden Benutzers.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte aus:

Wert	Beschreibung
0	Benutzer erfolgreich gesperrt.
1	Ungültige Anzahl Parameter.
2	Falscher Parametertyp.
3	Angegebener Benutzer existiert nicht.

Wert	Beschreibung
4	Der aktuelle angemeldete Benutzer verfügt nicht über die Berechtigung zum Sperren (d.h. der Benutzer verfügt nicht über Edit Security System).
5	Der Vorgang auf dem verteilten Sicherheitssystem ist fehlgeschlagen.
6	Benutzer kann nicht gesperrt werden.
7	Der aktuelle Sicherheitsmodus lässt kein Sperren/Entsperren von Benutzern zu.
8	Interner Fehler.

Beispiele

Sperren des Benutzers Bob:

BlockUser("Bob")

Sperren des in Position 3 des Arrays "badUsers" genannten Benutzers:

BlockUser(badUsers[3])

Sperren des aktuell angemeldeten Benutzers:

BlockUser(UserName)

19.13.3 CheckESign

"CheckESign" ist eine interne Skript-Funktion, die den Benutzer auffordert, durch Eingabe von Benutzername und Passwort eine elektronische Signatur für ein Ereignis zur Verfügung zu stellen. Sie kann aufgerufen werden, um bestimmte Ausdrücke und Skripte zu sichern, genau wie die Option **E-Sign** dem Sichern von Bildobjekten dient.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
CheckESign	Sicherheit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
CheckESign()
```

Diese Funktion hat keine Parameter. Durch Aufruf der Funktion wird ein Sicherheitsdialog angezeigt, in dem der Benutzer Benutzername und Passwort eingeben muss.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte aus:

Wert	Beschreibung
0	Benutzername und/oder Passwort nicht akzeptiert.
1	Benutzername und Passwort akzeptiert.



Benutzernamen und Passwörter werden im Ordner Sicherheit, Seite 407 gespeichert.

Beispiele

CheckESign - Beispiele

```
CheckESign()
```

19.13.4 CreateUser

Diese Funktion erstellt einen neuen Benutzer im Sicherheitssystem des Projekts.

Anhang: WinStudio-interne Skriptsprache

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
CreateUser	Sicherheit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
CreateUser(strUserName,strGroup,strPassw{ | ,optStrUserFullName })
```

Eingaben

strUserName

Der Name des zu erstellenden Benutzers.

strGroup

Der Name der Gruppe, der der Benutzer zugeordnet werden soll.

strPassw

Das Benutzerpasswort.

optStrUserFullName

Der vollständige Name des Benutzers.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte aus:

Wert	Beschreibung
-1	Interner Fehler; wenden Sie sich an den technischen Support.
0	Neuer Benutzer erfolgreich erstellt.
1	Ungültige Anzahl Parameter.
2	Falscher Parametertyp.
3	Benutzername bereits vorhanden.
4	Gruppe nicht vorhanden.
5	Speichern in Konfigurationsdatei fehlgeschlagen.
6	Ungültiger Benutzer.
7	Vollständiger Name der Benutzers bereits vorhanden.



Mit dieser Funktion erstellte Benutzer werden nicht im Ordner "Sicherheit" des Projekts angezeigt, weil sie in einer sekundären Datenbank gespeichert werden. Zur Verwaltung dieser Datenbank verwenden Sie das Programm "ExtUser.exe" (zu finden im Unterordner "Bin" des Anwendungsordners).

Beispiele

CreateUser - Beispiele

```
CreateUser( "Bob", "Admin", "Chocolate", "Bob Smith" )
CreateUser( "Albert", "Engineering", "EMC2", "Albert Jones" )
```

19.13.5 ExportSecuritySystem

Diese Funktion dient dem Export der Sicherheitssystemkonfiguration in eine verschlüsselte Datei.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ExportSecurity-System	Sicherheit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ExportSecuritySystem(strFileName,strPassword)
```

Eingaben

strFileName

Der vollständige Dateipfad und -name zum Speichern der Konfigurationsdatei.

strPassword

Das Hauptpasswort für das Sicherheitssystem. Das gleiche Passwort wird für den Schutz der exportierten Datei verwendet.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte aus:

Wert	Beschreibung
-2	Falscher Parametertyp.
-1	Ungültige Anzahl Parameter.
0	Sicherheitsdaten konnten nicht geschrieben werden.
1	Datei erfolgreich exportiert.

Beispiele

ExportSecuritySystem - Beispiele

```
ExportSecuritySystem( "C:\security.txt" )

ExportSecuritySystem( "C:\security.txt", "mypa55w0rd" )
```

19.13.6 GetSecuritySystemStatus

Diese Funktion ruft den Status des Sicherheitssystems und dessen Verbindung zum Authentifizierungsserver ab, wenn der Sicherheitsmodus entweder Verteilt–Client oder Domäne (LDAP) ist.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetSecuritySys-temStatus	Sicherheit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetSecuritySystemStatus({ | { optNumType | 0 | 1 | 2 } })

GetSecuritySystemStatus()
GetSecuritySystemStatus( optNumType )
```

Eingaben

optNumType

Legt fest, welche Aktionen bei der Statusaktualisierung ausgeführt werden.

Wert	Beschreibung
0	Ausführen einer Schnellprüfung mittels Ping oder Bind (je nach Servereinstellungen), aber keine weiteren Aktionen.
1	Erneutes Laden von Benutzern und Gruppen vom Authentifizierungsserver erzwingen.
2	Benutzer und Gruppen aus dem Zwischenspeicher löschen.

Dies ist ein optionaler Parameter; wird kein Wert angegeben, ist der Standardwert "0".

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte aus:

Wert	Sicherheitsmodus Verteilt–Client	Sicherheitsmodus Domäne (LDAP)
0	Kein Cache	Verbindungstimeout
1	Aktualisierter Cache	Bindungstimeout
2	Veralteter lokaler Cache	Abfragetimeout
3	Veralteter Server-Cache	Getrennt
4	Vom Server getrennt	Verbunden
5	entf.	Abfrage ergab keine Benutzer oder Gruppen
6	entf.	Ungültige/r Benutzer oder Gruppe

Diese Funktion gibt den gleichen Wert aus, der an den im Feld **Status-Tag** konfigurierten Projekt-Tag gesendet wird, in den Servereinstellungen Sicherheitssystem, Seite 412.

Außerdem gibt es neben dem Aufruf dieser Funktion weitere Ereignisse, bei denen eine Statusaktualisierung erfolgt:

- Wenn ein Benutzer sich am Projekt anmeldet. Besonders wenn ein Benutzer sich über den eingebauten Dialog "LogOn" anmeldet (geöffnet entweder durch Aufruf der Funktion "LogOn" oder durch Auswahl des Menübefehls **LogOn** im Viewer), wird der Status aktualisiert, bevor der Dialog angezeigt wird.
- Wenn der Authentifizierungsserver offline ist und das Wiederholintervall (konfiguriert in den Servereinstellungen des Sicherheitssystems) abgelaufen ist.
- Wenn die Sicherheitssystemeinstellungen in der Entwicklungsungebung geöffnet werden.

Bei jeder Aktualisierung des Status wird der neue Wert sofort an den im Feld **Status-Tag** konfigurierten Projekt-Tag gesendet.

Beispiele

Status des Sicherheitssystems holen:

GetSecuritySystemStatus()

Erneutes Laden aller Benutzer und Gruppen vom Authentifizierungsserver durch das Sicherheitssystem erzwingen:

GetSecuritySystemStatus(1)

19.13.7 GetUserFullName

Diese Funktion ruft den vollständigen Namen (falls angegeben) eines angegebenen Benutzers im Projektsicherheitssystem ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetUserFullName	Sicherheit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetUserFullName(strUserName,"tagUserFullName")
```

Eingaben

strUserName

Name eines Benutzers im Projektsicherheitssystem.

tagUserFullName

Name eines Tags (Typ String), der nach dem Funktionsaufruf den vollständigen Namen des angegebenen Benutzers erhält. Wenn für den angegebenen Benutzer kein vollständiger Name definiert ist, erhält der Tag eine leere Zeichenkette ("").

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht die Funktion, den Wert des angegebenen Tags abzurufen.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte aus:

Wert	Beschreibung
0	Angegebener Benutzer existiert nicht.
1	Erfolgreich; angegebener Benutzer ist vorhanden.

Beispiele

Holen des vollständigen Namens des aktuell angemeldeten Benutzers (über den System-Tag "UserName"):

GetUserFullName(UserName, "UserFullName")

Holen den vollständigen Namens des Benutzers "engineer1":

GetUserFullName("engineer1", "UserFullName")

19.13.8 GetUserNames

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetUserNames	Sicherheit	synchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
GetUserNames( "tagUsers", optNumUserType, "opttagGroups" )
```

Eingaben

tagUsers

Name des Array-Tags, der die Liste der Benutzernamen erhält.

optNumUserType

Typ der abzurufenden Benutzer

0	Gibt alle Benutzer aus
1	Nur während der Laufzeit erstellte Benutzer
2	Nur mithilfe der Entwicklungsumgebung erstellte Benutzer

opttagGroups

Name des Array-Tags, der die Liste der Gruppenamen für jeden Benutzer erhält.

Rückgabewert

Die Anzahl der Benutzer oder eine negative Zahl aus der folgenden Liste:

−1	Ungültige Anzahl Parameter
−2	tagUsers ist ungültig
−3	optNumUserType ist ungültig
−4	opttagGroups ist ungültig
−5	Fehler, Funktion kann im Thin Client nicht aufgerufen werden

Beispiele

Tag-Name	Ausdruck
NumberOfUsers	GetUserNames("UsersArray") // Retrieves all users, stores names in the UsersArray tag and the number of users in the NumberOfUsers tag.
NumberOfUsers	GetUserNames("UsersArray", 1) // Retrieves all users created during runtime, stores names in the UsersArray tag and the number of users in the NumberOfUsers tag.
NumberOfUsers	GetUserNames("UsersArray", 2) // Retrieves all users created in the development environment, stores names in the UsersArray tag and the number of users in the NumberOfUsers tag.
NumberOfUsers	GetUserNames("UsersArray", 2, "Groups") // Retrieves all users created in the development environment, stores names in the UsersArray tag and the number of users in the NumberOfUsers tag. The group name per each user is stored in the Groups tag.

19.13.9 GetUserPwdAging

Gibt die verbleibende Zeit vor Ablauf des Passwortes für einen bestimmten Benutzer zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetUserPwdAging	Sicherheit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetUserNames( strUser )
```

Eingaben

strUser

Benutzername, dessen Passwortalterung geprüft wird.

Rückgabewert

≤0	Anzahl der Stunden seit Ablauf des Passworts.
0	Angegebener Benutzer ist nicht angemeldet.
>0	Anzahl der verbleibenden Stunden bis Ablauf des Passworts.

☞ Wenn diese Funktion nicht korrekt ausgeführt wird (z.B. ungültiger Benutzername) oder wenn der angegebene Benutzer nicht angemeldet ist, meldet die Funktion die Qualität BAD.

Beispiele

Tag-Name	Ausdruck
TagHoursToExpire	GetUserPwdAging("John") // Returns the number of hours before the password for the User "John" expires.
TagHoursToExpire	GetUserPwdAging(UserName) // Returns the number of hours before the password for the current User logged on the system expires.

19.13.10 GetUserState

Ruft den aktuellen Status eines ausgewählten Benutzers ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetUserState	Sicherheit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetUserState(strUserName)
```

Eingaben

strUserName

Name des Benutzers

Rückgabewert Diese Funktion gibt die folgenden möglichen Werte aus:

Wert	Beschreibung
-3	Angegebener Benutzer existiert nicht.
-2	Falscher Parametertyp.
-1	Ungültige Anzahl Parameter.
0	Angegebener Benutzer ist entsperrt.
1	Angegebener Benutzer ist gesperrt.

Beispiele

Tag-Name	Ausdruck
Tag	GetUserState("Bob")
Tag	GetUserState ("Albert")

19.13.11 ImportSecuritySystem

Diese Funktion importiert eine Sicherheitssystemkonfiguration aus einer externen Datei.

Anhang: WinStudio-interne Skriptsprache

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ImportSecurity-System	Sicherheit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ImportSecuritySystem(strSecuritySystemPassword,strFileName,strFilePassword{|,optNumMode})
```

Eingaben

strSecuritySystemPassword

Das Hauptpasswort für die aktuelle Sicherheitssystemkonfiguration des Projekts. (Das Sicherheitssystem muss aktiviert sein.)

strFileName

Der vollständige Dateipfad und -name der zu importierenden Konfigurationsdatei. (Die Datei muss zuvor aus einem "WinStudio"-Projekt exportiert werden, mithilfe des Konfigurationswerkzeugs Sicherheitssystem, Seite 431 oder der Funktion ExportSecuritySystem, Seite612.)

strFilePassword

Das Passwort für die angegebene Konfigurationsdatei.

optNumMode

Ein numerischer Wert, der festlegt, wie die importierten Einstellungen eingefügt werden sollen:

Wert	Beschreibung
0	Anhängen der importierten Einstellungen an die aktuellen Einstellungen. Bei Widersprüchen werden die aktuellen Einstellungen durch die importierten Einstellungen ersetzt.
1	Anhängen der importierten Einstellungen an die aktuellen Einstellungen. Bei Widersprüchen werden die aktuellen Einstellungen beibehalten.
2	Die aktuellen Einstellungen vollständig durch die importierten Einstellungen ersetzen.

Dies ist ein optionaler Parameter; wird kein Wert angegeben, ist der Standardwert "0".

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte aus:

Wert	Beschreibung
-2	Falscher Parametertyp.
-1	Ungültige Anzahl Parameter.
0	Sicherheitsdaten konnten nicht gelesen werden.
1	Datei erfolgreich importiert.

Beispiele

Programm:

```
ExportSecuritySystem( "curr3ntPa55w0rd", "C:\security.txt", "1mp0rtPa55w0rd" )
ExportSecuritySystem( "curr3ntPa55w0rd", "C:\security.txt", "1mp0rtPa55w0rd", 2 )
```

19.13.12 RemoveUser

Entfernt einen Benutzer aus dem WinStudio-Sicherheitssystem.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
RemoveUser	Sicherheit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
RemoveUser( strUserName )
```

Eingaben

strUserName

Der Name des zu entfernenden Benutzers.

Rückgabewert

Diese Funktion liefert folgende Rückgabewerte:

Wert	Beschreibung
0	Benutzer erfolgreich entfernt.
1	Ungültige Anzahl Parameter.
2	Falscher Parametertyp.
3	Der aktuelle angemeldete Benutzer verfügt nicht über die Berechtigung zum Entfernen (d.h. der Benutzer verfügt nicht über Edit Security System).
4	Benutzer kann nicht entfernt werden.
5	Angegebener Benutzer existiert nicht.
6	Fehler auf Komponentenebene.
7	Speichern in Konfigurationsdatei fehlgeschlagen.
8	Der aktuelle Sicherheitsmodus lässt kein Entfernen von Benutzern zu.

Beispiele

Tag-Name	Ausdruck
Tag	RemoveUser("Bob")
Tag	RemoveUser("Albert")



Diese Funktion kann nur zum Entfernen von Benutzern, die mithilfe der Funktion CreateUser( ), Seite611 erstellt wurden, verwendet werden.

19.13.13 SetPassword

Legt ein neues Passwort für einen Benutzer fest.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetPassword	Sicherheit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
SetPassword( strUserName, optStrNewPassword )
```

Eingaben

strUserName

Name des Benutzers

optStrNewPassword

"Optional" Das neue Passwort

☞ Wird dieser Parameter ausgelassen, öffnet die Funktion "SetPassword" einen Dialog "Neues Passwort eingeben", so dass der Benutzer ein neues Passwort eingeben kann.

Rückgabewert

Die Funktion liefert einen numerischen Wert mit folgender Bedeutung:

Wert	Beschreibung
-1	Interner Fehler; wenden Sie sich an den technischen Support.
0	Passwort erfolgreich festgelegt.
1	Ungültige Anzahl Parameter.
2	Falscher Parametertyp.
3	Angegebener Benutzer existiert nicht.
4	Eintrittsvarianter Aufruf nicht zulässig.
5	Benutzer hat auf Abbrechen geklickt.
6	Gruppe nicht vorhanden.
7	Passwort zu unsicher.
8	Ungültiges Passwort.
9	Ungültiger Benutzer.

Beispiele

Tag-Name	Ausdruck
Tag	SetPassword("Bob")
Tag	SetPassword("Albert", "anemarie")

19.13.14 UnblockUser

Diese Funktion entsperrt einen gesperrten Benutzer im Sicherheitssystem.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
UnblockUser	Sicherheit	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
UnblockUser( strUserName )
```

Eingaben

strUserName

Der Name des zu entsperrenden Benutzers.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte aus:

Wert	Beschreibung
0	Benutzer erfolgreich entsperrt.
1	Ungültige Anzahl Parameter.
2	Falscher Parametertyp.

Wert	Beschreibung
3	Angegebener Benutzer existiert nicht.
4	Der aktuelle angemeldete Benutzer verfügt nicht über die Berechtigung zum Entsperren (d.h. der Benutzer verfügt nicht über Edit Security System).
5	Der Vorgang auf dem verteilten Sicherheitssystem ist fehlgeschlagen.
6	Angegebener Benutzer kann nicht entsperrt werden.
7	Der aktuelle Sicherheitsmodus lässt kein Sperren/Entsperren von Benutzern zu.
8	Interner Fehler.

Beispiele

Tag-Name	Ausdruck
Tag	UnblockUser("Bob")
Tag	UnblockUser("Albert")

19.14 Funktionen zur Modulaktivität

19.14.1 Funktionen zur Modulaktivität – Allgemeines

Diese Funktionen dienen der Verwaltung der verschiedenen Laufzeitmodule eines Projekts – z.B. Hintergrundtasks, Datenserver und Projekt-Viewer – sowie dem Zusammenspiel dieser Module mit dem Betriebssystem.

Modulaktivität

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
AppActivate, Seite 622	asynchron	unterstützt	unterstützt	unterstützt
AppIsRunning, Seite 624	synchron	unterstützt	unterstützt	unterstützt
AppPostMessage, Seite 624	synchron	unterstützt	unterstützt	unterstützt
AppSendKeys, Seite 625	synchron	unterstützt	nicht unterstützt	unterstützt
CleanReadQueue, Seite 626	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
CloseSplashWindow, Seite 626	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
DisableMath, Seite 627	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt
EnableMath, Seite 627	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt
EndTask, Seite 627	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt
ExitWindows, Seite 628	asynchron	unterstützt	nicht unterstützt	unterstützt
IsScreenOpen, Seite 629	synchron	unterstützt	unterstützt	nicht unterstützt
IsTaskRunning, Seite 629	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
IsViewerInFocus, Seite 630	synchron	unterstützt	nicht unterstützt	nicht unterstützt

Anhang: WinStudio-interne Skriptsprache

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
KeyPad, Seite 630	asynchron	unterstützt	unterstützt	unterstützt
LogOff, Seite 632	asynchron	unterstützt	unterstützt	unterstützt
LogOn, Seite 632	asynchron	unterstützt	unterstützt	unterstützt
Math, Seite 633	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
PostKey, Seite 633	synchron	unterstützt	unterstützt	unterstützt
Recipe, Seite 634	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
Report, Seite 635	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
RunGlobalProcedureOnServer, Seite 640	synchron	unterstützt	unterstützt	unterstützt
RunVBScript, Seite 643	synchron	unterstützt	unterstützt	unterstützt
SecureViewerReload, Seite 644	synchron	nicht unterstützt	nicht unterstützt	unterstützt (nur Secure Viewer)
SendKeyObject, Seite 644	synchron	unterstützt	nicht unterstützt	nicht unterstützt
SetAppPath, Seite 646	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
SetViewerInFocus, Seite 647	synchron	unterstützt	nicht unterstützt	unterstützt
SetViewerPos, Seite 647	synchron	unterstützt	nicht unterstützt	unterstützt
ShutDown, Seite 648	synchron	unterstützt	unterstützt	nicht unterstützt
StartTask, Seite 648	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt
ViewerPostMessage, Seite 649	asynchron	unterstützt	unterstützt	unterstützt
Wait, Seite 650	synchron	unterstützt	unterstützt	unterstützt
WinExec, Seite 651	asynchron	unterstützt	unterstützt	unterstützt
WinExecIsRunning, Seite 653	synchron	unterstützt	unterstützt	unterstützt

19.14.2 AppActivate

"AppActivate" ist eine interne Skript-Funktion, die ein anderes, bereits geöffnetes Anwendungsfenster aktiviert (d.h. in den Vordergrund bringt).

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
AppActivate	Modulaktivität	asynchron	unterstützt	unterstützt	unterstützt

Syntax

```
AppActivate(strAppTitle{ | ,{ optNumActiv | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 }{ |,optNumTimeout } })
```

Eingaben

strAppTitle

Der vollständige Titel des Anwendungsfensters (wie in der Titelleiste gezeigt).

optNumActiv

Steuert, wie das angegebene Fenster aktiviert werden soll:

Wert	Befehl	Beschreibung
0	SW_HIDE	Das aktuell aktive Fenster wird verborgen und das angegebene Fenster wird aktiviert.
1	SW_SHOWNORMAL	Das angegebene Fenster wird aktiviert und angezeigt. Wird das Fenster minimiert oder maximiert, wird es auf seine Originalgröße und -position zurückgesetzt. Dieser Befehl sollte für das erstmalige Anzeigen eines Fensters verwendet werden.
2	SW_SHOWMINIMIZED	Das angegebene Fenster wird aktiviert und dann minimiert.
3	SW_SHOWMAXIMIZED	Das angegebene Fenster wird aktiviert und dann maximiert.
4	SW_SHOWNOACTIVATE	Das angegebene Fenster wird angezeigt, aber nicht aktiviert. Wird das Fenster minimiert oder maximiert, wird es auf seine Originalgröße und -position zurückgesetzt.
5	SW_SHOW	Das angegebene Fenster wird mit der aktuellen Größe und Position aktiviert und angezeigt. Diese Funktion ist ähnlich der Funktion **SW_SHOWNORMAL**, mit dem Unterschied, dass ggf. minimierte oder maximierte Fenster in diesem Zustand verbleiben.
6	SW_MINIMIZE	Das angegebene Fenster wird minimiert und dann wird das nächste geöffnete Fenster aktiviert.
7	SW_SHOWMINNOACTIVATE	Das angegebene Fenster wird als minimiertes Fenster angezeigt, aber nicht aktiviert.
8	SW_SHOWNA	Das angegebene Fenster wird in der aktuellen Größe und Position angezeigt, aber nicht aktiviert. Diese Funktion ist ähnlich der Funktion SW_SHOWNOACTIVATE, mit dem Unterschied, dass ggf. minimierte oder maximierte Fenster in diesem Zustand verbleiben.
9	SW_RESTORE	Das angegebene Fenster wird aktiviert und angezeigt. Wird das Fenster minimiert oder maximiert, wird es auf seine Originalgröße und -position zurückgesetzt. Dieser Befehl sollte beim Wiederherstellen eines minimierten Fensters verwendet werden.

Dieser Parameter ist optional. Wenn kein Wert angegeben wird, ist der Standardbefehl "SW_RESTORE".

optNumTimeout

Die Timeoutzeit (in Millisekunden) für die erfolgreiche Ausführung der Funktion. Wird die Funktion aus welchem Grund auch immer nicht innerhalb dieses Zeitraums ausgeführt, wird sie abgebrochen.

Dieser Parameter ist optional. Wird kein Wert angegeben, beträgt die Standard-Timeoutzeit 5 s (oder 5000 ms).

Rückgabewert

Diese Funktion gibt einen der folgenden Werte zurück:

Wert	Beschreibung
0	**ERROR**: Das angegebene Anwendungsfenster wurde nicht aktiviert oder hat anderweitig nicht innerhalb der angegebenen Timeoutzeit reagiert.
1	**SUCCESS**: Das angegebene Anwendungsfenster wurde erfolgreich aktiviert.



"AppActivate" ist ähnlich der Funktion ShowWindow in Microsoft Windows API und lässt viele der gleichen Möglichkeiten zu. Weitere Informationen finden Sie in der Dokumentation zu Windows API.

Beispiele Anzeigen des Microsoft Word Dokuments "test.doc":

AppActivate("test.doc — Microsoft Word", 5)

19.14.3 AppIsRunning

"AppIsRunning" ist eine interne Skript-Funktion, die überprüft, ob ein anderes Anwendungsfenster (z.B. ein Microsoft Word Dokument) geöffnet ist und ausgeführt wird.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
AppIsRunning	Modulaktivität	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
AppIsRunning(strAppTitle{ | ,optNumTimeout })
```

Eingaben

strAppTitle

Der vollständige Titel des Anwendungsfensters (wie in der Titelleiste gezeigt).

optNumTimeout

Die Timeoutzeit (in Millisekunden) für die erfolgreiche Ausführung der Funktion. Wird die Funktion aus welchem Grund auch immer nicht innerhalb dieses Zeitraums ausgeführt, wird sie abgebrochen.

Dieser Parameter ist optional. Wird kein Wert angegeben, beträgt die Standard-Timeoutzeit 5 s (oder 5000 ms).

Rückgabewert Diese Funktion gibt einen der folgenden Werte zurück:

Wert	Beschreibung
0	**ERROR**: Das angegebene Anwendungsfenster ist nicht geöffnet oder hat anderweitig nicht innerhalb der angegebenen Timeoutzeit reagiert.
1	**SUCCESS**: Das angegebene Anwendungsfenster ist geöffnet und wird ausgeführt.



"AppIsRunning" ist ähnlich der Funktion IsWindow in Microsoft Windows API. Weitere Informationen finden Sie in der Dokumentation zu Windows API.

Beispiele Überprüfen, ob das Microsoft Word Dokument "test.doc" geöffnet ist und ausgeführt wird:

AppIsRunning("test.doc — Microsoft Word")

19.14.4 AppPostMessage

"AppPostMessage" ist eine interne Skript-Funktion, die eine Windows-Systemmeldung an ein anderes Anwendungsfenster sendet.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
AppPostMessage	Modulaktivität	asynchron	unterstützt	unterstützt	unterstützt

Syntax

```
AppPostMessage(strAppTitle,strMessage,numWParam,numLParam{ | ,optNumTimeout })
```

Eingaben

strAppTitle

Der vollständige Titel des Anwendungsfensters (wie in der Titelleiste gezeigt).

strMessage

Der Name oder Code der Systemmeldung.

Die Meldungen "CLOSE", "MINIMIZE", "MAXIMIZE" und "RESTORE" können als String-Werte in Anführungszeichen eingegeben werden. Alle anderen Meldungscodes müssen als numerische Werte angegeben werden.

numWParam

Zusätzliche meldungsspezifische Information.

numLParam

Zusätzliche meldungsspezifische Information.

optNumTimeout

Die Timeoutzeit (in Millisekunden) für die erfolgreiche Ausführung der Funktion. Wird die Funktion aus welchem Grund auch immer nicht innerhalb dieses Zeitraums ausgeführt, wird sie abgebrochen.

Dieser Parameter ist optional. Wird kein Wert angegeben, beträgt die Standard-Timeoutzeit 5 s (oder 5000 ms).

Rückgabewert

Diese Funktion gibt einen der folgenden Werte zurück:

Wert	Beschreibung
0	**ERROR**: Das angegebene Anwendungsfenster ist nicht geöffnet oder hat anderweitig nicht innerhalb der angegebenen Timeoutzeit reagiert.
1	**SUCCESS**: Das angegebene Anwendungsfenster ist geöffnet und wird ausgeführt.

"AppPostMessage" ist ähnlich der Funktion PostMessage in Microsoft Windows API und lässt viele der gleichen Möglichkeiten zu. Weitere Informationen sowie eine Liste der verfügbaren Systemmeldungen finden Sie in der Dokumentation für Windows API.

Beispiele

Schließen der Taschenrechneranwendung:

AppPostMessage("Calculator", "CLOSE", 3, 1)

19.14.5 AppSendKeys

Sendet Tastaturbefehle an die aktive Anwendung.

Anhang: WinStudio-interne Skriptsprache

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
AppSendKeys	Modulaktivität	synchron	unterstützt	nicht unterstützt	unterstützt

Syntax

```
AppSendKeys( strKeys1, strKeys2, … , strKeysN )
```

Eingaben

strKeys (1-N)
String-Tags mit den zu verwendenden Tastaturbefehlen.

Rückgabewert Gibt keine Werte zurück.

Beispiele

Tag-Name	Ausdruck
	AppSendKeys("S", "t", "u", "d", "i", "o", "<ENTER>")
	AppSendKeys("<Alt>F")

☞ Sie können <ALT>, <STRG> oder <SHIFT> im Text angeben, um den entsprechenden Code für die Tastaturbefehle "Alt", "Strg" oder "Shift" zu senden. Zum Senden des Zeichens "<" geben Sie "<<" im Text an.

19.14.6 CleanReadQueue

Entfernt alle Lesemeldungen aus dem Kommunikationsmodul.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
CleanReadQueue	Modulaktivität	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
CleanReadQueue()
```

Diese Funktion hat keine Parameter.

Rückgabewert Gibt keine Werte zurück.

Beispiele

Tag-Name	Ausdruck
	CleanReadQueue()

☞ Diese Funktion sollte nicht in neuen Projekten verwendet werden. Sie ist aber aus Kompatibilitätsgründen weiterhin gültig für bereits erstellte "WinStudio" -Projekte aus früheren Versionen.

19.14.7 CloseSplashWindow

Schließt den Startbildschirm "WinStudio".

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
CloseSplashWindow	Modulaktivität	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
CloseSplashWindow()
```

Diese Funktion hat keine Parameter.

Rückgabewert Gibt keine Werte zurück.

Beispiele

Tag-Name	Ausdruck
	CloseSplashWindow()

19.14.8 DisableMath

"DisableMath" ist eine interne Skript-Funktion, die die Ausführung aller Mathematikarbeitsblätter stoppt

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DisableMath	Modulaktivität	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
DisableMath()
```

Diese Funktion hat keine Parameter.

Rückgabewert Für diese Funktion gibt es keine Rückgabewerte.

Zur weiteren Ausführung der Mathematik-Arbeitsblätter, Seite 367 rufen Sie die Funktion EnableMath, Seite 627 auf.

Beispiele

Tag-Name	Ausdruck
	DisableMath()

19.14.9 EnableMath

"EnableMath" ist eine interne Skript-Funktion, die die Ausführung aller Mathematikarbeitsblätter wieder aktiviert

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
EnableMath	Modulaktivität	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
EnableMath()
```

Diese Funktion hat keine Parameter.

Rückgabewert Für diese Funktion gibt es keine Rückgabewerte.

In den meisten Fällen wurde die Ausführung durch Aufruf der Funktion DisableMath, Seite 627 angehalten.

Beispiele

Tag-Name	Ausdruck
	EnableMath()

19.14.10 EndTask

Beendet das aktuell ausgeführte "WinStudio"-Modul.

Anhang: WinStudio-interne Skriptsprache

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
EndTask	Modulaktivität	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
EndTask(strTask)
```

Eingaben

strTask

Name des zu beendenden Tasks (aus nachfolgender Liste):

BGTASK	Hintergrund-Tasks
VIEWER	Viewer
DBSPY	Datenbankspion
LOGWIN	LogWin
DRIVERDriverName	Driver
UNIDDECL	DDE-Client
UNINDDE	DDE-Server
UNIODBC	ODBC
TCPSERVER	TCP/IP-Server
TCPCLIENT	TCP/IP-Client
OPCCLIENT	OPC

Rückgabewert

Gibt keine Werte zurück.

Beispiele

Tag-Name	Ausdruck
	EndTask("Viewer")

☞ Zum Schließen eines Treibers muss die folgende Syntax verwendet werden:

EndTask("DriverDriverName")

Wobei DriverName der Name der **.dll**-Datei des Treibers ist. Zum Beispiel:

EndTask("DriverMODBU")

19.14.11 ExitWindows

Beendet das Windows-Betriebssystem auf eine festgelegte Weise.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ExitWindows	Modulaktivität	asynchron	unterstützt	nicht unterstützt	unterstützt

Syntax

```
ExitWindows(numExitCode)
```

Eingaben

numExitCode

Ein numerischer Code zur Festlegung, wie Windows beendet werden soll:

Wert	Beschreibung
0	Neu starten
1	Abmelden
2	Herunterfahren

Rückgabewert

Gibt keine Werte zurück.

Beispiele

Tag-Name	Ausdruck
	ExitWindows(1)

19.14.12 IsScreenOpen

Überprüft, ob ein WinStudio-Bild geöffnet ist.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
IsScreenOpen	Modulaktivität	asynchron	unterstützt	unterstützt	nicht unterstützt



Diese Funktion kann nicht mit Tasks, Seite 61 oder im Skript Globale Prozeduren, Seite 799 verwendet werden.

Syntax

```
IsScreenOpen(strScreen{ | ,optNumID })
```

Eingaben

strScreen

Der Name des zu überprüfenden WinStudio-Bildes.

optNumID

Die eindeutige Instanznummer der Bildes. (Die Nummer wird zugewiesen, wenn das Bild mit der Funktion Open, Seite 604 geöffnet wird.)

Dieser Parameter ist optional. Wenn kein Wert angegeben wird, ist die Standard-ID 0.

Rückgabewert

0	Das Bild ist nicht geöffnet.
1	Das Bild ist geöffnet.

Beispiele

Tag-Name	Ausdruck
Tag	IsScreenOpen("main") // Is "main" screen open?
Tag	IsScreenOpen("main", 10) // Is "main" screen with ID 10 open?

19.14.13 IsTaskRunning

Überprüft, ob ein Hintergrundtask ausgeführt wird.

Anhang: WinStudio-interne Skriptsprache

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
IsTaskRunning	Modulaktivität	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
IsTaskRunning(strTask)
```

Eingaben

strTask

Der Name des zu überprüfenden Tasks.

Rückgabewert

0	Task wird nicht ausgeführt.
1	Task wird ausgeführt.

Beispiele

Läuft der Projekt-Viewer (**Viewer.exe**)?

IsTaskRunning("Viewer")

19.14.14 IsViewerInFocus

Überprüft, ob der Projekt-Viewer ("Viewer.exe") auf dem Bildschirm aktiviert ist.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
IsViewerInFocus	Modulaktivität	synchron	unterstützt	nicht unterstützt	nicht unterstützt

Syntax

```
IsViewerInFocus()
```

Diese Funktion hat keine Parameter.

Rückgabewert

0	Viewer ist nicht aktiviert.
1	Viewer ist aktiviert.

Beispiele

IsViewerInFocus - Beispiele

```
IsViewerInFocus()
```

19.14.15 KeyPad

Zeigt eine virtuelle Tastatur an, mit der der Benutzer einen Tag-Wert eingeben kann.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
KeyPad	Modulaktivität	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
KeyPad("strTagName",optStrKeyboardName,optNumIsPassword,optStrHint,optNumMin,optNumMax,optNumESign,optStrConfirmation)
```

Eingaben

strTagName

Der Name des Tags, welcher mit Hilfe der "Virtuellen Tastatur" beschrieben werden soll

Dieser Name muss in Anführungszeichen gesetzt werden; anderenfalls versucht das Projekt, den Inhalt des Tags abzurufen.

optStrKeyboardName

Art der zu öffnenden "Virtuellen Tastatur" (z.B. "Alphanumerisch", "Erweitertes Tastenfeld" oder "Tastenfeld"). Wird dieser Parameter ausgelassen, wird die "virtuelle Standardtastatur" geöffnet.

optNumIsPassword

Wenn dieser Parameter einen Wert ungleich 0 (Null) enthält, erscheinen die in der "virtuellen Tastatur" eingegebenen Zeichen als Sternchen (" "*""). Diese Option ist hilfreich bei der Passworteingabe durch den Benutzer.

optStrHint

Der für diesen Parameter angegebene Wert wird in der Titelleiste des Fensters "Virtuelle Tastatur" angezeigt, wenn die Option **Tipp anzeigen** in den Projekteinstellungen, Seite 75 ausgewählt ist.

optNumMin, optNumMax

Numerische Minimal- und Maximalwerte für den Tag bei Verwendung der Tastaturart "Tastenfeld", wenn die Option **Anzeigen Felder MIN/MAX** in den Projekteinstellungen, Seite 75 ausgewählt ist. (Diese Werte werden für alle anderen Tastaturarten ignoriert.) Diese Parameter sind optional. Zur Implementierung müssen aber beide angegeben werden. Wird nur ein Parameter angegeben – z.B. Min aber nicht Max – wird dieser ignoriert.

optNumESign

Wird dieser Parameter mit einem Wert ungleich 0 (Null) belegt, wird der Benutzer aufgefordert, sich zum Vervollständigen der Eingabe anzumelden und die Aktion wird im Protokoll vermerkt.

optStrConfirmation

Der für diesen Parameter festgelegte Wert wird in einem Bestätigungsdialog angezeigt, den der Benutzer zum Abschließen der Eingabe bestätigen muss.



Die Bestätigung kann nicht automatisiert oder umgangen werden; der Dialog kann nur durch einen tatsächlichen Tastendruck oder Mausklick durch den Benutzer bestätigt werden.

Dieser Parameter ist optional. Ist kein Wert angegeben, wird kein Dialog angezeigt.

Rückgabewert

Fehler	Beschreibung
0	Erfolgreich
1	Fehler
2	Tag existiert nicht
3	Fehler Eintrittsvarianz, Funktion wird bereits ausgeführt
4	Ungültige Anzahl Parameter
5	Interner Fehler, weitere Informationen beim technischen Support erfragen

Beispiele

Tag-Name	Ausdruck
Tag	KeyPad("tagA")
Tag	KeyPad("tagA", "EnhKeypad")
Tag	KeyPad("tagA", "EnhKeypad", 1)
Tag	KeyPad("tagA", "EnhKeypad", 1, "My Input", 0, 100)

19.14.16 LogOff

Diese Funktion meldet den aktuellen Benutzer ab und meldet dann den Standardbenutzer an (normalerweise „Guest“).

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
LogOff	Modulaktivität	asynchron	unterstützt	unterstützt	unterstützt

Syntax

```
LogOff()
```

Diese Funktion hat keine Parameter.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte aus:

Wert	Beschreibung
0	Fehler
1	Erfolgreich

Beispiele

LogOff - Beispiele

```
LogOff()
```

19.14.17 LogOn

Mit dieser Funktion wird entweder ein bestimmter Benutzer angemeldet oder ein "Anmelde"-Dialog angezeigt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
LogOn	Modulaktivität	asynchron	unterstützt	unterstützt	unterstützt

Syntax

```
LogOn({ | optStrUsername,optStrPassword })
```

Eingaben

optStrUsername

Name des anzumeldenden Benutzers.

optStrPassword

Passwort des angegebenen Benutzers.

"optStrUsername" und "optStrPassword" sind optionale Parameter. Werden sie nicht angegeben, zeigt das Projekt stattdessen einen Anmeldedialog, Seite 432 an, um den aktuellen Bediener der Station zum Anmelden aufzufordern.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte zurück:

Wert	Beschreibung
0	Fehler (z.B. Benutzername oder Passwort ungültig) oder Abbruch.
1	erfolgreich

Beispiele

Anzeigen eines Anmeldedialogs:

LogOn()

Benutzername "Albert" mit Passwort "EMC2" anmelden:

LogOn("Albert", "EMC2")

19.14.18 Math

Führt das angegebene Mathematikarbeitsblatt aus.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Berechnung	Modulaktivität	synchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
Math( numWorksheet )
```

Eingaben

numWorksheet

Nummer des auszuführenden Mathematikarbeitsblattes.

Rückgabewert

Gibt keine Werte zurück.

Beispiele

Tag-Name	Ausdruck
	Math(6)

HINWEIS

Das Ausführen eines Mathematikarbeitsblatts aus einem anderen Modul heraus hält das Modul an, bis das Mathematikarbeitsblatt fertig ist.

Daher sollte diese Funktion nur wenn unbedingt erforderlich verwendet werden, um eine Beeinträchtigung der Leistung der anderen Module zu vermeiden.

19.14.19 PostKey

Sendet einen Tastaturcode an das aktuell angezeigte WinStudio-Bild.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
PostKey	Modulaktivität	synchron	unterstützt	unterstützt	unterstützt

Diese Funktion kann nicht mit Tasks, Seite 61 oder im Skript Globale Prozeduren, Seite 799 verwendet werden.

Syntax

```
PostKey(numKeydownOrKeyup, numwParam, numlParam)
```

Eingaben

numKeydownOrKeyup

Numerischer Tag mit dem Wert 0 (für das Ereignis Taste drücken) oder 1 (für das Ereignis Taste lösen).

numwParam

Numerischer Tag mit dem zu sendenden Tastaturcode.

numlParam

Numerischer Tag mit dem Wert des Parameters "lParam" einer Windows-Message.

Rückgabewert

Gibt keine Werte zurück.

Beispiele

Tag-Name	Ausdruck
	PostKey(0, 0x24, 0)

19.14.20 Recipe

Führt das angegebene Rezepturarbeitsblatt aus.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Rezeptur	Modulaktivität	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Recipe( strFunction )
```

Eingaben

strFunction

String-Tag zur Festlegung der auszuführenden Aktion und des zu verwendenden Rezepturarbeitsblatts

im Format **[Aktion]:[Rezepturblatt]**.

Aktionen:

Save	Speichert Daten in einer Datendatei.
Load	Lädt Daten aus einer Datendatei.
Delete	Löscht eine Datendatei.
Init	Initialisiert eine Datendatei mit dem Wert 0 in allen Tags.

Rückgabewert

0	Kein Fehler
1	Wenn der Tag numerisch ist
2	Ausdruck enthält nicht ":"
3	Vorangehender Befehl zum ungültigen ":"
4	Task vom System nicht gefunden
5	Festplattenfehler

Beispiele

Tag-Name	Ausdruck
Tag	Recipe("Save:Recipe1")
Tag	Recipe("Load:Recipe5")

- Zum Ausführen der Rezepturfunktionen muss der Hintergrundtask ausgeführt werden.
- Wird diese Funktion auf einem Thin Client aufgerufen, wird der Befehl an den Server gesendet (über TCP/IP) und der Task, Seite 361 Rezeptur, Seite 361 auf dem Server führt den Befehl aus. Daher sollten Sie beachten, dass mit dem Gültigkeitsbereich **Lokal** anstelle von **Server** konfigurierte Tags trotzdem noch auf dem Server aktualisiert werden.

19.14.21 Report

Führt das angegebene Berichtsarbeitsblatt aus und sendet das Ergebnis an die Festplatte, den Drucker oder eine PDF-Datei.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Bericht	Modulaktivität	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Report( "strFunction" , optNumOrientation )
```

Eingaben

strFunction

Zeichenkette zur Festlegung des durchzuführenden Ausgabetyps und des auszugebenden Berichtsarbeitsblatts, gemäß der Syntax

"Ausgabetyp:Dateiname", wobei…

Ausgabetyp entweder **Disk** (sichert Datendatei auf die Festplatte), **Prn** (sendet Bericht an Standarddrucker) oder **Pdf** (erzeugt eine PDF-Datei des Berichts) ist und **Dateiname** der Name der auszugebenden Berichtarbeitsblattdatei (***.rep**) ist.

optNumOrientation

Stellen Sie die Seitenausrichtung wie folgt ein: "0" (Standard) ist Hochformat, "1" ist Querformat. Dieser Parameter wird ignoriert, wenn **Vorgang** nicht **Prn** ist.

Einige Merkmale dieser Funktion werden bei Ausführen des Projekts auf einem Windows-CE-Gerät nicht unterstützt: es können keine PDFs erzeugt werden, die Seitenausrichtung kann nicht mit dem Parameter **optNumOrientation** eingestellt werden und es werden keine Berichtsarbeitsblätter im RTF-Format unterstützt.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	strFunction ist mit einem numerischen Wert (ungültig) konfiguriert
2	strFunction enthält nicht "":"" (ungültig)
3	strFunction enthält einen ungültigen Ausgabentyp vor dem "":""

Wert	Beschreibung
4	Hintergrundtask wird nicht ausgeführt (siehe Tipp unten)
5	Festplattenfehler (z.B. Festplatte voll, schreibgeschützte Datei kann nicht überschrieben werden oder ungültiger Pfad)
6	Angegebenes Berichtsarbeitsblatt ist nicht vorhanden



Zum Ausführen dieser Funktion muss der Hintergrundtask ausgeführt werden. Anderenfalls gibt die Funktion den Fehlercode "4" zurück. Weitere Informationen finden Sie unter <Task-Ausführung>.

Beispiele

Tag-Name	Ausdruck
	Report("Disk:Report1.rep")
	Report("Prn:Report2.rep", 0)
	Report("Prn:Report3.rep", 1)
	Report("Pdf:Report1.rep")

19.14.22 RunGlobalProcedureAsync

Diese Funktion führt eine globale Prozedur asynchron in einem eigenen Thread aus, so dass andere ausgeführte Skripte nicht beeinträchtigt werden. Diese Prozedur wird auf dem Projektserver ausgeführt, kann aber von jedem lokalen oder Remote-Client aufgerufen werden.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
RunGlobalProcedureAsync	Modulaktivität	asynchron	unterstützt	unterstützt	nicht unterstützt, siehe Hinweis

Syntax

```
RunGlobalProcedureAsync(strProcedureName{ | ,optStrArgument1,…,optStrArgumentN })

RunGlobalProcedureAsync( strProcedureName )
RunGlobalProcedureAsync( strProcedureName, optStrArgument1, …, optStrArgumentN )
```

Eingaben

strProcedureName

Der Name der Prozedur (d.h. eine in Prozeduren, Seite 799 definierte VBScript-Funktion oder -Subroutine) zur asynchronen Ausführung.

optStrArgument1, …, optStrArgumentN

Werte zur Weitergabe an die Parameter der Prozedur. Argumente müssen als Zeichenketten weitergegeben werden.

Rückgabewert

Wenn die Prozedur erfolgreich ausgeführt wurde, gibt diese Funktion eine Thread-Kennung zurück, die mit der Funktion RunGlobalProcedureAsyncGetStatus, Seite 637 verwendet werden kann. Andernfalls gibt diese Funktion einen Fehlercode aus:

Wert	Beschreibung
-1	Funktion wird vom Viewer/Web Thin Client nicht unterstützt.
-2	Ungültige Anzahl Parameter. Es muss mindestens der Name der Prozedur angegeben werden.
-3	Maximale Anzahl der Threads überschritten. Siehe Hinweis
-4	VBScript-Parameter konnten nicht zur Ausführung kompiliert werden.
-5	Thread-Ausführung konnte nicht gestartet werden.
-100	Interner Fehler. Bitte wenden Sie sich an den technischen Support.



Es muss unbedingt beachtet werden, dass diese Funktion nur von Hintergrundtasks auf dem Projektserver aufgerufen werden kann (z.B. Mathmatik, Skript, Scheduler). Sie kann nicht direkt durch einen Projekt-Client aufgerufen werden, auch wenn der Projekt-Client auf der gleichen Arbeitsstationen oder dem gleichen Gerät wie der Projektserver ausgeführt wird, weil der Client-Prozess nur über einen Thread verfügt. Zum indirekten Aufrufen der Funktion aus einem Projekt-Client konfigurieren Sie ein Arbeitsblatt Berechnung, Seite 367 oder Skript, Seite 368, das über eine Freigabe (z.B. Variable) ausführbar ist. Durch die Projektierung eines Eingabeobjekts in einem Bild kann die Ausführung des Programmcodes manuell gestartet werden.

Außerdem kann die maximale Anzahl an VBScript-Threads, die asynchron ausgeführt werden können, durch manuelles Bearbeiten der Projektdatei (d.h. **projektname.APP**) und Ändern der folgenden Einstellung konfiguriert werden:

[Script]

MaxAsyncThreads=8

Die Voreinstellung beträgt 8 Threads. Die einzige Beschränkung liegt in den Systemressourcen. Das Erhöhen der Anzahl der Threads kann die Laufzeitleistung beeinträchtigen.

Beispiele

Unter Annahme der folgenden im Ordner Prozeduren definierten Prozedur...

RunGlobalProcedureAsync - Beispiele

```
Function AddMe(intNumber)
  If intNumber >= 6 Then
    AddMe = 0
  Else
    AddMe = intNumber + 2
  End If
End Function
```

...wird die Prozedur durch Aufrufen der Funktion **RunGlobalProcedureAsync** ausgeführt...

RunGlobalProcedureAsync("AddMe", "2")

...und die Funktion gibt eine Thread-Kennung aus, die mit der Funktion **RunGlobalProcedureAsyncGetStatus** verwendet werden kann.

19.14.23 RunGlobalProcedureAsyncGetStatus

Diese Funktion ruft den Status einer oder mehrerer globaler Prozeduren ab, die asynchron durch Aufrufen der Funktion "RunGlobalProcedureAsync" ausgeführt wurden. Jede Prozedur wird in einem eigenen Thread ausgeführt, so dass andere Threads nicht beeinträchtigt werden.

Anhang: WinStudio-interne Skriptsprache

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
RunGlobalProcedureAsyncGetStatus	Modulaktivität	synchron	unterstützt	unterstützt	nicht unterstützt, siehe Hinweis

Syntax

```
RunGlobalProcedureAsyncGetStatus({ | optNumThreadID |"optTagThreadIDs","optTagStatus","optTagParameters"})

RunGlobalProcedureAsyncGetStatus( )
RunGlobalProcedureAsyncGetStatus( optNumThreadID )
RunGlobalProcedureAsyncGetStatus("optTagThreadIDs", "optTagStatus", "optTagParameters")
```

Eingaben

optNumThreadID

Die von der Funktion RunGlobalProcedureAsync, Seite 636 bei erfolgreicher Ausführung der Prozedur zurückgegebene Thread-Kennung.

optTagThreadIDs

Der Name eines Array-Tags, der die Thread-Kennung aller aktuell ausgeführten und kürzlich beendeten Threads erhält.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

optTagStatus

Der Name eines Array-Tags, der die Zustände aller aktuell ausgeführten und kürzlich beendeten Threads erhält.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

optTagParameters

Der Name eines Array-Tags, der die Parameter aller aktuell ausgeführten und kürzlich beendeten Threads erhält.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert

Wenn die Funktion erfolgreich ist, hängen die möglichen Rückgabewerte davon ab, wie die Funktion aufgerufen wurde:

- Wenn die Funktion ohne Parameter aufgerufen wurde...

  RunGlobalProcedureAsyncGetStatus( )

  ...dann ist der Rückgabewert die Gesamtzahl der aktuell ausgeführten Threads.

- Wenn die Funktion nur mit dem Parameter **optNumThreadID** aufgerufen wurde...

  RunGlobalProcedureAsyncGetStatus(optNumThreadID)

...dann ist der Rückgabewert entweder "0", zur Anzeige, dass der Thread noch ausgeführt wird, oder der von der Prozedur ausgegebene Wert.

- Wenn die Funktion mit den Array-Tags aufgerufen wurde...

  RunGlobalProcedureAsyncGetStatus(" optTagThreadIDs ", " optTagStatus ", " optTagParameters")

  ...dann erhalten die Tags die entsprechenden Werte für alle aktuell ausgeführten und kürzlich beendeten Threads.

Wenn die Funktion fehlschlägt, gibt sie einen der folgenden Fehlercodes zurück:

Wert	Beschreibung
-1	Funktion wird vom Viewer/Web Thin Client nicht unterstützt.
-2	Ungültige Thread-Kennung.
-3	Ungültige optTagThreadIDs.
-4	Ungültiger optTagStatus.
-5	Ungültige optTagParameter.
-100	Interner Fehler. Bitte wenden Sie sich an den technischen Support.



Es muss unbedingt beachtet werden, dass diese Funktion nur von Hintergrundtasks auf dem Projektserver aufgerufen werden kann (z.B. Mathematik, Skript, Scheduler). Sie kann nicht direkt durch einen Projekt-Client aufgerufen werden, auch wenn der Projekt-Client auf der gleichen Arbeitsstation oder dem gleichen Gerät wie der Projektserver ausgeführt wird, weil der Client-Prozess nur über einen Thread verfügt. Zum indirekten Aufrufen der Funktion aus einem Projekt-Client konfigurieren Sie ein Arbeitsblatt Mathematik, Seite 367 oder Skript, Seite 368, das über eine Freigabe (z.B Variable) ausführbar ist. Durch die Projektierung eines Eingabeobjekts in einem Bild kann die Ausführung des Programmcodes manuell gestartet werden.

Außerdem wird bei erfolgreichem Aufruf von **RunGlobalProcedureAsync** eine Kennung für den erzeugten Thread ausgegeben und die Prozedur wird in diesem Thread ausgeführt. Der Status des Threads wird in einem internen Puffer gespeichert und kann mithilfe der Funktion **RunGlobalProcedureAsyncGetStatus** abgerufen werden. Der Puffer wird gelöscht, wenn:

- die Funktion **RunGlobalProcedureAsyncGetStatus** aufgerufen wurde und der Thread-Status ungleich "0" ist (Thread wird ausgeführt) oder
- die maximale Puffergröße überschritten wurde, der Thread nicht mehr ausgeführt wird und ein Aufruf für das Starten eines neuen Threads ausgeführt wurde.

19.14.24 RunGlobalProcedureOnFalse

Diese Funktion führt direkt eine globale Prozedur aus, wenn ein bestimmter Tag/Ausdruck den Wert FALSE annimmt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
RunGlobalProcedureOnFalse	Modulaktivität	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
RunGlobalProcedureOnFalse("strCondition",strProcedureOnFalse)
```

Eingaben

strCondition

Ein Projekt-Tag oder ein Ausdruck

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

strProcedureOnFalse

Der Name der Prozedur (d.h. eine in Prozeduren, Seite 799 definierte VBScript-Funktion oder -Subroutine) die ausgeführt wird, wenn **strCondition** den Wert FALSE (Null) annimmt.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte zurück:

Wert	Beschreibung
0	Fehler
1	Erfolgreich

☞ Wenn diese Funktion aufgerufen wird, bleibt sie aktiv, bis die Laufzeitumgebung beendet wird. Das bedeutet, dass die Prozedur jedes Mal ausgeführt wird, wenn der Tag/Ausdruck den Wert FALSE annimmt. Wenn der Tag/Ausdruck den Wert FALSCH annimmt, wird die Funktion jedoch nicht kontinuierlich, sondern nur einmal ausgeführt, "während" der Tag-Ausdruck den Wert FALSE hat.

Die Prozedur wird auf dem Client ausgeführt. Zum Ausführen einer Prozedur auf dem Server wird die Funktion RunGlobalProcedureOnServer, Seite 640 verwendet.

Beispiele

Wenn die Variable **TagOnFalse** den Wert FALSE (Null) annimmt, wird die Prozedur **UsingOnFalse** ausgeführt:

RunGlobalProcedureOnFalse("TagOnFalse", "UsingOnFalse")

19.14.25 RunGlobalProcedureOnServer

Diese Funktion führt direkt eine globale Prozedur mit Parametern aus. Diese Prozedur wird auf dem Projektserver ausgeführt, kann aber von jedem lokalen oder Remote-Client aufgerufen werden.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
RunGlobalProcedureOnServer	Modulaktivität	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
RunGlobalProcedureOnServer(strProcedureName{ | ,optStrArgument1,…,optStrArgumentN })
RunGlobalProcedureOnServer( strProcedureName, optStrArgument1, …, optStrArgumentN )
```

Eingaben

strProcedureName

Der Name der Prozedur (d.h. eine in Prozeduren, Seite 799 definierte VBScript-Funktion oder -Subroutine), die auf dem Server ausgeführt werden soll.

optStrArgument1, …, optStrArgumentN

Werte zur Weitergabe an die Parameter der Prozedur. Argumente müssen als Zeichenketten (String) weitergegeben werden.

Rückgabewert

Diese Funktion gibt den von der aufgerufenen Prozedur zurückgegebenen Wert zurück.

Beispiele

Unter Annahme der folgenden im Ordner Prozeduren definierten Prozedur...

RunGlobalProcedureOnServer - Beispiele

```
Function AddMe(intNumber)
  If intNumber >= 6 Then
    AddMe = 0
  Else
    AddMe = intNumber + 2
  End If
End Function
```

...wird die Prozedur durch Aufrufen der Funktion **RunGlobalProcedureOnServer** ausgeführt...

RunGlobalProcedureOnServer("AddMe", "2")

...und gibt den Wert 4 zurück.

19.14.26 RunGlobalProcedureOnTrigger

Diese Funktion führt direkt eine globale Prozedur aus, wenn sich der Wert eines bestimmten Tags ändert.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
RunGlobalProcedureOnTrigger	Modulaktivität	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
RunGlobalProcedureOnTrigger("strTagName",strProcedureOnTrigger)
```

Eingaben

strTagName

Name eines Projekt-Tags, welches bei einer Wertänderung eine Funktion aufruft.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

strProcedureOnTrigger

Der Name der Prozedur (d.h. eine in Prozeduren, Seite 799 definierte VBScript-Funktion oder -Subroutine), die ausgeüfhrt werden soll, wenn sich der Wert des angegebenen Tags ändert.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte zurück:

Wert	Beschreibung
0	Fehler
1	Erfolgreich

☞ Wenn diese Funktion aufgerufen wird, bleibt sie aktiv, bis das Laufzeitsystem beendet wird. Das bedeutet, dass die Prozedur jedes Mal ausgeführt wird, wenn sich der Wert des Tags ändert.

Die Prozedur wird auf dem Client ausgeführt. Zum Ausführen einer Prozedur auf dem Server wird die Funktion RunGlobalProcedureOnServer, Seite 640 verwendet.

Beispiele

Wenn sich der Wert von der Variablen **TagTrigger** ändert, wird die Prozedur **UsingTrigger** ausgeführt:

RunGlobalProcedureOnTrigger("TagTrigger", "UsingTrigger")

19.14.27 RunGlobalProcedureOnTrue

Diese Funktion führt direkt eine globale Prozedur aus, wenn ein bestimmter Tag/Ausdruck den Wert TRUE annimmt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
RunGlobalProcedureOnTrue	Modulaktivität	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
RunGlobalProcedureOnTrue("strCondition",strProcedureOnTrue)
```

Eingaben

strCondition

Ein Projekt-Tag oder ein Ausdruck zum Auslösen der Funktion. Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

strProcedureOnTrue

Der Name der Prozedur (d.h. eine in Prozeduren, Seite 799 definierte VBScript-Funktion oder -Subroutine), die ausgeführt werden soll, wenn "strCondition" den Wert TRUE (nicht Null) annimmt.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte zurück:

Wert	Beschreibung
0	Fehler
1	Erfolgreich

Wenn diese Funktion aufgerufen wird, bleibt sie aktiv, bis das Laufzeitsystem beendet wird. Das bedeutet, dass die Prozedur jedes Mal ausgeführt wird, wenn der Tag/Ausdruck "strCondition" den Wert TRUE annimmt. Wenn der Tag/Ausdruck den Wert TRUE annimmt, wird die Funktion jedoch nicht kontinuierlich, sondern nur einmal ausgeführt.

Die Prozedur wird auf dem Client ausgeführt. Zum Ausführen einer Prozedur auf dem Server wird die Funktion RunGlobalProcedureOnServer, Seite 640 verwendet.

Beispiele

Wenn der Wert der Variablen **TagOnTrue** den Wert TRUE (nicht Null) annimmt, wird die Prozedur **UsingOnTrue** ausgeführt:

RunGlobalProcedureOnTrue("TagOnTrue", "UsingOnTrue")

19.14.28 RunVBScript

Führt eine Anweisung in der VBScript-Sprache aus.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
RunVBScript	Modulaktivität	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
RunVBScript ( strScript, "optTagReturnError" )
```

Eingaben

strScript

Skript-Anweisung, die von der Funktion ausgeführt werden soll

optTagReturnError

Name des Tags, der ggf. von der Anweisung erzeugten Fehler erhält (z.B. „Division durch Null“). Der Tag vom Typ String muss in Anführungszeichen stehen.

Rückgabewert

Wert	Beschreibung
0	Fehler
1	Erfolgreich

Beispiele

Tag-Name	Ausdruck
TagResult	**RunVBScript("MsgBox(Time)")** // Executes the MsgBox function page 827 from VBScript and displays the current time.
	RunVBScript(TagStatement) // Executes the statement configured in the value of the string tag **TagStatement**.
	RunVBScript("$TagC=$TagA/$TagB", "TagError") // Writes in **TagC** the result of **TagA** divided by **TagB**. The error generated by the operation (if any) is written to the string tag **TagError** (e.g., "Division by zero").



Diese Funktion ist hilfreich, um VBScript-Anweisungen in Tasks auszuführen, die nur die interne Sprache unterstützen (z.B. Scheduler-Gruppen). Es können außerdem in den globalen Prozeduren erstellte VBScript-Funktionen aufgerufen werden.



Die Station muss die in dieser Funktion konfigurierten VBScript-Anweisungen unterstützen, um sie ausführen zu können.

19.14.29 SecureViewerReload

"SecureViewerReload" ist eine interne Skript-Funktion, die das Programm "Secure Viewer" schließt und es dann mit einer anderen Konfigurationsdatei neu lädt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SecureViewerReload	Modulaktivität	synchron	nicht unterstützt	nicht unterstützt	nur Secure Viewer

Syntax

```
SecureViewerReload(strFileName)
```

Eingaben

strFileName

Der Dateipfad einer INI-Datei (***.ini**), welche die neue Konfiguration beschreibt. (Wenn sich die Datei im gleichen Ordner wie **Viewer.exe** befindet, wird nur der Dateiname benötigt.) Die Datei sollte die gleiche Struktur haben und die gleichen Einstellungen enthalten wie die Standardkonfigurationsdatei (**Viewer.ini**).

Dieser Parameter kann ein String-Tag oder ein konstanter Text in Anführungszeichen sein.

Rückgabewert Die Funktion gibt keinen Wert zurück.

Beispiele *SecureViewerReload - Beispiele*

```
SecureViewerReload( configFile1 )

SecureViewerReload( "C:\Program Files\Secure Viewer\Bin\Config1.ini" )
```

Siehe auch

- Konfigurieren des Thin Client, Seite 473

19.14.30 SendKeyObject

Sendet Tastaturcodes an Objekte im aktuell angezeigten WinStudio-Bild. Diese Funktion kann zum Auslösen von Aktionen auf dynamischen Bildobjekten verwendet werden.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SendKeyObject	Modulaktivität	synchron	unterstützt	nicht unterstützt	nicht unterstützt

Diese Funktion kann nicht mit Tasks, Seite 61 oder im Skript Globale Prozeduren, Seite 799 verwendet werden.

Syntax

```
SendKeyObject( numEvent, strMainKey, numShift, numCtrl, numAlt, strTargetScreen, optNumID)
```

Eingaben

numEvent

Der Ereigniscode für „Beim Drücken“, „Während gedrückt“ oder „Beim Loslassen“ der Befehlsanimation, Seite 243, wie folgt:

0	Beim Drücken
1	Während gedrückt
2	Beim Loslassen



Bei der Verwendung des Events "Beim Drücken" ist folgendes zu beachten: Bei jedem Aufruf der Funktion **SendKeyObject()** sendet WinStudio den parametrierten Tastencode genau ein mal. Aus diesem Grund werden alle Aktionen, welche über das Tastencode ausgelöst werden, nur einmal gestartet. Dies gilt auch für Aktionen die in einem Bildobjekt im Registerblatt "Während" programmiert sind.

strMainKey

Tastaturcodes zum Senden an ein WinStudio-Bild. Zulässig sind folgende Werte:

- "F1" … "F20"
- "+ "
- "-"
- "/"
- "*"
- "HOME"
- "END"
- "INSERT"
- "DELETE"
- "DOWN"
- "UP"
- "LEFT"
- "RIGHT"
- "PAGEUP"
- "PAGEDOWN"
- "SPACE"
- "RETURN"
- "BACKSPACE"
- "ESCAPE"
- "A" … "Z"

numShift

Ein numerischer Wert oder Tag, der definiert, ob die SHIFT-Taste gedrückt ist ("0": ja, "1": nein).

numCtrl

Ein numerischer Wert oder Tag, der definiert, ob die CTRL-Taste gedrückt ist ("0": ja, "1": nein).

numAlt

Ein numerischer Wert oder Tag, der definiert, ob die ALT-Taste gedrückt ist"0": ja, "1": nein).

strTargetScreen

Der Name des WinStudio-Bildes, das den Tastaturcode erhält.

☞ Die Parameter "numShift", "numCtrl", "numAlt" und "strTargetScreen" sind optional. Wenn allerdings einer der Parameter konfiguriert wird, müssen die anderen auch konfiguriert werden.

optNumID

Die spezifische Instanznummer des Bildes. (Die ID wird zugewiesen, wenn das Bild mit der Funktion Open(), Seite 604 geöffnet wird.) Dieser Befehlsparameter ist optional; die Standardkennung ist 0.

Rückgabewert

Diese Funktion gibt keine Werte zurück.

Beispiele

Tag-Name	Ausdruck
	SendKeyObject(0, "R", 1, 0, 0, "main", 10) // Sends Shift-R to the "main" screen with ID 10.

19.14.31 SetAppPath

Legt den neuen Dateipfad für den Projektordner fest. Nach dem Ausführen dieser Funktion sucht "WinStudio" alle Projektdateien (d.h. Bilder, Alarme, Trends, Datenbank, Ereignisse) in diesem Ordner.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetAppPath	Modulaktivität	asynchron	unterstützt	unterstützt	unterstützt

Syntax

```
SetAppPath(strPath)
```

Eingaben

strPath

String-Variable oder konstanter Text in Anführungszeichnen mit dem Dateipfad

Rückgabewert

0	Pfad konnte nicht gesetzt werden.
1	Pfad wurde erfolgreich gesetzt

Beispiele

Tag-Name	Ausdruck
	SetAppPath("C:\Studio\")

Wenn der Rechner sich in einem Netzwerk befindet, kann die Syntax **//IP-Adresse oder Host-Name/-Pfad** zum Definieren eines Speicherorts auf einem anderen Netzlaufwerk verwendet werden.

19.14.32 SetViewerInFocus

"SetViewerinFocus" ist eine interne Skript-Funktion, die den Viewer in den Vordergrund (d.h. vor alle anderen geöffneten Fenster) bringt und ihn dann auf Bildschirmgröße vergrößert.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetViewerInFocus	Modulaktivität	synchron	unterstützt	nicht unterstützt	unterstützt

Syntax

```
SetViewerInFocus()
```

Keine Parameter vorhanden.

Rückgabewert

Diese Funktion gibt keinen Wert zurück.

Mit der Einführung von Windows XP verhindern Systemsicherheitsfunktionen, dass Programmfenster sich selbständig ohne Benutzereingabe vor andere Fenster setzen. Beim Aufruf dieser Funktion macht die Windows-Taskleiste durch Blinken auf die Aktivierung des Fensters aufmerksam. (Anti-Viren-Programme können diesen Mechanimus ggf. als verdächtig anzeigen.) Das Programmfenster wird erst in den Vordergrund geblendet, wenn der Benutzer das Programm auswählt.

Um dies zu umgehen, müssen Sie **SetViewerInFocus** mindestens einmal im Startup-Skript, Seite 369 Ihres Projekts aufrufen. Nach dem kompletten Hochlauf des Laufzeitsystems und einem erneuten Aufruf der Funktion **SetViewerInFocus** wird der Viewer im Vordergrund eingeblendet.

Beispiele

SetViewerInFocus - Beispiele

```
SetViewerInFocus()
```

Siehe auch

- Projekteinstellungen – Registerkarte Laufzeit-Desktop, Seite die 75
- StartupSkript, Seite 369

19.14.33 SetViewerPos

Legt Höhe, Breite und Position des Projekts-Viewers oder Thin Clients fest.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetViewerPos	Modulaktivität	synchron	unterstützt	nicht unterstützt	unterstützt

Syntax

```
SetViewerPos( numLeft, numTop, optNumWidth, optNumHeight )
```

Eingaben

numLeft

Ein numerischer Tag, der die linke Randposition des Viewers in Pixeln angibt.

numTop

Ein numerischer Tag, der die obere Randposition des Viewers in Pixeln angibt.

optNumWidth

"Optionaler" numerischer Tag, der die Breite des Viewers in Pixeln angibt.

optNumHeight

"Optionaler" numerischer Tag, der die Höhe des Viewers in Pixeln enthält.

Rückgabewert

0	Fehler
1	Erfolgreich

Beispiele

Tag-Name	Ausdruck
Tag	SetViewerPos(50, 50, 640, 480)

Wenn die optionalen Parameter (**numWidth** und **numHeight**) weggelassen werden, wird der Viewer in der Standardgröße aus den Projekteinstellungen geöffnet.

19.14.34 ShutDown

Schließt alle aktiven Module der Runtimeumgebung.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ShutDown	Modulaktivität	asynchron	unterstützt	unterstützt	nicht unterstützt

Syntax

```
ShutDown()
```

Diese Funktion hat keine Parameter.

Rückgabewert

Die Funktion gibt keine Werte zurück.



Diese Funktion schließt nicht <Entwicklungsumgebung>, Datenbankspion, Seite 93 oder LogWin, Seite 93.

Beispiele

Tag-Name	Ausdruck
	ShutDown()

19.14.35 StartTask

Startet ein Runtime-Modul, das noch nicht ausgeführt wird.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
StartTask	Modulaktivität	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
StartTask(strTask)
```

Eingaben

strTask

Name des zu startenden Tasks (aus folgender Liste):

Wert	Beschreibung
BGTask	Hintergrund-Tasks
Viewer	Viewer
DBSpy	Datenbankspion
LogWin	LogWin
Driver	Driver
UniDDECl	DDE-Client
UniNDDE	DDE-Server
UniODBC	ODBC
TCPServer	TCP/IP-Server
TCPClient	TCP/IP-Client
OPCClient	OPC-Client

Rückgabewert

Die Funktion gibt keine Werte zurück.

Beispiele

Tag-Name	Ausdruck
	StartTask("Viewer")

☞ Die Funktion **StartTask("Driver")** startet alle in diesem Projekt konfigurierten Treiber. Zum Starten eines bestimmten Treibers muss die Funktion WinExec(), Seite 651 verwendet werden.

19.14.36 ViewerPostMessage

Sendet eine Windows-Systemmeldung an das angegebene WinStudio-Bild.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ViewerPostMes-sage	Modulaktivität	asynchron	unterstützt	unterstützt	unterstützt

☞ Diese Funktion kann nicht mit Tasks, Seite 61 oder im Skript Globale Prozeduren, Seite 799 verwendet werden.

Syntax

```
ViewerPostMessage( strScrTitle, numMessage, numwParam, numlParam, optNumID )
```

Eingaben

strScrTitle

Der Name der Bildes, an das die Meldung gesendet werden soll.

numMessage

Die Nummer der zu sendenden Windows-Systemmeldung.

numwParam

Ein numerischer Wert oder Tag mit einem meldungsspezifischen Wert, der an den Parameter **wParam** der Windows-Systemmeldung weitergegeben wird

numlParam

Ein numerischer Wert oder Tag mit einem meldungsspezifischen Wert, der an den Parameter **lParam** der Windows-Systemmeldung weitergegeben wird.

optNumID

Die spezifische Instanznummer des WinStudio-Bildes. (Die ID wird zugewiesen, wenn das Bild mit der Funktion Open(), Seite 604 geöffnet wird.) Dieser Befehlsparameter ist optional; die Standardkennung ist 0.

Rückgabewert Diese Funktion gibt keine Werte zurück.

Die Funktion emuliert die Funktion PostMessage in Microsoft Windows. Weitere Informationen sowie eine vollständige Liste aller verfügbaren Windows-Systemmeldungen finden Sie im Microsoft Developers Network.

Beispiele

Tag-Name	Ausdruck
	ViewerPostMessage("main", 16, 3, 1, 10)// Sends message 16 to the "main" screen with ID 10.

19.14.37 Wait

Hält die Ausführung des Laufzeitsystems für eine festgelegte Anzahl von Millisekunden an.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Wait	Modulaktivität	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Wait(numMillisec)
```

Eingaben

numMillisec

Ein numerischer Variable oder ein konstanter Wert; die Anzahl der Millisekunden für die Wartezeit.

Rückgabewert Die Funktion gibt keine Werte zurück.

Diese Funktion wird normalerweise nur in Mathematik-Arbeitsblättern, Seite 367 verwendet, sie kann aber auch in jeder VB-Skripten, Seite 797 verwendet werden.

Die Verwendung dieser Funktion in Ihrem Projekt ist gefährlich, weil während des Anhaltens des Laufzeitsystems alle eingehenden Informationen aus gemeinsamen verwendeten Datenbanken und angeschlossenen Geräten ignoriert werden. Eine Anwendung mit dieser Funktion muss gründlich getestet werden.

Beispiele

Tag-Name	Ausdruck
	Wait(1000)

19.14.38 WinExec

"WinExec" ist eine interne Skript-Funktion, die einen Windows-Befehl so ausführt, als wäre dieser in der Eingabeaufforderung eingegeben worden.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
WinExec	Modulaktivität	asynchron	unterstützt	unterstützt (siehe Hinweis)	unterstützt

Syntax

```
WinExec(strCommand{ | ,{ optNumState | 0 | 1 | 2 | 3 | 4 | 7 }{ | ,{ optNumSync | 0 | 1 },
                "optTagReturnOrHandle" } })
```

Eingaben

strCommand

String-Variable oder konstanter Text in Anführungszeichen mit dem Kommando für die Befehlsausführung

optNumState

Anfangszustand des Programms, das durch den Befehl gestartet wird:

Wert	Beschreibung
0	Versteckt das Programm und übergibt die Kontrolle einem anderen Programm.
1	Das Programm wird aktiviert und angezeigt.
2	Das Programm wird aktiviert und als Symbol angezeigt.
3	Das Programm wird aktiviert und maximiert.
4	Das Programm wird in der zuletzt verwendeten Größe angezeigt. Das Programm ist noch aktiv.
7	Zeigt das Programm als Symbol an. Das Programm ist noch aktiv.

Dieser Parameter ist optional. Wenn kein Wert angegeben wird, ist der Standardwert "1".



Dieser Parameter wird auf Windows-CE-Betriebssystemen nicht unterstützt; der übergebene Parameterwert wird hier ignoriert und es wird der Standardwert verwendet.

optNumSync

Parameter zur Festlegung, ob der Befehl synchron oder asynchron ausgeführt wird:

Wert	Beschreibung
0	Asynchrone Ausführung; die Funktion erzeugt sofort einen Rückgabewert.
1	Synchrone Ausführung; die Funktion erzeugt einen Rückgabewert, wenn die Ausführung des Befehls beendet wurde.

Dieser Parameter ist optional. Wenn kein Wert angegeben wird, ist die Standardwert "0".



Um zu überprüfen, ob ein asynchron ausgeführter Befehl beendet ist, verwenden Sie den Parameter "optTagReturnOrHandle" unten und die Funktion WinExecIsRunning, Seite 653.

optTagReturnOrHandle

Der Name eines Projekt-Tags, der den Rückgabewert des Befehls speichert:

- Wenn der Befehl asynchron ausgeführt wird, erhält der Tag einen Handle, der mit der Funktion WinExecIsRunning, Seite 653 ausgeführt werden kann, um zu bestimmen, ob der Befehl noch ausgeführt wird.
- Wenn der Befehl synchron ausgeführt wird, erhält der Tag den Ausgangscode des Befehls. (Dies ist unabhängig vom Rückgabewert der Funktion.)

Dies ist ein optionaler Parameter, es gibt jedoch keinen Standardwert.



Der Tag-Name muss, wie im Syntaxdiagramm gezeigt, in Anführungszeichen stehen. Andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte zurück:

Wert	Beschreibung
0	Befehl wurde nicht erfolgreich ausgeführt.
1	Befehl wurde erfolgreich ausgeführt.

Bitte beachten Sie, dass der Rückgabewert nur anzeigt, ob die Ausführung des Befehls erfolgreich "gestartet" wurde, besonders bei asynchroner Ausführung. Er zeigt nicht an, wann oder wie die Ausführung des Befehls beendet wurde.

Beispiele

Aufrufen von Notepad und direktes Springen zur nächsten Zeile des Skripts:

```
WinExec( "C:\Windows\System32\notepad.exe", 4 )
```

Aufrufen von MSPaint und direktes Springen zur nächsten Zeile des Skripts:

```
WinExec( "C:\Windows\System32\mspaint.exe" )
```

Aufrufen einer Batch-Datei, Ausführen im versteckten Modus, Warten, bis diese beendet ist und Speichern des Ausgangcodes im Tag "return":

```
WinExec( "CMD /C call C:\Temp\MyBatch.bat", 0, 1, "return" )
```

Aufrufen einer VBSript-Datei, Ausführung im versteckten Modus, direktes Fortfahren, Speichern des Handle im Tag "handle":

```
WinExec( "CMD /C call C:\Temp\MyScript.vbs", 0, 0, "handle" )
```



Aufrufen von VBScript-Dateien wird auf Windows-CE-Betriebssystemen nicht unterstützt.

19.14.39 WinExecIsRunning

Prüft, ob ein von der Funktion "WinExec" gestartetes Programm noch ausgeführt wird.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
WinExecIsRunning	Modulaktivität	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
WinExecIsrunning( numHandle, "optTagReturn" )
```

Eingaben

numHandle

Handle-Nummer, die beim Aufruf der Funktion "WinExec" an den Tag "optStrReturnorHandle" zurückgegeben wurde.

"optTagReturn"

Tag mit Rückgabewert des Programms, welches mit Hilfe der **WinExec**-Funktion aufgerufen wurde

Rückgabewert

0	Erfolgreiche Ausführung.
-1	Ungültige(r) Parameter.
-2	Datei konnte nicht geöffnet werden. Festplatte ist schreibgeschützt oder Dateiname ist ungültig.

Beispiele

Tag-Name	Ausdruck
Tag	WinExecIsRunning(numHandle)
Tag	WinExecIsRunning(numHandle, "return")

19.15 Dateifunktionen

19.15.1 Dateifunktionen – Allgemeines

Diese Funktionen werden zum Lesen, Schreiben, Drucken, Verschieben und Löschen von externen Dateien verwendet.

Datei

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
DeleteOlderFiles, Seite 654	synchron	unterstützt	unterstützt	unterstützt
DirCreate, Seite 655	synchron	unterstützt	unterstützt	unterstützt
DirDelete, Seite 655	synchron	unterstützt	unterstützt	unterstützt
DirLength, Seite 656	synchron	unterstützt	unterstützt	unterstützt
DirRename, Seite 657	synchron	unterstützt	unterstützt	unterstützt
FileCopy, Seite 657	synchron	unterstützt	unterstützt	unterstützt
FileDelete, Seite 658	synchron	unterstützt	unterstützt	unterstützt
FileLength, Seite 659	synchron	unterstützt	unterstützt	unterstützt
FileRename, Seite 659	synchron	unterstützt	unterstützt	unterstützt

Anhang: WinStudio-interne Skriptsprache

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
FindFile, Seite 661	synchron	unterstützt	unterstützt	unterstützt
FileWrite, Seite 659	synchron	unterstützt	unterstützt	unterstützt
FindPath, Seite 662	synchron	unterstützt	unterstützt	unterstützt
GetFileAttributes, Seite 662	synchron	unterstützt	unterstützt	unterstützt
GetFileTime, Seite 663	synchron	unterstützt	unterstützt	unterstützt
GetHstInfo, Seite 663	synchron	unterstützt	unterstützt	unterstützt
GetLine, Seite 664	synchron	unterstützt	unterstützt	unterstützt
HST2TXT, Seite 667	asynchron	unterstützt	unterstützt	unterstützt
HST2TXTIsRunning, Seite 668	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
PDFCreate, Seite 671	synchron	unterstützt	nicht unterstützt	unterstützt
Print, Seite 672	asynchron	unterstützt	unterstützt	unterstützt
RDFileN, Seite 673	synchron	unterstützt	unterstützt	unterstützt
WebGetFile, Seite 673	synchron	unterstützt	unterstützt	unterstützt

19.15.2 DeleteOlderFiles

Löscht Dateien aus einem Verzeichnis, die älter als ein Datum sind und der konfigurierten Dateimaske entsprechen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DeleteOlderFiles	Modulaktivität	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DeleteOlderFiles( strPath, strMask, strDate )
```

Eingaben

strPath

Der Pfad zum Ordner der löschenden Dateien.

strMask

Die Maske der zu löschenden Dateien.

strDate

Der Stichtag. Alle Dateien älteren Datums werden gelöscht.



Das Datumsformat (z.B. **JMT** oder **TMJ**) mit den entsprechenden Trennzeichen ("/",":","." usw.) ist in den Projekteinstellungen festgelegt.

Rückgabewert

Gibt die Anzahl aller gelöschten Dateien zurück.

Beispiele

Tag-Name	Ausdruck
Tag	DeleteOlderFiles("C:\Studio\Project\HST\", "*.hst", "04/12/2002")

19.15.3 DirCreate

Erstellt das angegebene Verzeichnis.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DirCreate	Datei	synchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
DirCreate( strDirectory, optBooFullPath )
```

Eingaben

strDirectory

Pfad und Name des zu erstellenden Verzeichnisses.

optBooEmptyOnly

Optionaler Parameter. Wird dieser Parameter weggelassen oder mit dem Wert 0 vorbelegt, wird das Verzeichnis nur erstellt, wenn alle übergeordneten Verzeichnisse vorhanden sind. Wenn dieser Parameter einen Wert ungleich 0 hat, wird der vollständige im Parameter **strDirectory** festgelegte Pfad erstellt.

Rückgabewert

Wert	Beschreibung
-1	Ungültige Parameter
0	Verzeichnis konnte nicht erstellt werden (z.B. Laufwerk ist nicht vorhanden).
1	Verzeichnis wurde erfolgreich erstellt.

Beispiele

Tag-Name	Ausdruck
Tag	DirCreate("C:\Studio\Temp") // The Temp folder is created in the C:\Studio path (only if the C:\Studio path already exists).
Tag	DirCreate("C:\Studio\Temp",1) // The C:\Studio\Temp full path is created.

19.15.4 DirDelete

Löscht das angegebene Verzeichnis.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DirDelete	Datei	asynchron	unterstützt	unterstützt	unterstützt

Syntax

```
DirDelete( strDirectory, optBooEmptyOnly )
```

Eingaben

strDirectory

Pfad und Name des zu löschenden Verzeichnisses.



Dieser Parameter unterstützt Platzhalter ("*" und "?").

optBooEmptyOnly

Optionaler Funktionsparameter. Wenn dieser Parameter den Wert 1 hat, wird das Verzeichnis nur gelöscht, wenn es leer ist. Standardmäßig – d.h. wird der Parameter ausgelassen oder mit dem Wert 0 vorbelegt – wird das Verzeichnis gelöscht, unsabhängig davon ob es leer ist oder nicht.

Rückgabewert

Wert	Beschreibung
−2	Versuch, ein nicht leeres Verzeichnis zu löschen, wenn dieser Vorgang nicht zulässig ist (d.h. **optBooEmptyOnly** ist ungleich 0).
−1	Ungültige Parameter.
0	Verzeichnis konnte nicht gelöscht werden (d.h. Verzeichnis ist nicht vorhanden).
1	Verzeichnis wurde erfolgreich gelöscht.

Beispiele

Tag-Name	Ausdruck
Tag	DirDelete("C:\Studio\Temp") // The Temp folder from C:\Studio is deleted.
Tag	DirDelete("C:\Studio\Temp", 1) // The Temp folder from C:\Studio is deleted only if it is empty.

19.15.5 DirLength

Gibt die Größe eines Verzeichnisses zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DirLength	Datei	synchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
DirLength(strPath)
```

Eingaben

strPath

Pfad des zu prüfenden Verzeichnisses.

Rückgabewert

−2	Verzeichnis nicht vorhanden.
−1	Ungültige Parameter
n	Größe (in Bytes) der Dateien und Unterordner in dem Verzeichnis

HINWEIS

Diese Funktion wird synchron ausgeführt, d.h. das Laufzeitsystem hält an, während es auf die Ausgabe der Funktion wartet.

Wenn das angegebene Verzeichnis sehr große Datenmengen enthält, kann dies u.U. zu einer Unterbrechung der Abarbeitung von mehreren Sekunden führen.

Beispiele

Tag-Name	Ausdruck
Tag	DirLength("C:\Studio") // Returns the size (in bytes) of all files and sub-folders from C:\Studio.

19.15.6 DirRename

Benennt ein Verzeichnis um.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DirRename	Datei	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
DirRename(strPath,strDirectoryFrom,strDirectoryTo)
```

Eingaben

strPath

Pfad des Verzeichnisses, welches umbenannt werden soll



Diese Funktion unterstützt Platzhalter ("*" und "?").

strDirectoryFrom

Der ursprüngliche Name des Verzeichnisses.

strDirectoryTo

Der neue Name des Verzeichnisses.

Rückgabewert

Wert	Beschreibung
−1	Ungültige Parameter
0	Verzeichnis konnte nicht umbenannt werden (z.B. **strDirectoryFrom** ist nicht vorhanden).
1	Verzeichnis wurde erfolgreich umbenannt.

Beispiele

Tag-Name	Ausdruck
Tag	DirRename("C:\Studio\","Temp", "New") // C:\Studio\Temp is renamed to C:\Studio\New.

19.15.7 FileCopy

Kopiert Dateien in ein übergebenes Verzeichnis

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
FileCopy	Datei	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
FileCopy(strSourceFile,strTargetFile{ | ,optNumTimeOut})
```

Anhang: WinStudio-interne Skriptsprache

Eingaben

strSourceFile

Pfad und Name der zu kopierenden Datei(en).



Diese Funktion unterstützt Platzhalter ("*" und "?").

strTargetFile

Der Dateipfad, in den die Dateien kopiert werden sollen.

optNumTimeOut

Numerischer Tag mit einer Ganzzahl zur Festlegung der Timeoutzeit für den Vorgang.

Bei Verwendung dieses Parameters gibt die Funktion nach der festgelegten Timeoutzeit den Wert –1 zurück.

Der Kopiervorgang wird dadurch nicht abgebrochen. Stattdessen wird ein interner Prozess zum Fertigstellen des Kopiervorgangs erstellt.

Rückgabewert

Wert	Beschreibung
−1	Timeout Zeit überschritten.
0	Datei(en) konnte(n) nicht kopiert werden.
1	Datei(en) erfolgreich kopiert.



Diese Funktion wird synchron ausgeführt, d.h. das Projekt hält an, während es auf die Fertigstellung der Funktion wartet. Große Datenmengen und langsame Netzwerkverbindungen können u.U. zu einer längeren Programmunterbrechung führen.

Beispiele

Tag-Name	Ausdruck
Tag	FileCopy("C:\Studio\Project\HST*.hst", "C:\Temp\Hst\", 1000)
Tag	FileCopy("C:\Studio\Project opert.txt", "C:\Temp\Tuesday_Report.txt", 500)

19.15.8 FileDelete

Löscht die angegebene Datei.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
FileDelete	Datei	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
FileDelete(strFile)
```

Eingaben

strFile

Pfad und Name der zu löschenden Datei.

Rückgabewert

Wert	Beschreibung
0	Datei konnte nicht gelöscht werden
Real	Gibt die Größe der gelöschten Datei zurück

Beispiele

Tag-Name	Ausdruck
Tag	FileDelete("C:\Studio\Readme.txt")

19.15.9 FileLength

Ermittelt die Größe einer Datei.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
FileLength	Datei	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
FileLength(strFile)
```

Eingaben

strFile

Pfad und Name der Datei.

Rückgabewert Gibt die Größe der angegebenen Datei in Bytes zurück.

Beispiele

Tag-Name	Ausdruck
Tag	FileLength("C:\Readme.txt")

19.15.10 FileRename

Benennt eine Datei um.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
FileRename	Datei	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
FileRename(strOldName,strNewName)
```

Eingaben

strOldName

Pfad und Name der Datei, die umbenannt werden soll .

strNewName

Pfad und neuer Name der Datei.

Rückgabewert Diese Funktion gibt keinen Wert zurück.

Beispiele

FileRename - Beispiele

```
FileRename( "C:\readme.txt", "C:\readthis.txt" )
```

19.15.11 FileWrite

Schreibt eine Zeichenkette in eine bestimmte ASCII- oder Unicode-Datei. Wenn die Datei nicht vorhanden ist, wird sie von der Funktion neu angelegt.

Anhang: WinStudio-interne Skriptsprache

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
FileWrite	Datei	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
FileWrite(strFileName,strWriteText{ | ,optNumAppend })
```

Eingaben

strFileName

Pfad und Namen der Datei, welche mit der Funktion beschrieben werden soll (String-Variable oder Zeichenkette in Anführungszeichen.

Standardmäßig wird die Datei in Ihren Projektordner geschrieben (d.h. der Ordner mit der Datei **projekt_name.APP**). Zum Schreiben in einen anderen Ordner geben Sie den vollständigen Dateipfad ein.

strWriteText

Eine Zeichenkette in Anführungszeichen oder ein String-Tag, welcher den Text enthält, der in die Datei geschrieben werden soll.

optNumAppend

Ein numerischer Wert oder der Name eines Integer-Tags mit einer Kennung für das Dateihandling (siehe Tabelle):

Wert	Beschreibung
0	Erstellt eine neue ASCII-Datei mit dem angegebenen Dateinamen. Ist die Datei bereits vorhanden, so wird sie überschrieben.
1	Anhängen an eine vorhandene ASCII-Datei mit dem gegebenen Dateinamen. Wenn die Datei nicht vorhanden ist, wird eine neue Datei erstellt.
2	Erstellt eine neue Unicode-Datei (UTF-16LE) mit dem angegebenen Dateinamen. Ist die Datei bereits vorhanden, so wird sie überschrieben.
3	Anhängen an eine vorhandene Unicode-Datei mit dem gegebenen Dateinamen. Wenn die Datei nicht vorhanden ist, wird eine neue Datei erstellt.

Dieser Parameter ist optional. Wenn ein anderer Wert als die Werte 0 bis 3 angegeben wird oder wenn der Parameter nicht verwendet wird, wird der Standardwert 0 verwendet.

Rückgabewert

0	Erfolgreiche Ausführung.
−1	Ungültige(r) Parameter.
−2	Datei konnte nicht geöffnet werden. Festplatte ist schreibgeschützt oder Dateiname ist ungültig.

Beispiele

Tag-Name	Ausdruck
Tag	FileWrite("c: est.txt", "This is a test")
Tag	FileWrite(strFileName, strWriteText)
Tag	FileWrite(strFileName, strWriteText, 1)

19.15.12 FindFile

Skript-Funktion zum Suchen nach Dateien, die einem bestimmten Suchbegriff entsprechen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
FindFile	Datei	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
FindFile(strFile{ | ,"optTagFilesFound"{ | ,optNumTimeout } })
```

Eingaben

strFile

Pfad und Name(n) der zu suchenden Datei(en).

Zum Auffinden von mehreren Dateien können Platzhalter ("*") verwendet werden. Zum Beispiel ***.gif** zum Suchen aller GIF-Dateien oder **log*.txt** zum Suchen aller Protokolldateien in einer Sequenz (z.B. **log001.txt**, **log002.txt**, **log003.txt**).

Standardmäßig sucht die Funktion nur im Projektordner. Es kann aber ein Dateipfad (relativ oder absolut) zum Suchen in einem beliebigen Ordner angegeben werden. Zum Beispiel: Wenn **strFile** als

\\laufwerkname oder IP-adresse\Logs\log*.txt

...definiert ist, durchsucht die Funktion das Verzeichnis **Logs** auf dem angegebenen Netzlaufwerk.

optTagFilesFound

Ein Tag-Array (vom Typ String), das die Namen der entsprechenden Dateien aufnimmt. Dieser Array-Name muss in Anführungszeichen gesetzt werden; anderenfalls versucht die Funktion, den Inhalt des Arrays abzurufen.

Dieser Parameter ist optional. Wird kein Wert angegeben, werden die Dateinamen nicht gespeichert und die Funktion gibt nur die Anzahl der gefundenen Dateien aus. Weitere Informationen siehe „Rückgabewert“.

☞ Das Array erhält nur die Dateinamen und "nicht" deren Pfade, auch wenn "strFile" zum Suchen außerhalb des Standardverzeichnisses konfiguriert wird.

optNumTimeout

Die Timeoutzeit (in Millisekunden) nach deren Ablauf die Such-Funktion abgebrochen wird.

Dieser Parameter ist optional. Wird kein Wert angegeben, sucht die Funktion weiter bis das gesamte angegebene Verzeichnis durchsucht ist.

Rückgabewert

Diese Funktion gibt einen der folgenden Werte zurück:

Wert	Beschreibung
-1	Zeitüberschreitung der Funktion.
0	Keine passenden Dateien gefunden.
n	Anzahl der gefundenen Dateien.

☞ Diese Funktion kann zwar von jedem Projekt-Client aufgerufen werden, sie wird aber immer auf dem Projekt-Server ausgeführt. Standardmäßig wird nur der Projektordner des Servers durchsucht und wenn "strFile" einen relativen Dateipfad enthält, muss dieser relativ zum Projektordner angegeben werden.

Da die Funktion synchron auf dem Projektserver ausgeführt wird, kann eine fehlerhafte Parametrierung zu einem Stillstand des Servers und aller angeschlossenen Clients führen, insbesondere dann, wenn der Parameter "optNumTimeout" nicht verwendet wird.

Bitte beachten Sie, dass der Parameter "optNumTimeout" auf Windows CE-Geräten nicht unterstützt wird.

Beispiele

Suchen aller Textdateien im Projektordner des Servers:

FindFile("*.txt")

Suchen aller Microsoft-Word-Dateien im Projektordner und anschließendes Senden der Dateinamen an **StringArray**, mit einer Timeoutzeit von 1000 Millisekunden:

FindFile("*.doc", "StringArray", 1000)

19.15.13 FindPath

Überprüft, ob ein Verzeichnis vorhanden ist.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
FindPath	Datei	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
FindPath(strPathName)
```

Eingaben

strPathName

Der Dateipfad, nach dem gesucht wird.

Rückgabewert

Wert	Beschreibung
0	Pfad nicht gefunden
1	Pfad gefunden

Beispiele

Tag-Name	Ausdruck
Tag	FindPath("C:\Windows")

19.15.14 GetFileAttributes

Liest die Attribute einer bestimmten Datei.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetFileAttributes	Datei	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetFileAttributes(strFile)
```

Eingaben

strFile

Pfad und der Name der Datei, deren Attribute ermittelt werden sollen.

Rückgabewert

Wert	Beschreibung
−1	Fehler
1	Schreibgeschützt
2	Versteckt
4	System
16	Verzeichnis
32	Archivieren
128	normal
256	Vorübergehend

Beispiele

Tag-Name	Ausdruck
Tag	GetFileAttributes("C:\Readme.txt")

19.15.15 GetFileTime

Liest Uhrzeit und Datum der letzten Dateiänderung aus.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetFileTime	Datei	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetFileTime(strFileName,numFormat)
```

Eingaben

strFileName

Pfad und Name und der Datei.

numFormat

Parameter, der das Format der Abfrage festlegt:

- 0: Gibt Datum und Uhrzeit der letzten Dateiänderung zurück.
- 1: Gibt nur das Datum der letzten Dateiänderung zurück.
- 2: Gibt nur die Uhrzeit der letzten Dateiänderung zurück.

Rückgabewert

Gibt Uhrzeit und/oder Datum der letzten Dateiänderung zurück.

Beispiele

Tag-Name	Ausdruck
Tag	GetFileTime("C:\Readme.txt")

19.15.16 GetHstInfo

Gibt Startzeit, Endzeit und Dauer der angegeben Historiendatei (*.HST) zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetHstInfo	Datei	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetHstInfo(strFileName,numInfoType)
```

Eingaben

strFileName

Pfad und Name der auszulesenden Historiendatei.

numFormat

Parameter, der die Art der zurückgegebenen Informationen festlegt:

- 0: Gibt die Startzeit der Dateiaufzeichnung zurück.
- 1: Gibt die Endzeit der Dateiaufzeichnung zurück.
- 2: Gibt die Dauer der Dateiaufzeichnung (in Stunden) zurück.

Rückgabewert

Gibt die Startzeit, Endzeit oder die Aufzeichnungsdauer in der *.HST-Datei zurück. Wenn die Datei nicht gelesen werden kann oder die angegebenen Informationen nicht ausgegeben werden können, wird ein Fehlercode zurückgegeben:

Wert	Beschreibung
-1	Startzeit konnte nicht abgefragt werden; überprüfen Sie, ob die Historiendatei vorhanden und gültig ist.
-2	Endzeit konnte nicht abgefragt werden; überprüfen Sie, ob die Historiendatei vorhanden und gültig ist.
-3	Interner Programmfehler; bitte wenden Sie sich an den technischen Support.
-4	Der Studio-TCP/IP-Server hat eine Uhrzeit ausgegeben, die nicht mit dem in der Projektmaske oder Webseite festgelegten Format kompatibel ist. Bitte verwenden Sie das Werkzeug Projekt überprüfen, Seite 64 zum Aktualisieren des Projekts und versuchen Sie es erneut.

Beispiele

Tag-Name	Ausdruck
Tag	GetHstInfo("batch", 0)
Tag	GetHstInfo("hst/02060801.hst", 1)
Tag	GetHstInfo("C:\batch.bat", 2)

19.15.17 GetLine

Ruft eine bestimmte Zeile oder einen Suchbegriff aus einer Textdatei ab und speichert dann die Zeile in einem String-Tag.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetLine	Datei	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetLine(strFileName, Search, "tagStore", optNumCase, "optOveflowTag", optRunFromServer)
```

Eingaben

strFileName

Pfad und Name der zu durchsuchenden Datei. Der Parameter kann ein vollständig qualifizierter Pfad (z.B. **C:\Datei.txt**) oder ein einfacher Dateiname (z.B. **Datei.txt**) sein. Im letzteren Fall sucht das Projekt in den folgenden Verzeichnissen nach der Datei:

- **Lokale Station**: Das Projekt sucht im Projektordner und im Unterordner **Web** nach der Datei.
- **Thin Client**: Ist der Parameter **optRunFromServer** auf 0 eingestellt, ist der Pfad, in dem die Datei gesucht wird, unbestimmt. Wenn er auf 1 eingestellt ist, wird in der im Browser eingegebenen URL nach der Datei gesucht. Wird diese dort nicht gefunden, wird in der Backup-URL gesucht.

☞ Für die Ausführung der Funktion auf einem Web Client empfehlen wir, den Parameter **optRunFromServer** auf 1 einzustellen und Ihre Dateien in den Unterordner **Web** des Projekts zu legen.

Search

Dieser Parameter dient der Adressierung der zu kopierenden Zeile. Es gibt zwei Möglichkeiten der Zeilenadressierung abhängig vom Datentyp des Parameters:

- Wenn es ein Wert oder Tag vom Typ **String** ist, durchsucht die Funktion die Textdatei nach dem ersten Auftreten der Zeichenkette und kopiert dann die gesamte Zeile mit der Zeichenkette in den von **tagStore** angegebenen Tag. Zusätzliche Vorkommen werden gezählt (siehe Rückgabewert unten), aber nicht kopiert.
- Wenn es ein **numerischer** Wert oder Tag ist, geht die Funktion zu dieser Zeilennummer in der Textdatei und kopiert dann die Zeile in den von **tagStore** angegebenen Tag. Die erste Zeile der Datei ist Zeile 0.

tagStore

Name des String-Tags, der den Inhalt der Zeile erhält, auf die die Funktion zeigt. Dieser Name muss in Anführungszeichen stehen. Andernfalls verwendet die Funktion den Inhalt des Tags und nicht den Namen.

optNumCase

"Optionaler" numerischer Tag zur Festlegung, ob bei der Suche Groß- und Kleinschreibung beachtet werden muss.

- 0: Groß-/Kleinschreibung nicht beachten
- 1: Groß-/Kleinschreibung beachten

optOverflowTag

"Optionaler" numerischer Tag, der das Ergebnis der Überlaufprüfung erhält.

- 0: OK
- 1: Überlauf

optRunFromServer

"Optionaler" numerischer Tag, der nur ausgewertet wird, wenn die Funktion auf einem Thin Client ausgeführt wird. Dabei hat der Parameter folgende Bedeutung:

- 0: Ruft die Datei aus dem Thin-Client-Gerät ab (diesen Wert nicht mit nicht vollständig qualifizierten Namen verwenden)
- 1: Ruft die Datei vom Webserver ab. Wenn der Dateiname keine URL ist, ignoriert die Funktion den Projektpfad und sucht nach der Datei in der URL, in der sich die Bilddateien befinden.

Rückgabewert

Wenn die Funktion erfolgreich ausgeführt wurde, gibt sie die Gesamtzahl der Zeilen aus, in denen der Suchbegriff gefunden wurde. Anderenfalls gibt die Funktion einen der folgenden Fehler aus:

Wert	Beschreibung
0	Suchbegriff wurde in der Zieldatei nicht gefunden
-1	Datei nicht gefunden
-2	Ungültiger Parameter "strFileName"
-3	Ungültiger Parameter "search"
-4	Ungültiger Parameter "tagStore"
-5	Ungültiger Parameter "optNumCase"
-6	Ungültiger Parameter "optOverflowTag"
-7	Ungültige Parameteranzahl
-8	Ungültige Zeilennummer

Hinweise

☞ Diese Funktion unterstützt nur ASCII- und UTF-16LE-Textverschlüsselung. (UTF-16LE ist die Unicode-Implementierung, die nativ von Windows unterstützt wird.) Wenn diese Funktion zum Abrufen von Text aus einer mit UTF-8 oder UTF-16-BE formatierten Datei verwendet wird, können ungültige Zeichen auftreten.

☞ Diese Funktion kann nur bis zu 509 Zeichen in einem Funktionsaufruf lesen. Wenn eine Zeile mehr als 509 Zeichen beinhaltet (d.h. alphanumerisch + CR + LF),interpretiert die Funktion die Zeichenkette so, als würde sie in mehreren aufeinander folgende Zeilen mit der maximalen Länge von 509 Zeichen stehen. Der Text aus Zeile 100 mit 1024 Zeichen (d.h. 1022 alphanumerische + CR + LF) verteilt sich dann "virtuell" auf die Zeilen 100, 101 und 102. Die Funktion muss deshalb dreimal aufgerufen werden.

Dies hat Auswirkungen auf alle nachfolgenden Zeilenzugriffe der Funktion. Die Zeile 101 der Quelldatei hat dann die virtuelle Zeilennummer 103 usw. . Daher sollte die Quelldatei auf 509 Zeichen pro Zeile beschränkt werden, um unnötig kompliziertes Zählen der Zeilen zu vermeiden.

... zum Auslesen der Zeile 100 muss die Funktion dreimal aufgerufen werden:

```
GetLine( "C:\FileName.txt", 100, "strTagStore[1]")  //Reads the first 509 characters
GetLine( "C:\FileName.txt", 101, "strTagStore[2]")  //Reads the second 509 characters
GetLine( "C:\FileName.txt", 102, "strTagStore[3]")  //Reads the last 6 characters
```

Beispiele

Tag-Name	Ausdruck
Tag	GetLine("C:\TechRef v61.doc", "Studio version 6.1", "ReturnedLine")// Gets the first occurance of "Studio version 6.1".
Tag	GetLine("C:\Readme.txt", 1, "ReturnedLine", 0, "Overflow")// Gets the second line of the file.

19.15.18 HST2TXT

Exportiert Informationen aus einer binär aufgezeichneten Trenddatei (*.hst) in eine einfache Text- (***.txt**) oder kommagetrennte (***csv**) Datei.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
HST2TXT	Datei	asynchron	unterstützt	unterstützt	unterstützt

Syntax

```
HST2TXT( strStartDate, strStartTime, numDuration, numGroup, strTargetFile, optStrSeparator,
         optNumMilliseconds, optStrFormat, optNumInterval )
```

Eingaben

strStartDate

Das Startdatum der auzulesenden Daten.

strStartTime

Die Startzeit der auzulesendenDaten.

numDuration

Numerischer Tag mit der Dauer der Daten in Stunden.

numGroup

Numerischer Tag mit der Trendgruppennummer.

strTargetFile

Pfad und Name der zu schreibenden Datei.

optStrSeparator

"Optionales" Datentrennzeichen für die Datei. Wird der Parameter weggelassen, verwendet die Funktion das TAB-Zeichen (\t) zum Trennen der Werte in der Textdatei.

optNumMilliseconds

"Optionaler" numerischer Tag. Ist dieser Parameter falsch ("0"), zeigt die erstellte Datei auf dem Zeitstempel für jede Historienstichprobe keine Millisekundengenauigkeit.

optStrFormat

String Tag oder Zeichenkette in Anführungszeichen, welche das Datumsformat der Einträge in der Zieldatei festlegt:

- "DMY": Tag, Monat, Jahr
- "MDY": Monat, Tag, Jahr
- "YMD": Jahr, Monat, Tag

Wird dieser optionale Parameter nicht eingetragen, verwendet die Funktion DMY für den Zeitstempel in der Textdatei.

optNumInterval

"Optionaler" numerischer Parameter, der das Stichprobenintervall angibt. Nur Zeileneinträge aus diesem Intervall werden als Text in die Zieldatei geschrieben; alle anderen Zeilenelemente in der Trendhistorie werden verworfen.

Wenn zum Beispiel **optNumInterval** den Wert 10 hat, wird nur jede zehnte Zeile geschrieben.

Rückgabewert

Wert	Beschreibung
−3	Ungültige Anzahl Parameter
−2	Dll-Funktionen nicht gefunden
−1	"InStudiot.dll" nicht gefunden
0	Funktion wurde erfolgreich ausgeführt
1	Fehler. Vorherige Ausführung von **HST2TXT** wurde noch nicht fertiggestellt

Beispiele

Tag-Name	Ausdruck
Tag	HST2TXT("04/14/2002", "06:30:00", 0.1, 3, "C:\Studio\data.txt", "\")
Tag	HST2TXT("04/14/2002", "06:30:00", 0.1, 3, "C:\Studio\data.csv", "," , "MDY")
Tag	HST2TXT("04/14/2002", "06:30:00", 0.1, 3, "C:\Studio\data.csv", "," , "MDY", 10)



Bei Verwendung des Kommazeichens (,) als **optStrSeparator** erzeugt die Funktion eine Datei im CSV-Format (Comma Separated Values). Diese ist hilfreich für das Erzeugen von Trendhistoriendateien, die mit Microsoft Excel geöffnet werden können

19.15.19 HST2TXTIsRunning

Gibt den Status der Funktion "HST2TXT" zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
HST2TXTIsRunning	Datei	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
HST2TXTIsRunning()
```

Diese Funktion hat keine Parameter.

Rückgabewert

Wert	Beschreibung
0	HST2TXT wird noch ausgeführt.
−1	Der letzte Konvertierungsvorgang wurde korrekt ausgeführt.
−2	Reserviert.
−3	Datei nicht gefunden. Für die angegebene Gruppe sind keine Historiendateien im konfigurierten Zeitintervall vorhanden.
-4	**HST**-Datei kann nicht geöffnet werden.
-5	ASCII-Datei kann nicht erstellt/geöffnet werden.
-6	Dateiinformationen können nicht aus der **HST**-Datei ausgelesen werden.
-7	Ungültiger Dateityp.
-8	Titelbereichinformationen können nicht aus der **HST**-Datei ausgelesen werden.
-9	Ungültige Anzahl von Tags in Titelbereichsinformation ("0 > nTags > 250").
-10	Kann Titelbereichsdatei (**.hdr**) nicht erstellen.
-20	InStudiot.dll nicht gefunden.
-30	Kein Zugriff auf dll-Funktion.

Beispiele

HST2TXTIsRunning - Beispiele

```
HST2TXTIsRunning()
```

19.15.20 LookupContains

Diese Funktion überprüft, ob eine externe Datei das angegebene Schlüsselwort in der Schlüsselspalte enthält.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
LookupContains	Datei	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
LookupContains(strKey)
```

Eingaben

strKey

Das Schlüsselwort, nach dem in der Schlüsselspalte der Datei gesucht wird.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte aus:

Wert	Beschreibung
0	Angegebenes Schlüsselwort nicht gefunden.
1	Angegebenes Schlüsselwort gefunden.



Die externe Datei muss durch Aufruf der Funktion LookupLoad, Seite 670 bereits geladen sein.

Beispiele

LookupContains - Beispiele

```
LookupContains( "customer167" )
```

19.15.21 LookupGet

Diese Funktion ruft einen Wert aus einer Tabelle in einer externen Datei ab. Dabei wird der Wert in der Wertespalte mit Hilfe einer Schlüsselspalte referenziert.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
LookupGet	Datei	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
LookupGet(strKey)
```

Eingaben

strKey

Das Schlüsselwort, nach dem in der Schlüsselspalte der Datei gesucht wird.

Rückgabewert

Diese Funktion gibt den über das Schlüsselwort referenzierten Wert (als String) aus.

Wird kein Wert gefunden, gibt diese Funktion **strKey** zurück.



Die externe Datei muss durch Aufruf der Funktion LookupLoad, Seite 670 bereits geladen sein.

Beispiele

LookupGet - Beispiele

```
LookupGet( "customer167" )
```

19.15.22 LookupLoad

Diese Funktion lädt eine externe Textdatei, die zum Nachschlagen von Tabellenwerten verwendet werden kann. Eine Spalte der Datei ist als Schlüsselspalte und eine andere Spalte ist als Wertespalte ausgewiesen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
LookupLoad	Datei	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
LookupLoad(strFileName,numColKey,numColValue,strDelimiters)
```

Eingaben

strFileName

Pfad und Name der externen Datei.

numColKey

Die Nummer der Spalte/des Felds mit den Schlüsselwörtern.

numColValue

Die Nummer der Spalte/des Felds mit den gewünschten Werten.

strDelimiters

Das Begrenzungszeichen, das die Spalten/Felder trennt.

Rückgabewert

Diese Funktion gibt die Anzahl der Zeilen in der angegebenen Datei zurück.

Wenn die angegebene Datei nicht gefunden werden kann, gibt diese Funktion eine negative Zahl als Fehlercode zurück.



Diese Funktion lädt nur die angegebene Datei; die Datei selbst wird nicht manipuliert. Zum Auslesen der Datei stehen die Funktionen LookupContains, Seite 669 und LookupGet, Seite 670 zur Verfügung.

Es kann immer nur eine Datei geladen werden. Bei erneutem Aufruf der Funktion mit einer neuen Datei wird die alte Datei aus dem Projektspeicher gelöscht.

Beispiele

LookupLoad - Beispiele

```
LookupLoad( "C:\Temp\customerlist.csv", 1, 4, "," )
```

19.15.23 PDFCreate

Erstellt eine PDF-Datei aus der angegebenen Quelldatei.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
PDFCreate	Datei	synchron	unterstützt	nicht unterstützt	unterstützt

Syntax

```
PDFCreate(strSourceFile{ | ,optStrPdfFile })
```

Eingaben

strSourceFile

String zur Angabe von Dateipfad und -name der gewünschten Quelldatei (***.doc**, ***.txt**, oder ***.rtf**). Wenn kein vollständiger Pfad angegeben wird, sucht die Funktion im Projektordner nach der Quelldatei.

optStrPdfFile

"Optionaler" String-Parameter zur Festlegung von Dateipfad und -name der erstellten PDF-Datei. Wird dieser Parameter nicht verwendet – d.h. wenn kein Dateipfad oder -name angegeben ist – wird die PDF-Datei am gleichen Speicherort wie die Quelldatei und mit dem gleichen Namen gespeichert. Es wird nur eine neue Erweiterung hinzugefügt. Zum Beispiel:

aus **\pfad\MeinDokument.rtf** wird **\pfad\MeinDokument.pdf**.



Wenn der Dateiname ohne Pfad eingegeben wird, ist ein Backslash ("\") nicht erforderlich.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Fehler in PDF-Profilinformation
3	Fehler beim Speichern der PDF-Datei
4	Job abgebrochen
101	Fehler beim Initialisieren der PDF-Ressource
102	Angegebene Quelldatei nicht gefunden
103	Fehler beim Erzeugen der PDF-Datei

Anhang: WinStudio-interne Skriptsprache

Wert	Beschreibung
104	Falsche Anzahl Parameter
105	Falscher Parametertyp

☞ Diese Funktion unterstützt nur die Ausführung von jeweils einem Job. Wenn mehr als ein Benutzer oder Befehl gleichzeitig versuchen, die Funktion aufzurufen, schlägt die Funktion fehl und gibt den Wert "101" zurück.

Beispiele

Tag-Name	Ausdruck
	PDFCreate("C:\Report1.rtf")
	PDFCreate("C:\Report2.doc", "C:\Converted1.pdf")
	PDFCreate("C:\Report3.txt", "C:\Data\Converted1.pdf")

19.15.24 Print

Druckt eine Textdatei.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Drucken	Datei	asynchron	unterstützt	unterstützt	unterstützt

Syntax

```
Print(strFilePath{ | ,optNumOrientation })
```

Eingaben

strFilePath

Pfad und Name der zu druckenden Textdatei.

optNumOrientation

Parameter zur Einstellung der Seitenausrichtung:

- 0 (Standard) = Hochformat
- 1 = Querformat

Rückgabewert Gibt keine Werte zurück.

☞ Der Parameter **optNumOrientation** wird nicht unterstützt, wenn das Projekt auf einem Windows-CE-Betriebssystem ausgeführt wird.

Beispiele

Tag-Name	Ausdruck
	Print("C:\ReadMe.txt")
	Print("C:\ReadMe.txt", 1)
	Print(TagFileName, 0)

☞ Diese Funktion kann nur zum Drucken von Textdateien verwendet werden. Daten in anderen Formaten (z.B. Bilder, Binärdateien usw.) können mit dieser Funktion nicht gedruckt werden.

19.15.25 RDFileN

Öffnet ein "Datei-Browser"-Fenster zum Auswählen einer Datei.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
RDFileN	Datei	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
RDFileN("tagSelectedFile", strSearchPath, strMask, optNumChangeDir)
```

Eingaben

tagSelectedFile

Parameter in Form eines String-Tags, in welchem der Pfad und Name der Datei eingetragen wird, die im Dateibrowser ausgewählt wurde. Der Tag-Name **muss** in Anführungszeichen gesetzt werden, andernfalls versucht das Projekt, den Inhalt des Tags abzurufen. Es darf kein VBScript-Variablenname sein.

strSearchPath

Der Dateipfad (Verzeichnis), welcher im Browser geöffnet wird.

strMask

Die Maske zum Filtern der Dateien, die im Browser angezeigt werden.

optNumChangeDir

"Optionaler" numerischer Tag zur Festlegung, ob der Bediender das Suchverzeichnis ändern darf. Wird dieser Parameter weggelassen oder auf "TRUE" (1) gesetzt, kann der Bediener in dem geöffneten Fenster zu anderen Verzeichnissen wechseln. Wird er auf "FALSE" (0) gesetzt, ist das Fenster auf das in **strSearchPath** festgelegte Verzeichnis beschränkt.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Einer der Parameter ist keine Zeichenkette
2	Parameter 1 enthält einen ungültigen Tag-Namen
3	Vorgang durch den Benutzer abgebrochen

Beispiele

Tag-Name	Ausdruck
Tag	RDFileN("FileName", "C:\Studio\", "*.doc", 1)

19.15.26 WebGetFile

Lädt eine Datei von einer bestimmten Web-Adresse herunter und speichert sie lokal.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
WebGetFile	Datei	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
WebGetFile(strURL,strLocalPath)
```

Eingaben

strURL

Die URL (d.h. die Internetadresse) der herunterzuladenden Datei (String-Variable oder Zeichenkette in Anführungszeichen).

strLocalPath

Der vollständige Dateipfad, wo die Datei gespeichert werden soll (String-Variable oder Zeichenkette in Anführungszeichen).

Rückgabewert

Wert	Beschreibung
−1	Timeout
0	Datei nicht gefunden
1	Datei erfolgreich gespeichert

Beispiele

Tag-Name	Ausdruck
Tag	WebGetFile("http://www.the-internet.com/myfile.txt", "C:\myfile.txt")
Tag	WebGetFile(myURL, myFilePath)

19.16 Grafische Funktionen

19.16.1 Grafische Funktionen – Allgemeines

Diese Funktionen dienen dem Handhaben und Drucken von WinStudio-Bildern.

grafisch

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
AutoFormat, Seite 674	synchron	unterstützt	unterstützt	unterstützt
GetScrInfo, Seite 675	synchron	unterstützt	unterstützt	unterstützt
PrintSetup, Seite 676	asynchron	unterstützt	unterstützt	unterstützt
PrintWindow, Seite 677	asynchron	unterstützt	unterstützt	nicht unterstützt
ResetDecimalPointsTable, Seite 678	synchron	unterstützt	unterstützt	unterstützt
RGBColor, Seite 679	synchron	unterstützt	unterstützt	unterstützt
RGBComponent, Seite 679	synchron	unterstützt	unterstützt	unterstützt
SetDecimalPoints, Seite 681	synchron	unterstützt	unterstützt	unterstützt
SetDisplayUnit, Seite 682	synchron	unterstützt	unterstützt	unterstützt
SetTagDisplayUnit, Seite 683	synchron	unterstützt	unterstützt	unterstützt

19.16.2 AutoFormat

Formatiert eine reelle Zahl mit einer vorgegebenen Anzahl Dezimalstellen, gemäß der von der Funktion "SetDecimalPoints" erzeugten virtuellen Formatierungstabelle. (Dies ist ähnlich der Funktion "Format", mit dem Unterschied dass die Anzahl der Dezimalstellen nicht angegeben werden muss.)

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
AutoFormat	grafisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
AutoFormat(numValue)
```

Eingaben

numValue

Die zu formatierende reelle Zahl.

Rückgabewert Diese Funktion gibt eine formatierte Zeichenkette zurück.

Beispiele In den folgenden Beispielen wurde vor dem Aufruf von "AutoFormat" die Funktion SetDecimalPoints, Seite 681 verwendet, um 3 Dezimalstellen für Werte größer oder gleich 1,5 zu setzen und 1 Dezimalstelle für Werte kleiner oder gleich –3.

Tag-Name	Ausdruck
Tag	AutoFormat(1.543210) // Returned value = "1.543"
Tag	AutoFormat(−3.123456) // Returned value = "−3.1"

19.16.3 GetScrInfo

Ruft aus dem Projekt Informationen über ein geöffnetes WinStudio-Bild ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetScrInfo	grafisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetScrInfo(strScreenName, "tagResult", optNumResultType, optNumID)
```

Eingaben

strScreenName

Der Name des Bildes, dessen Informationen abgerufen werden sollen.

tagResult

Der Name des Tags, der die von der Funktion zurückgegebenen Daten erhält. Dieser Name muss in Anführungszeichen gesetzt werden, andernfalls versucht das Projekt, den Inhalt des Tags abzurufen. Für die Rückgabe von mehreren Werten (optNumResultType =0) muss ein Array-Tag angegeben werden.

optNumResultType

Parameter zur Konfiguration des Rückgabewertes

Wert	Beschreibung
0	Standardwert. Schreibt die Bildnkoordinaten für OBEN, LINKS, UNTEN und RECHTS in die aufeinanderfolgenden Positionen des durch den Parameter tagResult festgelegten Array-Tags, Seite 125.
1	Schreibt die OBERE Bildkoordinate in die durch den Parameter tagResult festgelegten Tag.

Wert	Beschreibung
2	Schreibt die LINKE Bildkoordinate in die durch den Parameter tagResult festgelegten Tag.
3	Schreibt die UNTERE Bildkoordinate in die durch den Parameter tagResult festgelegten Tag.
4	Schreibt die RECHTE Bildkoordinate in die durch den Parameter tagResult festgelegten Tag.

Dieser Parameter ist optional; der Standardwert ist 0.

optNumID

Die spezifische Instanznummer des Bildes. (Die ID wird zugewiesen, wenn das Bild mit der Funktion Open, Seite 604 geöffnet wird.) Dieser Parameter ist optional; die Standardkennung ist 0.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
-1	Der erste und/oder zweite Parameter sind keine Zeichenketten.
-2	Fehler bei Speicherzuweisung.
-3	optNumResultType ist null, aber tagResult gibt keinen Bereichs-Tag vor.
-4	Ungültiger Tag durch Parameter tagResult.

Beispiele

Tag-Name	Ausdruck
TagErrorCode	GetScrInfo("main" , "TagXY[0]") // Retrieves the TOP, LEFT, BOTTOM and RIGHT coordinates of the "main" screen and then writes them to the first four positions of the array tag TagXY.
TagErrorCode	GetScrInfo("main", "TagXY", 3) // Retrieves the BOTTOM coordinate of the "main" screen and then writes it to TagXY.
TagErrorCode	GetScrInfo("main" , "TagXY", 2, 10) // Retrieves the LEFT coordinate of the "main" screen with ID 10 and then writes it to TagXY.

19.16.4 PrintSetup

Öffnet den Standard-Konfigurationsdialog des Betriebssystems, aus dem heraus der Drucker ausgewählt und konfiguriert werden kann.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
PrintSetup	grafisch	asynchron	unterstützt	unterstützt	unterstützt



Diese Funktion kann nicht mit Tasks, Seite 61 oder im Skript Globale Prozeduren, Seite 799 verwendet werden.

Syntax

```
PrintSetup()
```

Diese Funktion hat keine Parameter.

Rückgabewert Gibt keine Werte zurück.

Beispiele *PrintSetup - Beispiele*

```
PrintSetup()
```

19.16.5 PrintWindow

Druckt einen Screenshot eines WinStudio-Bildes. Das Bild muss nicht geöffnet und aktiviert sein; die Funktion kann ein im Hintergrund ausgeführtes Bild und sogar geschlossene Bilddateien drucken.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
PrintWindow	grafisch	asynchron	unterstützt	unterstützt	unterstützt

☞ Diese Funktion kann nicht mit Tasks, Seite 61 oder im Skript Globale Prozeduren, Seite 799 verwendet werden.

Syntax

```
PrintWindow(strScreenName , optNumOrientation, optNumID, optStrMnemonicList)
```

Eingaben

strScreenName

Der Name des zu druckenden Bildes. Wird dieser Parameter weggelassen, wird das aktuell aktive Bild gedruckt. (Dieser Parameter darf nicht verwendet werden, wenn die Funktion auf einem Windows-CE-Zielsystem ausgeführt wird.)

optNumOrientation

Parameter zur Festlegung der Papierausrichtung:

Wert	Beschreibung
0	Hochformat
1	Querformat

Standardwert ist 0.

☞ Der Parameter **optNumOrientation** wird nicht unterstützt, wenn das Projekt auf einem Windows-CE-Zielsystem ausgeführt wird.

optNumID

Die spezifische Instanznummer des Bildes. (Die ID wird zugewiesen, wenn das Bild mit der Funktion Open, Seite 604 geöffnet wird.) Dieser Befehlsparameter ist optional; die Standardkennung ist 0.

optStrMnemonicList

Eine Zeichenkette, die beschreibt, wie die benutzerdefinierten Eigenschaften von generischen Objekten oder verknüpften Symbolen, Seite 220 im WinStudio-Bild beim Drucken des Bildes vervollständigt werden. Diese Zeichenkette hat die Syntax...

 # Label:Value

...wobei **Label** der Name der Eigenschaft ist und **Value** der Tag, Ausdruck oder direkte Wert, den die Eigenschaft erhält. Es können auch mehrere, durch leerzeichen getrennte Übergabewerte konfiguriert werden. Beispiele finden Sie nachstehend im Abschnitt Beispiele.

Der Parameter **optStrMnemonicList** funktioniert nicht bei bereits geöffneten Bildern; wenn das Bild geöffnet wurde, haben die benutzerdefinierten Eigenschaften bereits ihre Standardwerte erhalten.

Rückgabewert

Diese Funktion gibt keinen Wert zurück.

Beispiele

Tag-Name	Ausdruck
	PrintWindow( ) // Prints the currently active screen in portrait orientation.
	PrintWindow("Main", 1) // Prints the "Main" screen in landscape orientation.
	PrintWindow(TagScreenName) // Prints the screen specified by TagScreenName.
	PrintWindow("Main", 1, 10) // Prints the "Main" screen with ID 10.
	PrintWindow ("Main", 1, 0, "#Mne1:Tag1 #Mne2:Tag2") // Prints the "Main" screen, replacing the custom properties **Mne1** and **Mne2** with **Tag1** and **Tag2**, respectively.

Diese Funktion kann zum Drucken von grafischen Berichten verwendet werden, die die Objekte Alarm-/Ereignissteuerung, Seite 290 und Trendsteuerung, Seite 309 beinhalten.

19.16.6 ResetDecimalPointsTable

Setzt die von der Funktion "SetDecimalPoints" erzeugte virtuelle Einstellungstabelle zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ResetDecimalPointsTable	grafisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ResetDecimalPointsTable()
```

Diese Funktion hat keine Parameter.

Rückgabewert

Diese Funktion gibt keinen Wert zurück.

Beispiele

Tag-Name	Ausdruck
	ResetDecimalPointsTable( ) // Resets the virtual table of settings.

19.16.7 RGBColor

Gibt die Nummer der durch die RGB-Codes (rot, grün, blau) definierten Farbe zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
RGBColor	grafisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
RGBColor(numRed,numGreen,numBlue)
```

Eingaben

numRed

RGB Code für den roten Farbanteil

numGreen

RGB Code für den grünen Farbanteil

numBlue

RGB Code für den blauen Farbanteil

Rückgabewert

Diese Funktion gibt die Nummer der durch die RGB-Codes (rot, grün, blau) definierten Farbe zurück.

Beispiele

Tag-Name	Ausdruck
TagColor	RGBColor(51,153,102) // This function returns the value 13434828, which is the color code for Sea Green.
TagColor	RGBColor(TagRed,TagGreen,TagBlue) // This function returns the color code of the RGB values set in the tags TagRed, TagGreen and TagBlue, respectively.



Eine Liste der RGB-Codes und Farbwerte für die am häufigsten verwendeten Farben finden Sie im Abschnitt Farbschnittstelle, Seite 265.

19.16.8 RGBComponent

"RGBComponent" ist eine interne Skript-Funktion, die aus einer Farbnummer den Farbcode einer der drei Grundfarben (rot, grün oder blau) ermittelt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
RGBComponent	grafisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
RGBComponent(numColor,{ numComponent | 0 | 1 | 2 })
```

Eingaben

numColor

Der Dezimalcode für eine 24-bit RGB-Farbe, die ein Ganzzahlwert zwischen 0 und 16777215 sein kann. (Dieses Farbmodell ist auch als „Truecolor“ oder „16 Millionen von Farben“ bekannt.)

numComponent

Die Grundfarbe welche ermittelt werden soll: "0" ist rot, "1" ist grün und "2" ist blau.

Rückgabewert Diese Funktion gibt einen Ganzzahlwert zwischen 0 und 255 zurück, der den jeweiligen Farbcode der konfigurierten Grundfarbe darstellt



Eine Liste der häufig verwendeten RGB-Farbcodes und die entsprechenden „normalen“ Farbnamen finden Sie unter Farbschnittstelle, Seite 265.

Beispiele Abruf der roten Grundfarbe in Farbnummer 13434828 (d.h. seegrün):

RGBComponent(13434828, 0)

Abruf des RGB-Codes für die in **TagComponent** festgelegte Grundfarbe (rot, blau, grün) aus der Farbnummer welche in **TagCode** angegeben ist .

RGBComponent(TagCode, TagComponent)

19.16.9 SaveScreenShot

Diese Funktion macht einen Screenshot der WinStudio-Bildes und speichert diesen in einer Bilddatei.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SaveScreenShot	grafisch	synchron	unterstützt	unterstützt (nur für geöffnete Bilder)	auf dem Server ausgeführt

Syntax

```
SaveScreenShot({ | optStrScreenName{ | ,optStrOutputFile{ | ,{ optNumFormat | 0 | 1 | 2 | 3 | 4 | 5 } } } })

SaveScreenShot()
SaveScreenShot( optStrScreenName )
SaveScreenShot( optStrScreenName, optStrOutputFile )
SaveScreenShot( optStrScreenName, optStrOutputFile, optNumFormat )
```

Eingaben

optStrScreenName

Der Name einer Bilddatei (***.scr**). Wird kein Dateipfad angegeben, muss die Datei sich im Ordner "Screens" unterhalb des Projektordners befinden (z.B. **\projektname\Screens\bildname.scr**).

Wenn die Screenshot Funktion auf einem Windows PC ausgeführt wird, müssen die Bilder nicht zwingend geöffnet sein. Auf Windows CE-Geräten hingegen funktioniert das Drucken nur bei geöffneten Bildern.

Dies ist ein optionaler Parameter; wenn kein Wert (oder "") angegeben ist, wird das aktuell geöffnete und aktivierte Bild verwendet.

optStrOutputFile

Der Name der Ausgabedatei. Ist kein Dateipfad angegeben, wird die Bilddatei im Web-Ordner des Projekts gespeichert (z.B. **\projektname\web\bildname.jpg**)

Dies ist ein optionaler Parameter; ist kein Wert angegeben, wird der Bildname des selektierten Bildes (**optStrScreenName**) verwendet.

optNumFormat

Das Format der Bilddatei:

Wert	Beschreibung
0	BMP
1	JPG
2	PNG
3	GIF
4	TIFF
5	Auto

Dies ist ein optionaler Parameter; wird kein Wert angegeben, ist der Standardwert "1" (JPG).

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte aus:

Wert	Beschreibung
0	erfolgreich
-1	Falsche Anzahl Parameter.
-2	Falsche Parametertypen.
-3	Ungültiges Verzeichnis.
-4	Zweiter Parameter darf nicht leer sein.
-5	Falsches Format/ungültige Option für dritten Parameter.
-6	Datei konnte nicht gespeichert werden.
-7	Kompatible Bitmap konnte nicht erzeugt werden.
-10000	Projekt wird nicht ausgeführt. (Dies passiert üblicherweise, wenn versucht wird, die Funktion im Datenbankspion zu verwenden.) Siehe Hinweis.

Damit diese Funktion ordnungsgemäß ausgeführt wird, muss das Projekt laufen und der Viewer-Task muss gestartet sein.

Beispiele

Speichern des aktuell aktivierten Bildes in **\projektname\Web\ScreenShot.JPG**:

SaveScreenShot()

Speichern der Bilddatei **main.scr** in **\projektname\Web\main.JPG**:

SaveScreenShot("main.scr")

Speichern des aktuell aktivierten Bildes als BMP mit dem Namen des aktuell angemeldeten Benutzers:

SaveScreenShot("", UserName, 0)

19.16.10 SetDecimalPoints

Legt die Anzahl der anzuzeigenden Dezimalstellen für eine angegebene Reihe von reellen Zahlen fest. Diese Einstellung wird von allen Eingabe- und Dynamikobjekten verwendet, bei denen die Option **Auto Format** aktiviert ist sowie von der Funktion **AutoFormat** selbst.

Anhang: WinStudio-interne Skriptsprache

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetDecimalPoints	grafisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
SetDecimalPoints(numBaseValue,numDecimalPoints)
```

Eingaben

numBaseValue

Der Basiswert der Reihe von reellen Zahlen. Für negative Werte beinhaltet der Bereich alle reellen Zahlen, die kleiner oder gleich diesem Wert sind. Für positive Werte beinhaltet der Bereich alle reellen Zahlen, die größer oder gleich diesem Wert sind. (Für die Festlegung beider Wertebereiche muss die Funktion zweimal mit den entsprechenden Werten aufgerufen werden)

numDecimalPoints

Die Anzahl der anzuzeigenden Dezimalstellen für die von "numBaseValue" angegebene Reihe von reellen Zahlen.

Rückgabewert

Wert	Beschreibung
0	Fehler
1	Erfolgreich

☞ Wenn Sie diese Funktion mehrmals mit unterschiedlichen Parametern aufgerufen, können Sie eine virtuelle Tabelle mit Formatierungseinstellungen aufbauen. Sie können für jede Reihe von reellen Zahlen eine andere Anzahl von Dezimalstellen festlegen und alle Einstellungen werden für die Dauer der Laufzeit gespeichert oder bis die Tabelle mithilfe der Funktion ResetDecimalPointsTable, Seite 678 zurückgesetzt wird.

☞ Diese Formatierung ändert "nicht" den tatsächlichen Wert eines Tags oder Ausdrucks. Sie ändert nur, wie die Werte von Bildobjekten angezeigt werden.

Beispiele

Tag-Name	Ausdruck
Tag	SetDecimalPoints(1.5, 3) // Displays 3 decimal places for all real numbers greater than or equal to 1.5.
Tag	SetDecimalPoints(−3, 1) // Displays 1 decimal place for all real numbers less than or equal to −3.

19.16.11 SetDisplayUnit

Sucht alle Tags und alle Rasterobjekt- und Trendobjektwerte, die über eine bestimmte technische Einheit verfügen (siehe Tag-Feld **Einheit**) und setzt dann die Felder **DisplayUnit**, **UnitDiv** und **UnitAdd** für diese Tags.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetDisplayUnit	grafisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
SetDisplayUnit(strUnitOrigin,strDisplayUnit,numDiv,numAdd)
```

Eingaben

strUnitOrigin

Die ursprüngliche Einheit.

strDisplayUnit

Der neue Wert für das Tag-Feld **DisplayUnit**.

numDiv

Der neue Wert für das Tag-Feld **UnitDiv**.

numAdd

Der neue Wert für das Tag-Feld **UnitAdd**.

Rückgabewert

Wert	Beschreibung
0	erfolgreich
-1	Falsche Parameteranzahl.
-2	"strUnitOrigin" Parameter ist leer.
-3	"numDiv" Parameter ist ungültig (gleich 0).

☞ Diese Funktion beeinflusst nur die Tag-Anzeige; die tatsächlichen Tag-Werte werden in keiner Weise verändert.

Beispiele

Tag-Name	Ausdruck
Tag	SetDisplayUnit("C", "F", 0.555556, 32) // For all tags and object values with a **Unit** of "C", the **DisplayUnit** tag field is set to "F", the **UnitDiv** tag field is set to 0.555556, and the **UnitAdd** tag field is set to 32.

19.16.12 SetTagDisplayUnit

Setzt die Variableneigenschaften **DisplayUnit**, **UnitDiv** und **UnitAdd** für eine bestimmte Variable.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetTagDisplayUnit	grafisch	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
SetTagDisplayUnit(strTagName,strDisplayUnit,numDiv,numAdd)
```

Eingaben

strTagName

Name der Variablen, deren Eigenschaften **DisplayUnit**, **UnitDiv** und **UnitAdd** gesetzt werden sollen. Wird diesem Parameter eine String-Variable zugewiesen, sollte diese den Namen der Variablen "enthalten", deren Eigenschaften gesetzt werden sollen.

strDisplayUnit
Der neue Wert für die Tag-Eigenschaft **DisplayUnit**.

numDiv
Der neue Wert für die Tag-Eigenschaft **UnitDiv**.

numAdd
Der neue Wert für die Tag-Eigenschaft **UnitAdd**.

Rückgabewert

Wert	Beschreibung
0	erfolgreich
-1	Falsche Parameteranzahl.
-2	Angegebener Tag nicht vorhanden.
-3	numDiv Parameter ist ungültig (gleich 0).

Beispiele

Tag-Name	Ausdruck
Tag	SetTagDisplayUnit("TagTemp", "F", 0.555556, 32) // For the tag "TagTemp", the **DisplayUnit** tag field is set to "F", the **UnitDiv** tag field is set to 0.555556, and the **UnitAdd** tag field is set to 32.

19.17 Übersetzungsfunktionen

19.17.1 Übersetzungsfunktionen – Allgemeines

Diese Funktionen dienen zur Übersetzung von Texten während der Laufzeit.

Übersetzung

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
Ext, Seite 684	synchron	unterstützt	unterstützt	unterstützt
SetLanguage, Seite 685	synchron	unterstützt	unterstützt	unterstützt
SetTranslationFile, Seite 685	synchron	unterstützt	unterstützt	unterstützt

19.17.2 Ext

Übersetzt den angegebenen Text mithilfe der aktiven Übersetzungsdatei (siehe "SetLanguage" bzw. "SetTranslationFile").

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Ext	Übersetzung	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
Ext(strText)
```

Eingaben

strText
Der zu übersetzende Text.

Rückgabewert Gibt die Textübersetzung mithilfe der aktiven Übersetzungsdatei zurück.

Beispiele

Tag-Name	Ausdruck
Tag	Ext("Start") // Returned value if translating to Portuguese = "Iniciar"
Tag	Ext("Stop") // Returned value if translating to German = "Stopp"

19.17.3 SetLanguage

Diese Funktion ändert die Sprache des Projekts auf eine in der Übersetzungstabelle konfigurierte Übersetzung.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetLanguage	Übersetzung	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
SetLanguage(numLanguageCode)
```

Eingaben

numLanguageCode

Der Code für die zu aktivierende Sprache. Die Sprache muss bereits in der Tabelle Übersetzungstabelle, Seite 437 konfiguriert sein.

Rückgabewert Diese Funktion gibt die folgenden möglichen Werte aus:

Wert	Beschreibung
0	Fehler
1	Erfolgreich

Die Sprachcodes sind durch ISO 639-1 vorgegeben.

Beispiele Einstellen der Sprache auf „English – United States“:

SetLanguage(1033)

Einstellen der Sprache auf „Französisch – Frankreich“:

SetLanguage(1036)

19.17.4 SetTranslationFile

Legt die aktive Übersetzungsdatei fest und übersetzt alle freigegebenen Texte innerhalb des Projekts.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetTranslationFile	Übersetzung	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
SetTranslationFile(strFileName{ | ,optStrColumnName })
```

Eingaben

strFileName

Der Name der Übersetzungsdatei

optStrColumnName

Der Name der Spalte aus der Übersetzungsdatei, deren Texte für die Übsetzung verwendet werden sollen. Dieser Parameter ist optional. Defaultmäßig wird die zweite Spalte aus der Übersetzungsdatei verwendet.

Rückgabewert

Wert	Beschreibung
0	erfolgreich
−1	Ungültige Parameteranzahl.
−2	Falscher Parametertyp.
−3	Übersetzungsdatei konnte nicht gefunden oder nicht geöffnet werden.

Beispiele

Tag-Name	Ausdruck
Tag	SetTranslationFile("Portuguese.tra")
Tag	SetTranslationFile("German.tra")
Tag	SetTranslationFile("Mytranslation.csv" , "Portuguese")
Tag	SetTranslationFile("Mytranslation.csv" , "German")

☞ Zur korrekten Verwendung dieser Funktion muss die Übersetzungsoption aus dem Dialog Projekteinstellungen, Seite 67 beachtet werden.

HINWEIS **Im Übersetzungswerkzeug, Seite 437 muss sich eine Übersetzungsdatei befinden.**

19.18 Multimediafunktionen

19.18.1 Multimediafunktionen – Allgemeines

Diese Funktionen werden zum Abspielen externer Audio- und Videodateien verwendet.

19.18.2 Play

Spielt eine bestimmte WAV-Audiodatei ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Play	Multimedia	synchron	unterstützt	unterstützt	unterstützt

☞ Damit diese Funktion auf einem Thin Client funktioniert, muss sich die WAV-Zieldatei im gleichen Dateipfad wie auf der Remote-Station befinden.

Syntax

```
Play(strFileName{ | ,optNumSynchronous })
```

Eingaben

strFileName

Der Name und Dateipfad der abzuspielenden WAV-Datei.

optNumSynchronous

Ein numerischer Parameter, der festlegt, ob die Funktion synchron oder asynchron ausgeführt werden soll:

Wert	Beschreibung
0	Asynchron (d.h. die Laufzeitumgebung fährt fort, ohne auf die Beendung der Funktion zu warten)
1	Synchron (d.h. die Laufzeitumgebung hält an und wartet auf die Beendung der Funktion)

Dies ist ein optionaler Parameter; wird kein Wert angegeben, ist der Standardwert "0".

Rückgabewert

Diese Funktion gibt keinen Wert zurück.

Beispiele

Tag-Name	Ausdruck
	Play("C:\Sounds\Wav\alarm.wav")

19.19 Systeminformationsfunktionen

19.19.1 Systeminformationsfunktionen – Allgemeines

Diese Funktionen dienen dem Abrufen von Informationen über den Rechner, auf dem das Projekt ausgeführt wird (entweder Server oder Client, je nach Funktion), sowie dem Ändern einiger Projekteinstellungen auf diesem Rechner.

Systeminfo

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
DbVersion, Seite 688	synchron	unterstützt	unterstützt	unterstützt
GetAppHorizontalResolution, Seite 689	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
GetAppPath, Seite 689	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
GetAppVerticalResolution, Seite 690	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
GetComputerIP, Seite 690	synchron	unterstützt	unterstützt	unterstützt
GetComputerName, Seite 690	synchron	unterstützt	nicht unterstützt	unterstützt
GetCursorX, Seite 691	synchron	unterstützt	unterstützt	unterstützt
GetCursorY, Seite 691	synchron	unterstützt	unterstützt	unterstützt
GetDisplayHorizontalResolution, Seite 691	synchron	unterstützt	unterstützt	unterstützt
GetDisplayVerticalResolution, Seite 692	synchron	unterstützt	unterstützt	unterstützt
GetFreeMemoryCE, Seite 692	synchron	nicht unterstützt	unterstützt	unterstützt
GetHardKeyModel, Seite 693	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
GetHardKeySN, Seite 693	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Anhang: WinStudio-interne Skriptsprache

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
GetIPAll, Seite 694	synchron	unterstützt	unterstützt	unterstützt
GetMemoryCE, Seite 694	synchron	nicht unterstützt	unterstützt	unterstützt
GetNetMACID, Seite 695	synchron	unterstützt	unterstützt	unterstützt
GetOS, Seite 696	synchron	unterstützt	unterstützt	unterstützt
GetPrivateProfileString, Seite 696	synchron	unterstützt	unterstützt	unterstützt
GetProductPath, Seite 697	synchron	unterstützt	unterstützt	unterstützt
GetRegValue, Seite 697	synchron	unterstützt	unterstützt	nicht unterstützt
GetRegValueType, Seite 698	synchron	unterstützt	unterstützt	nicht unterstützt
GetServerHostName, Seite 699	synchron	nicht unterstützt	nicht unterstützt	unterstützt
GetTickCount, Seite 699	synchron	unterstützt	unterstützt	unterstützt
InfoAppAlrDir, Seite 700	synchron	unterstützt	unterstützt	unterstützt
InfoAppHST, Seite 700	synchron	unterstützt	unterstützt	unterstützt
InfoDiskFree, Seite 700	synchron	unterstützt	nicht unterstützt	unterstützt
InfoResources, Seite 701	synchron	unterstützt	unterstützt	unterstützt
IsActiveXReg, Seite 701	synchron	unterstützt	unterstützt	unterstützt
IsAppChangedOnServer, Seite 702	synchron	unterstützt	unterstützt	unterstützt
NoInputTime, Seite 703	synchron	unterstützt	unterstützt	unterstützt
ProductVersion, Seite 703	synchron	unterstützt	unterstützt	unterstützt
RegSaveCE, Seite 704	synchron	nicht unterstützt	unterstützt	nicht unterstützt
ReloadAppFromServer, Seite 704	synchron	unterstützt	unterstützt	unterstützt
SaveAlarmFile, Seite 705	synchron	unterstützt	unterstützt	nicht unterstützt
SetAppAlarmPath, Seite 706	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
SetAppHSTPath, Seite 706	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
SetDateFormat, Seite 707	synchron	unterstützt	unterstützt	unterstützt
SetKeyboardLanguage, Seite 707	synchron	unterstützt	unterstützt	unterstützt
SetRegValue, Seite 708	synchron	unterstützt	unterstützt	nicht unterstützt
SetWebConfig, Seite 709	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
SNMPGet, Seite 711	synchron	unterstützt	nicht unterstützt	unterstützt
SNMPSet, Seite 712	synchron	unterstützt	nicht unterstützt	unterstützt
WritePrivateProfileString, Seite 714	synchron	unterstützt	unterstützt	unterstützt

19.19.2 DbVersion

"DbVersion" ist eine interne Skript-Funktion, die die Versionsnummer Ihrer Projekt-Tag-Datenbank abruft.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DbVersion	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DbVersion()
```

Diese Funktion hat keine Parameter.

Rückgabewert Diese Funktion gibt einen numerischen Wert zurück, der gleich der Versionsnummer der Tag-Datenbank ist.

Diese Funktion gilt nur für die native Datenbank innerhalb Ihres Projekts. Es gibt derzeit keine Funktion zum Abrufen der Versionsnummer einer externen oder historischen Datenbank.

Beispiele *DbVersion - Beispiele*

```
DbVersion()
```

19.19.3 GetAppHorizontalResolution

Ruft die horizontale Standardbildschirmauflösung (in Pixeln) des Projekts ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetAppHorizontalResolution	Systeminfo	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
GetAppHorizontalResolution()
```

Diese Funktion hat keine Parameter.

Rückgabewert Gibt den Standardwert, wie in der Projektkonfiguration gespeichert, zurück.Die Windows-Konfigurationwird nicht dabei nicht berücksichtigt.

Beispiele

Tag-Name	Ausdruck
Tag	GetAppHorizontalResolution() // Returned value = 640
Tag	GetAppHorizontalResolution() // Returned value = 800

19.19.4 GetAppPath

Gibt den Dateipfad des Projektordners zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetAppPath	Systeminfo	synchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
GetAppPath()
```

Diese Funktion hat keine Parameter.

Rückgabewert Gibt den Dateipfad als Zeichenkette zurück.

Beispiele

Tag-Name	Ausdruck
Tag	GetAppPath() // Returned value = "C:\DemoProject\"
Tag	GetAppPath() // Returned value = "C:\Studio\Projects\project_name\"

☞ Die Funktion gibt immer ein "\" am Pfadende aus.

19.19.5 GetAppVerticalResolution

Ruft die vertikale Standardbildschirmauflösung (in Pixeln) des Projekts ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Avg	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetAppVerticalResolution()
```

Diese Funktion hat keine Parameter.

Rückgabewert Gibt den Standardwert, wie in der Projektkonfiguration gespeichert, zurück. Die Windows-Konfiguration wird dabei nicht berücksichtigt.

Beispiele

Tag-Name	Ausdruck
Tag	GetAppVerticalResolution() // Returned value = 480
Tag	GetAppVerticalResolution() // Returned value = 600

19.19.6 GetComputerIP

Gibt die erste IP-Adresse des lokalen Rechners zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetComputerIP	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetComputerIP()
```

Diese Funktion hat keine Parameter.

Rückgabewert Gibt die erste IP-Adresse der lokalen Station als Zeichenkette zurück.

Beispiele

Tag-Name	Ausdruck
Tag	GetComputerIP() // Returned value = "192.168.0.1"
Tag	GetComputerIP() // Returned value = "248.12.2.78"

19.19.7 GetComputerName

Gibt den lokalen Rechnernamen zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetComputerName	Systeminfo	synchron	unterstützt	nicht unterstützt	unterstützt

Syntax

```
GetComputerName()
```

Diese Funktion hat keine Parameter.

Rückgabewert Gibt den Namen des lokalen Rechners als Zeichenkette zurück.

Beispiele

Tag-Name	Ausdruck
Tag	GetComputerName() // Returned value = "Terminal53"
Tag	GetComputerName() // Returned value = "BobsComputer"

19.19.8 GetCursorX

Ruft die X-Koordinate des Mauszeigers auf dem Bildschirm ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetCursorX	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetCursorX()
```

Diese Funktion hat keine Parameter.

Rückgabewert Diese Funktion gibt die X-Koordinate des Cursors auf dem Bildschirm zurück, oder "–1" bei Auftreten eines Fehlers.

Beispiele

Tag-Name	Ausdruck
	GetCursorX( ) // Returned value = 1024

19.19.9 GetCursorY

Ruft die Y-Koordinate des Mauszeigers auf dem Bildschirm ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetCursorY	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetCursorY()
```

Diese Funktion hat keine Parameter.

Rückgabewert Diese Funktion gibt die Y-Koordinate des Cursors auf dem Bildschirm zurück, oder "–1" bei Auftreten eines Fehlers.

Beispiele

Tag-Name	Ausdruck
	GetCursorY( ) // Returned value = 768

19.19.10 GetDisplayHorizontalResolution

Ruft die horizontale Auflösung (in Pixel) des an die lokale Station angeschlossenen Bildschirms ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetDisplayHorizontalResolution	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetDisplayHorizontalResolution()
```

Diese Funktion hat keine Parameter.

Anhang: WinStudio-interne Skriptsprache

Rückgabewert Diese Funktion gibt die horizontale Auflösung der Bildschirms als Ganzzahl zurück.

Beispiele

Tag-Name	Ausdruck
	GetDisplayHorizontalResolution() // Returned value = 1024

19.19.11 GetDisplayVerticalResolution

Ruft die vertikale Auflösung (in Pixel) des an die lokale Station angeschlossenen Bildschirms ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetDisplayVerticalResolution	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetDisplayVerticalResolution()
```

Diese Funktion hat keine Parameter.

Rückgabewert Diese Funktion gibt die vertikale Auflösung der Bildschirms als Ganzzahl zurück.

Beispiele

Tag-Name	Ausdruck
	GetDisplayVerticalResolution( ) // Returned value = 768

19.19.12 GetFreeMemoryCE

Gibt den verfügbaren freien Speicherplatz eines Windows-Embedded-Geräts zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetFreeMemoryCE	Systeminfo	synchron	nicht unterstützt	unterstützt	unterstützt

Syntax

```
GetFreeMemoryCE({ | optNumType })
```

Eingaben

optNumType

Ein numerischer Merker zur Festlegung, welche Art von freiem Speicher das Projekt von dem Gerät abrufen soll:

`0 – Gesamter freier Speicherplatz`

Dies ist ein optionaler Parameter; wird kein Wert angegeben, ist der Standardwert "0".

Rückgabewert

Wert	Beschreibung
>0	Größe des freien Speichers in Bytes.
−1	Datei **Coredll.dll** nicht gefunden.
−2	Funktion GetMemoryCE, Seite 694 nicht gefunden.
−3	Ungültiger optionaler Parameter.
−4	Speicherart nicht verfügbar.

Beispiele

Tag-Name	Ausdruck
Tag	GetFreeMemoryCE()

19.19.13 GetHardKeyModel

Gibt den Modellnamen Ihres Hardkeys zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetHardKeyModel	Systeminfo	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
GetHardKeyModel()
```

Diese Funktion hat keine Parameter.

Rückgabewert

Wenn der Hardkey installiert ist, gibt die Funktion eine Zeichenkette mit dem Modellnamen des Hardkeys zurück.

Wenn der Hardkey nicht installiert ist, gibt die Funktion "0" zurück.

☞ Der Hardkey muss vor der Ausführung dieser Funktion angeschlossen werden, andernfalls wird die Funktion nicht korrekt ausgeführt.

Beispiele

Tag-Name	Ausdruck
Tag	GetHardKeyModel() // Returned value = "Local Interface"
Tag	GetHardKeyModel() // Returned value = "Advanced Server"

19.19.14 GetHardKeySN

Gibt die Seriennummer des Hardkeys zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetHardKeySN	Systeminfo	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
GetHardkeySN()
```

Diese Funktion hat keine Parameter.

Rückgabewert

Wenn der Hardkey installiert ist, gibt die Funktion eine Zeichenkette mit der Seriennummer des Hardkeys zurück.

Wenn der Hardkey nicht installiert ist, gibt die Funktion "0" zurück.

☞ Der Hardkey muss vor der Ausführung dieser Funktion angeschlossen werden, andernfalls wird die Funktion nicht korrekt ausgeführt.

Beispiele

Tag-Name	Ausdruck
Tag	GetHardkeySN() // Returned value = 120.745
Tag	GetHardkeySN() // Returned value = 224.941

Anhang: WinStudio-interne Skriptsprache

19.19.15 GetIPAll

Gibt alle IP-Adressen der lokalen Station zurück und speichert diese in einem String-Array-Tag.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetIPAll	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetIPAll( "tagArrayIP", optRefresh )
```

Eingaben

tagArrayIP

Name des String-Array-Tags, der die gefundenen IP-Adressen erhält. Dieser Name muss in Anführungszeichen gesetzt werden, andernfalls versucht das Projekt, den Inhalt des Array-Tags abzurufen.

optRefresh

"**Optionaler**" Tag, der eine Aktualisierung dieser Funktion auslöst. Bei jeder Wertänderung dieses Tags aktualisiert das Projekt den Rückgabewert der Funktion.

Rückgabewert

Wert	Beschreibung
"n"	Anzahl der gefundenen IP-Adressen
−1	Ungültige Anzahl Parameter
−2	Ungültiger Parametertyp

Beispiele

Tag-Name	Ausdruck
Tag	GetIPAll("TagArrayIP") // Returned value = 1
Tag	GetIPAll("TagArrayIP", Second) // Returned value = 2

19.19.16 GetMemoryCE

Gibt den verfügbaren Gesamtspeicherplatz eines Windows-CE-Gerätes zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetMemoryCE	Systeminfo	synchron	nicht unterstützt	unterstützt	unterstützt

Syntax

```
GetMemoryCE({ | optNumType })
```

Eingaben

optNumType

Numerischer Parameter zur Festlegung, welche Art von Speicher ermittelt werden soll:

Wert	Beschreibung
0	Gesamter Programmspeicher
1	Gesamter Speicherplatz
2	Gesamtspeicher

Dies ist ein optionaler Parameter; wird kein Wert angegeben, ist der Standardwert "0".

Rückgabewert

Wert	Beschreibung
>0	Größe des Speichers in Bytes.
−1	Datei **Coredll.dll** nicht gefunden.
−2	Funktion GetMemoryCE nicht gefunden.
−3	Ungültiger optionaler Parameter.

Beispiele

Tag-Name	Ausdruck
Tag	GetMemoryCE(1)

19.19.17 GetNetMACID

Ruft die MAC-Adresse der installierten Netzwerkadapter ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetNetMACID	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetNetMACID( "optTagMACID", "optTagAdapterName" )
```

Eingaben

optStrMACID

Name eines String-Tags, in welchem die MAC-Adresse des Netzwerkadapters gespeichert wird. Wenn mehr als ein Netzwerkadapter auf der Station installiert ist, kann für diesen Parameter ein Array- Tag vom Typ String konfiguriert werden..

optStrAdapterName

Name eines String-Tags, in welchem der Namen des Netzwerkadapters gespeichert werden soll. Wenn mehr als ein Netzwerkadapter auf der Station installiert ist, kann für diesen Parameter ein Array- Tag vom Typ String konfiguriert werden.

Rückgabewert

Wert	Beschreibung
>0	Anzahl der gefundenen Netzwerkadapter.
0	Keine Netzwerkadapter gefunden.
-1	Ungültige Anzahl Parameter.
-2	Einer der Parameter ist keine Zeichenkette.
-3	Der in **optTagMACID** konfigurierte Tag ist nicht vorhanden.
-4	Der in **optTagAdapterName** konfigurierte Tag ist nicht vorhanden.

Beispiele

Tag-Name	Ausdruck
NumNIC	GetNetMACID("MACIDTag")
NumNIC	GetNetMACID("MACIDTag", "AdapterName")
NumNIC	GetNetMACID("MACIDTag[1]", "AdapterName[1]")

19.19.18 GetOS

Ruft das aktuelle Betriebssystem ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetOS	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetOS()
```

Diese Funktion hat keine Parameter.

Rückgabewert

Wert	Beschreibung
0	Windows 3,11
1	Windows 95/98/ME
2	Windows 2000/XP/Vista/7
3	Windows CE/Mobile

Beispiele

Tag-Name	Ausdruck
Tag	GetOS() //Returned value = 2

19.19.19 GetPrivateProfileString

Liest unter Verwendung des Standardformats von *.ini-Dateine einen bestimmten Parameter aus einer solchen Datei aus.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetPrivateProfileString	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetPrivateProfileString( strSection, strName, strDefault, strFileName )
```

Eingaben

strSection

Der zu lesende Abschnittsname in der ini-Datei.

strName

Der zu lesende Parameter in der ini-Datei.

strDefault

Die Standardeinstellung für den zu lesenden Parameter "strName". Wird der Parameter in der **.ini**-Datei nicht gefunden,gibt die Funktion den hier eingetragenen Standardwert zurück.

strFileName

Der Name und Dateipfad der auszulesenden **.ini**-Datei.

Rückgabewert Gibt den Wert des angegebenen Parameters zurück.

Beispiele

Tag-Name	Ausdruck
Tag	GetPrivateProfileString("boot loader", "timeout", "50", "C:\boot.ini") // Returned value = 30

19.19.20 GetProductPath

Ruft den Pfad zum Programmverzeichnis ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetProductPath	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetProductPath()
```

Diese Funktion hat keine Parameter.

Rückgabewert Gibt den Pfad zum Programmverzeichnis als Zeichenkette zurück.

Beispiele

Tag-Name	Ausdruck
Tag	GetProductPath() // Returned value = "C:\Program Files\Rexroth \IndraWorks\WinStudio"

19.19.21 GetRegValue

Ruft den Wert eines Registry-Eintrags in der Windows-Registry ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetRegValue	Systeminfo	synchron	unterstützt	unterstützt	nicht unterstützt

Syntax

```
GetRegValue( numMainKey, strKey, strValueName )
```

Eingaben

numMainKey

Numerischer Tag zur Auswahl des Hauptschlüssels (siehe Tabelle):

Wert	Beschreibung
0	HKEY_LOCAL_MACHINE
1	HKEY_CLASSES_ROOT
2	HKEY_CURRENT_USER
3	HKEY_USERS
4	HKEY_CURRENT_CONFIG
5	HKEY_PERFORMANCE_DATA

strKey

Pfad für den auszulesenden Registry-Eintrag unterhalb des Hauptschlüssels (String-Variable oder Zeichenkette in Anführungszeichen)

strVariableName

Name des abzurufenden Registry-Eintrags. Die maximale Länge beträgt 255 Zeichen.

Rückgabewert

Wenn die Funktion erfolgreich ist, wird der Wert des Registry-Eintrags zurückgegeben. Andernfalls wird einer der folgenden Fehlercodes zurückgegeben:

Wert	Beschreibung
−1	Ungültige Anzahl Parameter oder ungültiger Hauptschlüssel.
−2	Variablentyp wird nicht unterstützt. Aus der Registry können nur DWord- oder String-Werte ausgelesen werden.
−3	Variablenwert konnte nicht gelesen werden; bitte überprüfen Sie die Sicherheitseinstellungen.

Beispiele

Tag-Name	Ausdruck
Tag	GetRegValue(0, "HARDWARE\DESCRIPTION\System", "SystemBiosDate") // Returned value = "08/14/03"
Tag	GetRegValue(2, "Control Panel\Current", "Color Schemes") // Returned value = "Windows Standard "

19.19.22 GetRegValueType

Ruft den Datentyp eines Registry-Eintrags in der Windows-Registry ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetRegValue-Type	Systeminfo	synchron	unterstützt	unterstützt	nicht unterstützt

Syntax

```
GetRegValueType( numMainKey, strKey, strValueName )
```

Eingaben

numMainKey

Numerischer Tag zur Auswahl des Hauptschlüssels (siehe Tabelle):

Wert	Beschreibung
0	HKEY_LOCAL_MACHINE
1	HKEY_CLASSES_ROOT
2	HKEY_CURRENT_USER
3	HKEY_USERS
4	HKEY_CURRENT_CONFIG
5	HKEY_PERFORMANCE_DATA

strKey

Pfad des Registry-Eintrags unterhalb des Hauptschlüssels

strVariableName

Name des Registry- Eintrags, dessen Datentyp ermittelt werden soll.. Die maximale Länge beträgt 255 Zeichen.

Rückgabewert

Wert	Beschreibung
1	Der Variablentyp ist String.
0	Der Variablentyp ist DWord.
−1	Ungültige Anzahl Parameter oder ungültiger Hauptschlüssel.
−2	Variablentyp wird nicht unterstützt. Aus der Registry können nur DWord- oder String-Werte ausgelesen werden.
−3	Variablenwert konnte nicht gelesen werden; bitte überprüfen Sie die Sicherheitsberechtigungen.

Beispiele

Tag-Name	Ausdruck
Tag	GetRegValueType(0, "HARDWARE\DESCRIPTION\System", "SystemBiosDate") // Returned value = 1
Tag	GetRegValueType(2, "Control Panel\Desktop", "Smooth Scroll") // Returned value = 0

19.19.23 GetServerHostName

Ruft den Host-Namen der Serverstation, auf welchem die Runtimeumgebung läuft, ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetServerHost-Name	Systeminfo	synchron	nicht unterstützt	nicht unterstützt	unterstützt

Syntax

```
GetServerHostName()
```

Diese Funktion hat keine Parameter.

Rückgabewert Server-Host-Name für ISSymbol und 127.0.0.1 für andere.

19.19.24 GetTickCount

Ruft den aktuellen Wert des Taktzählers der Uhr ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetTickCount	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetTickCount()
```

Diese Funktion hat keine Parameter.

Rückgabewert Gibt die Anzahl der von der internen Uhr seit Initialisierung des Betriebssystems gezählten Millisekunden zurück.

Beispiele

Tag-Name	Ausdruck
Tag	GetTickCount() // Returned value = 9400907

19.19.25 InfoAppAlrDir

Gibt den Dateipfad des Projektunterordners Alarm zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
InfoAppAlrDir	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
InfoAppAlrDir()
```

Diese Funktion hat keine Parameter.

Rückgabewert Gibt das Verzeichnis **Alarm** des aktuellen Projekts als Zeichenkette zurück.

Beispiele

Tag-Name	Ausdruck
Tag	InfoAppAlrDir() // Returned value = "C:\DemoApp\Alarm\"
Tag	InfoAppAlrDir() // Returned value = "C:\Studio\Projects\App\Alarm\"

19.19.26 InfoAppHSTDir

Gibt den Dateipfad des Projektunterordners Historie zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
InfoAppHSTDir	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
InfoAppHstDir()
```

Diese Funktion hat keine Parameter.

Rückgabewert Gibt das Verzeichnis **Historie** des aktuellen Projekts als Zeichenkette zurück.

Beispiele

Tag-Name	Ausdruck
Tag	InfoAppHSTDir() // Returned value = "C:\DemoApp\HST\"
Tag	InfoAppHSTDir() // Returned value = "C:\Studio\Projects\App\HST\"

19.19.27 InfoDiskFree

Gibt den freien Festplatten-Speicherplatz auf dem lokalen Rechner zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
InfoDiskFree	Systeminfo	synchron	unterstützt	nicht unterstützt	unterstützt

Syntax

```
InfoDiskFree(strDisk)
```

Eingaben

strDisk

Der Name des zu überprüfenden Laufwerks.

Rückgabewert Gibt den verfügbaren freien Speicherplatz des Laufwerks in Bytes zurück.

Beispiele

Tag-Name	Ausdruck
Tag	InfoDiskFree("C") // Returned value = 2803804605.000000

19.19.28 InfoResources

Gibt die verfügbaren Ressourcen des lokalen Rechners zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
InfoResources	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
InfoResources(numSelect)
```

Eingaben

numSelect

Numerischer Parameter zur Festlegung, welche Ressource untersucht werden soll:

Wert	Beschreibung
0	Systemfunktionen (%)
1	GDI-Funktionen (%)
2	BENUTZER-Funktionen (%)
3	Speicher (in Bytes)

Beispiele

Tag-Name	Ausdruck
Tag	InfoResources(0) // Returned value = 76,000000
Tag	InfoResources(1) // Returned value = 76,000000
Tag	InfoResources(2) // Returned value = 80,000000
Tag	InfoResources(3) // Returned value = 16150528.000000

☞ Die einzige zulässige Auswahl an einem Windows PC ist **3**. Auswahl von **0**, **1** oder **2** gibt nur das Ergebnis **0.000000** zurück.

19.19.29 IsActiveXReg

Prüft, ob ein ActiveX-Steuerelement im Betriebssystem registriert ist.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
IsActiveXReg	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
IsActiveXReg( numType, strProgIDorFileName )
```

Eingaben

numType

Numerischer Parameter, der das Format des zweiten Parameters **strProgIDorFileName** festlegt:

Wert	Beschreibung
0	Überprüfen durch Programm-ID
1	Überprüfen durch Dateiname

strProgIDorFileName

Die Programm-ID oder der Dateiname inklusive Pfad des ActiveX-Steuerelements.

Rückgabewert

Wert	Beschreibung
0	ActiveX ist nicht registriert.
1	ActiveX ist registriert.

Beispiele

Tag-Name	Ausdruck
Tag	IsActiveXReg(0, "ISSYMBOL.ISSymbolCtrl.1") // Returned value = 0
Tag	IsActiveXReg(1, "C:\WinNT\system32\MediaPlayer.ocx") // Returned value = 1

19.19.30 IsAppChangedOnServer

Bei Ausführung auf dem Client prüft diese Funktion, ob die auf dem Server verfügbaren Projektdateien neuer als die aktuell auf dem Client vorhandenen Dateien sind.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
IsAppChange-dOnServer	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
IsAppChangedOnServer({ "optTagUpdateTrigger" })
```

Eingaben

optTagUpdateTrigger

Ein alphanumerischer String in Anführungszeichen oder der Name eines String-Tags, Seite 124, der die gewünschte Zeichenkette enthält. Diese Zeichenkette sollte wiederum der Name des Tags sein, der die Funktion auslöst – wenn sich der Tag-Wert ändert, wird die Funktion automatisch ausgeführt.

Dieser Parameter ist optional.

☞ Die Funktion nur dann ausgeführt, wenn sie ausdrücklich aufgerufen wird, wie bei der Befehlsanimation, Seite 243. Zum Ausführen der Funktion in regelmäßigen Zeitabständen kann einer der System-Tags, Seite 821 des Projekts wie **Tag** oder **Monat** verwendet werden.

Rückgabewert

Wert	Beschreibung
0	Nein
1	Ja (Die Dateien auf dem Server sind neuer)

Für diese Funktion bedeutet „Server“ die Station, auf der die Runtimeumgebung ausgeführt wird und auf der das TCP/IP-Servermodul aktiviert ist. Unter „Client“ wird ein Thin Client oder Secure Viewer verstanden, der über TCP/IP mit dem Server kommuniziert. Weitere Informationen unter Konfigurieren einer Web-Lösung, Seite 449“.

Wenn die Dateien auf dem Server neuer sind – d.h. wenn die Funktion WAHR ausgibt – kann ReloadAppFromServer, Seite 704 zum Aktualisieren des Clients verwendet werden.

Beispiele

Tag-Name	Ausdruck
Tag	IsAppChangedOnServer()
Tag	IsAppChangedOnServer("CheckVersion") // Function is automatically called when the value of **CheckVersion** changes.

19.19.31 NoInputTime

Gibt die seit der letzten Tastatureingabe vergangene Zeit an.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
NoInputTime	Systeminfo	synchron	unterstützt	unterstützt	nur Tastatureingabe

Syntax

```
NoInputTime({ "optTagUpdateTrigger" })
```

Eingaben

optTagUpdateTrigger

"Optionaler" Tag, der eine Aktualisierung auslöst, wenn diese Funktion in einem Textausgabe-Objekt, Seite 249 verwendet wird. Bei jeder Wertänderung dieses Tags wir der Rückgabewert aktualisiert.

Rückgabewert

Gibt die seit der letzten Tastatureingabe vergangene Zeit (in Sekunden) an.

Beispiele

Tag-Name	Ausdruck
Tag	NoInputTime()

Diese Funktion kann nicht direkt aus einem Textobjekt implementiert werden.

19.19.32 ProductVersion

Gibt die Nummer der WinStudio-Version zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ProductVersion	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ProductVersion()
```

Diese Funktion hat keine Parameter.

Rückgabewert

Gibt die Versionsnummer von WinStudio als reelle Zahl zurück.

Beispiele

Tag-Name	Ausdruck
Tag	ProductVersion() // Returned value = 7.000000

19.19.33 RegSaveCE

Speichert die WindowsCE-System-Registry. Diese Funktion kann nur angewendet werden, wenn die Fähigkeit zum Speichern der Registry im Image des Windows-CE-Geräts aktiviert ist.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
RegSaveCE	Systeminfo	synchron	nicht unterstützt	unterstützt	nicht unterstützt

Syntax

```
RegSaveCE()
```

Diese Funktion hat keine Parameter.

Rückgabewert

Wert	Beschreibung
0	erfolgreich
−1	HKEY_CLASSES_ROOT konnte nicht gespeichert werden.
−2	HKEY_CURRENT_USER konnte nicht gespeichert werden.
−3	HKEY_LOCAL_MACHINE konnte nicht gespeichert werden.
−4	HKEY_USERS konnte nicht gespeichert werden.
−5	Funktion auf NT-Plattform ausgeführt.

☞ Diese Funktion ruft die Funktion **RegFlushKey** aus dem Windows CE API auf. Die Funktionalität ist von der Gerätefirmware des CE Gerätes abhängig und wird von der IndraControl VEx Gerätefamilie (WinCE) unterstützt.

Beispiele

Tag-Name	Ausdruck
n/a	RegSaveCE( ) // Returned value = 0 if successful

19.19.34 ReloadAppFromServer

Bei Ausführung auf dem Client lädt diese Funktion die benötigten Projektdateien erneut vom Server, wobei der aktuelle Zustand des Projekts auf dem Client beibehalten wird.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ReloadAppFrom-Server	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ReloadAppFromServer()
```

Diese Funktion hat keine Parameter.

Rückgabewert Diese Funktion gibt immer 0 zurück.

Für diese Funktion bedeutet „Server“ die Station, auf der die Runtimeumgebung ausgeführt wird und auf der das TCP/IP-Servermodul läuft, und „Client“ bedeutet einen Thin Client oder Secure Viewer, der über TCP/IP mit dem Server kommuniziert. Weitere Informationen unter Konfigurieren einer Web-Lösung, Seite 449“.

Vor dem Aufrufen dieser Funktion kann IsAppChangedOnServer, Seite 702 verwendet werden, um die Version der bereits auf dem Client vorhandenen Projektdateien zu überprüfen. Wenn die Dateien auf dem Client mit den Dateien auf dem Server identisch sind, ist ein Aufruf dieser Funktion nicht erforderlich.

Beispiele

Tag-Name	Ausdruck
Tag	ReloadAppFromServer( )

19.19.35 SaveAlarmFile

Verwenden Sie diese Funktion zum Aktivieren/deaktivieren der Speicherfunktion für Alarmhistorie und zum Festlegen des Pfads, in dem die Alarmhistoriendateien gespeichert werden.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SaveAlarmFile	Systeminfo	synchron	unterstützt	unterstützt	nicht unterstützt

Syntax

```
SaveAlarmFile( numType, optRemotePath )
```

Eingaben

numType

Legt fest, ob die Alarmdatei gespeichert werden soll:

Wert	Beschreibung
0	Speichern der Alarmdatei auf die lokale Festplatte deaktivieren
1	Speichern der Alarmdatei auf die lokale Festplatte aktivieren
2	Speichern der Alarmdatei auf die lokale Festplatte und in den im Parameter "OptRemotePath" festgelegten Remote-Pfad aktivieren

optRemotePath

Tag mit dem Pfad und Namen des Remote-Rechners, in dem die Alarmdatei zusätzlich zum lokalen Rechner gespeichert wird, wenn **numType** gleich 2 ist.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Zweiter Parameter ist kein String
2	Zweiter Parameter fehlt

Beispiele

Tag-Name	Ausdruck
Tag	SaveAlarmFile(0)
Tag	SaveAlarmFile(1)
Tag	SaveAlarmFile(2, "Z:\Apps\AppDemo")

19.19.36 SetAppAlarmPath

Legt den "Alarm"-Pfad für das aktuelle Projekt fest.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetAppAlarmPath	Systeminfo	synchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
SetAppAlarmPath(strPath)
```

Eingaben

strPath

Der neue **"Alarm"**-Pfad für das aktuelle Projekt.

Rückgabewert

Diese Funktion gibt keinen Wert zurück.

Beispiele

Tag-Name	Ausdruck
	SetAppAlarmPath("C:\Studio\Alarm\")

19.19.37 SetAppHSTPath

Legt den Dateipfad (Verzeichnis) fest, in dem Trendhistoriendateien im proprietären Format (**.HST**) gespeichert werden.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetAppHSTPath	Systeminfo	synchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
SetAppHSTPath(strPath)
```

Eingaben

strPath

Der Dateipfad (Verzeichnis), in dem die Trendhistoriendateien gespeichert werden.

Rückgabewert

Diese Funktion gibt keinen Wert zurück.



Diese Funktion ist hilfreich, wenn der Dateipfad während der Laufzeit geändert werden soll. Sie können den Dateipfad auch auf ein Netzlaufwerk verweisen lassen, indem Sie diesen auf der lokalen Station abbilden oder durch Verwendung der folgenden Syntax:

\Netzwerkpfad\Dateipfad

Bitte beachten Sie, dass diese Funktion bestehende Historiendateien **nicht** aus dem Standardverzeichnis in ein neues kopiert, sondern nur den Dateipfad für die neuen Historiendateien festlegt, die nach dem Aufruf der Funktion gespeichert werden.

Beispiele

Tag-Name	Ausdruck
	SetAppHstPath("C:\Studio\History\")

19.19.38 SetDateFormat

Legt das Format und das Trennzeichen für den Datums-String fest.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetDateFormat	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
SetDateFormat( strSeparator, strFomat )
```

Eingaben

strSeparator

Das Trennzeichen für den Datums-String.

strFormat

String-Tag zur Festlegung der Reihenfolge von Monat (M), Tag (D) und Jahr (Y) im Datums-String.

Wert	Beschreibung
DMY	Tag, Monat, Jahr
MDY	Monat, Tag, Jahr
YMD	Jahr, Monat, Tag

Rückgabewert

Wert	Beschreibung
0	Kein Fehler
1	Ungültiger Parameter

Beispiele

Tag-Name	Ausdruck
Tag	SetDateFormat("/", "MDY") // Date = 04/18/2002
Tag	SetDateFormat(":", "MYD") // Date = 04:2002:18

19.19.39 SetKeyboardLanguage

Stellt die Sprache der virtuellen Tastaturen im Projekt ein.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetKeyboardLan-guage	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
SetKeyboardLanguage(strLanguage)
```

Eingaben

strLanguage

String-Tag mit dem für die virtuelle Tastatur verwendeten Sprachcode. Die aktuell verfügbaren Optionen sind u.a.:

Anhang: WinStudio-interne Skriptsprache

Wert	Beschreibung
EN	Englisch (Standard)
GE	Deutsch

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Fehler

Beispiele

Tag-Name	Ausdruck
Tag	SetKeyboardLanguage("EN")
Tag	SetKeyboardLanguage(tagLanguage)

19.19.40 SetRegValue

Legt den Wert eines Registry-Einrags fest.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetRegValue	Systeminfo	synchron	unterstützt	unterstützt	nicht unterstützt

Syntax

```
SetRegValue( numMainKey, strKey, strVariableName, numType, strOrNumValue )
```

Eingaben

numMainKey

Numerischer Tag zur Auswahl des Hauptschlüssels (siehe Tabelle):

Wert	Beschreibung
0	HKEY_LOCAL_MACHINE
1	HKEY_CLASSES_ROOT
2	HKEY_CURRENT_USER
3	HKEY_USERS
4	HKEY_CURRENT_CONFIG
5	HKEY_PERFORMANCE_DATA

strKey

Pfad des Registry-Eintrags unterhalb des Hauptschlüssels

strVariableName

Name des Registry Eintrags, der geändert werden soll. Die maximale Länge beträgt 255 Zeichen.

numType

Format des Registry Eintrags, der geändert werden soll (zur Zeit unterstützt: siehe Tabelle)

Wert	Beschreibung
0	DWord
1	Zeichenkette

strOrNumValue

Wert der geschrieben werden soll (String-Variable oder Zeichenkette in " ").

Rückgabewert

Wert	Beschreibung
0	erfolgreich
-1	Ungültige Anzahl Parameter oder ungültiger Hauptschlüssel.
-2	Ungültiger Typ.
-3	Registry-Schlüssel konnte nicht gelesen werden; bitte überprüfen Sie die Sicherheitsberechtigungen.

Beispiele

Tag-Name	Ausdruck
Tag	SetRegValue(0, "HARDWARE\DEVICEMAP\SERIALCOMM", "\Device \Serial1", 1, "COM3") // Returned value = 0 if successful
Tag	SetRegValue(2, "Control Panel\Desktop", "Smooth Scroll", 0, 1) // Returned value = 0 if successful

HINWEIS **Dieser Eintrag kann sich auf die Windows-Systemkonfiguration auswirken.**

Sie sollten mit größter Vorsicht vorgehen und die Registry "nur" bearbeiten, wenn Sie sich bezüglich der Konfiguration sicher sind.

19.19.41 SetWebConfig

Konfiguriert die Web-Einstellungen für das aktuelle Projekt. Diese Funktion aktualisiert die Einstellungen in den HTML-Dateien des Projekts.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetWebConfig	Systeminfo	synchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
SetWebConfig( strServerIP, optStrBackupURL, optStrPathFile, optNumHostPort, optStrSecondaryServerIP,
              optNumProtocolFlag, optNumGtwPort, optStrGtwIP, optStrSecondaryGtwIP, optStrISSymbolURL )
```

Eingaben

strServerIP

Datenserver IP-Adresse. IP-Adresse (oder Host-Name) des Rechners, auf dem das TCP-Server-Modul von "WinStudio" ausgeführt wird.

optStrBackupURL

Alternative URL für die Webseiten des Projekts (optional). Der Thin Client sucht in dieser URL nach den Webseiten, wenn diese nicht in der gleichen URL gefunden werden, die in das Adressfeld des Webbrowsers eingetragen ist.

optStrPathFile

Dateipfad und -name der zu aktualisierenden HTML-Datei (optional). Wird nur der Dateipfad ohne einen Dateinamen angegeben, werden alle HTML-Dateien im angegebenen Dateipfad aktualisiert.

☞ Es kann eine einzelne Datei angegeben werden (z.B. wenn nur eine Datei aktualisiert werden soll. Dies ist besonders hilfreich bei Projekten, die auf Windows-CE-Zielsystemen laufen.

optNumHostPort

TCP-Port-Nummer, die der Thin Client zum Datenaustausch mit dem TCP-Servermodul von "WinStudio" verwenden muss (optional).

optStrSecondaryServerIP

Alternative IP-Adresse für den Datenserver (optional). Der Thin Client versucht, sich mit dem TCP-Servermodul von "WinStudio" unter dieser IP-Adresse zu verbinden, wenn eine Verbindung mit dem TCP-Servermodul, das unter der im Parameter strServerIP angegebenen IP-Adresse ausgeführt wird, nicht zustande kommt.

optNumProtocolFlag

Bei Verwendung der Option Web Tunneling Gateway legt dieser Parameter fest, ob der Thin Client HTTP oder HTTPS (SSL – Secure Socket Layer) zum Datenaustausch mit dem Webserver verwendet (optional). Hat dieser Merker den Wert 0, verwendet der Thin Client HTTP. Hat dieser Merker den Wert 1, verwendet der Thin Client HTTPS (SSL).

optNumGtwPort

TCP-Port-Nummer, die der Thin Client bei Verwendung von Web Tunneling Gateway zum Datenaustausch mit dem Webserver verwenden muss (optional).

optStrGtwIP

IP-Adresse (oder Host-Name) des Rechners, auf dem der Web Tunneling Gateway ausgeführt wird (optional).

optStrSecondaryGtwIP

Alternative IP-Adresse (oder Host-Name) des Rechners, auf dem der Web Tunneling Gateway ausgeführt wird (optional). Der Thin Client versucht, sich mit dem Web Tunneling Gateway unter dieser IP-Adresse zu verbinden, wenn eine Verbindung mit dem Web Tunneling Gateway, der unter der im Parameter optStrGtwIP angegebenen IP-Adresse ausgeführt wird, nicht zustande kommt.

optStrISSymbolURL

URL, von der die aktualisierte Version von ISSymbol (ActiveX-Steuerelement) heruntergeladen werden muss, wenn diese nicht ordnungsgemäß in der Thin-Client-Station registriert ist (optional).

- Tags oder Ausdrücke können in dieser Funktion als Argumente verwendet werden. Daher kann diese Funktion zum automatischen Konfigurieren der Webeinstellungen während der Laufzeit verwendet werden, je nach Netzwerkeinstellungen des jeweiligen Projekts (IP-Adresse, Webserver-URL usw.)
- Nur der erste Parameter dieser Funktion ("strServerIP") muss zwingend angegeben werden. Alle anderen Parameter sind optional. Die Parameter, die nicht in der Funktion konfiguriert wurden, nehmen den in der Registerkarte Web, Seite 86 der Projekteinstellungen konfigurierten Standardwert an.
- Die folgenden Parameter sollten nur verwendet werden, wenn der Web Tunneling Gateway verwendet wird: "optNumProtocolFlag", "optNumGtwPort", "optStrGtwIP", "optStrSecondaryGtwIP" und "optStrISSymbolURL".

Rückgabewert

Fehler	Beschreibung
0	Kein Fehler
1	Ungültige Anzahl Parameter
2	Ungültige Server-IP-Adresse 1
3	Ungültige URL
4	Ungültiger optionaler Pfad
5	Keine Webseiten gefunden

Beispiele

Tag-Name	Ausdruck
Tag	SetWebConfig("192.168.1.28")
Tag	SetWebConfig("192.168.1.28", "http://192.168.1.28/")
Tag	SetWebConfig(GetComputerIP(), "http://" + GetComputerIP() + "/")
Tag	SetWebConfig("192.168.1.28", "http://192.168.1.28/", "C:\MyWebPages\")
Tag	SetWebConfig("192.168.1.28", "http://192.168.1.28/", "C:\MyWebPages\", 1234)
Tag	SetWebConfig("192.168.1.28", "http://200.0.0.10/", "C:\MyWebPages\", 1234, "192.168.1.29", 0, 80, "200.0.0.1", "200.0.0.10", "http://200.0.0.10/MyISSymbol/")

19.19.42 SNMPGet

Ruft Informationen von Rechnern oder Netzwerkgeräten über das Simple Network Management Protocol (SNMP) ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SNMPGet	Systeminfo	synchron	unterstützt	nicht unterstützt	unterstützt

Syntax

```
SNMPGet( strAddress, strCommunity, strOID, "strTagName" )
```

Anhang: WinStudio-interne Skriptsprache

Eingaben

strAddress

Die Adresse des Rechners/Computers (z.B. **"127.0.0.1"** oder **"localhost"**).

strCommunity

SNMP-Community-Name bei der Kommunikation mit dem Rechner (z.B. **"public"**).

strOID

Zu konsultierende OID (z.B. **".1.3.6.1.2.1.1.1.0"**).

strTagName

Name des Tags, der den angefragten Wert erhält.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert

Wert	Beschreibung
0	Kein Fehler
−1	Ungültige Anzahl Parameter
−2	Ungültiger Parameter
−3	Keine Verbindung zum Remote-Gerät
−4	Keine Verbindung zum Remote-Gerät
−5	Vorgang GET fehlgeschlagen
−6	Ungültige OID
−7	Ungültiger Tag-Name
−8	Ungültiger Tag-Typ
−9	Diese Funktion wird vom aktuellen Betriebssystem nicht unterstützt

Beispiele

Tag-Name	Ausdruck
ErrorTag	SNMPGet("127.0.0.1", "public", ".1.3.6.1.2.1.1.1.0", "SysDescrTag") // ErrorTag will receive the error code. If the function succeeds, the value in the OID ".1.3.6.1.2.1.1.1.0" will be saved in the tag SysDescrTag.

19.19.43 SNMPSet

Verwendet das Simple Network Management Protocol (SNMP) zum Setzen eines Werts auf einem Zielrechner oder Netzwerkgerät.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SNMPSet	Systeminfo	synchron	unterstützt	nicht unterstützt	unterstützt

Syntax

```
SNMPSet( strAddress, strCommunity, strOID, Value, optNumType )
```

Eingaben

strAddress

Die Adresse des Zielrechners oder Geräts (z.B. "127.0.0.1" oder "localhost").

strCommunity

Der SNMP-Community-Name (z.B. "public") bei der Kommunikation mit dem Zielrechner oder -gerät.

strOID

Die zu setzende Objekt-ID (OID).

Wert

Der in der angegebenen OID zu setzende Wert.

optNumType

Ein numerischer Wert oder ein Tag vom Typ Integer, Seite 124 zur Festlegung des Datentyps. Dieser Parameter ist optional, aber bei Verwendung muss er einen der folgenden Werte haben:

Wert	Typ	Beschreibung
0	OCTETSTRING	Eine ctet-String-Variable.
1	INTEGER32	Eine mit 32-Bit signierte Integer-Variable
2	TIMETICKS	Eine Zeittakt-Variable
3	GAUGE32	Messvariable
4	COUNTER32	Zählervariable
5	IPADDRESS	IP-Adress-Variable
6	OBJECTIDENTIFIER	Objektkennungsvariable
7	SEQUENCE	ASN-Sequenz-Variable
8	OPAQUE	Eine intransparente Variable

Rückgabewert

Wert	Beschreibung
0	Kein Fehler
−1	Ungültige Anzahl Parameter
−2	Ungültiger Parameter
−3	Keine Verbindung zum Remote-Gerät
−4	Keine Verbindung zum Remote-Gerät
−5	Vorgang SET fehlgeschlagen
−6	Ungültige OID
−7	Ungültiger Tag-Name
−8	Ungültiger Tag-Typ
−9	Diese Funktion wird vom aktuellen Betriebssystem nicht unterstützt

Beispiele

Tag-Name	Ausdruck
	SNMPSet("127.0.0.1", "public", ".1.3.6.1.2.1.1.1.0", 123, 1) //Sets an integer value of 123 to the specified OID on the localhost (127.0.0.1).

19.19.44 WritePrivateProfileString

Schreibt eine festgelegte Einstellung in die Initialisierungsdatei für den Projekt-Viewer, unter Verwendung des Standardformats der **.ini**-Datei.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
WritePrivateProfileString	Systeminfo	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
WritePrivateProfileString ( strSection, strName, strValue, strFileName )
```

Eingaben

strSection

Zu schreibender Abschnittsname in der ini-Datei.

strName

Zu schreibender Parameter innerhalb des gewählten Abschnitts

strValue

Zu schreibender Wert.

strFileName

Name und Pfad der zu schreibenden .ini-Datei.

Rückgabewert

Die Funktion gibt den Wert 1 zurück, wenn die Datei erfolgreich aktualisiert wurde.

☞ Die Funktion schreibt bei Ausführung auf einem Windows CE-Gerät die gesamte Datei neu, daher wird die Verwendung für längere Dateien auf Windows-CE-Geräten nicht empfohlen. Die Funktion fügt außerdem bei Ausführung auf einem Windows-CE-Gerät die folgenden Zeilen am Dateiende an: [FileBackUpControl] Valid=1

Beispiele

Tag-Name	Ausdruck
Tag	WritePrivateProfileString(Section, Name, Value, FileName)
Tag	WritePrivateProfileString("Options", "ds1", "Value", "C:\Viewer.ini")

19.20 Tag-Datenbank-Funktionen

19.20.1 Tag-Datenbank-Funktionen – Allgemeines

Diese Funktionen werden zum direkten Ändern der Werte von Projekt-Tags verwendet.

Tag-Datenbank

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
ExecuteAlarmAck, Seite 715	synchron	unterstützt	unterstützt	unterstützt
ForceTagChange, Seite 716	synchron	unterstützt	unterstützt	unterstützt
GetTagValue, Seite 716	synchron	unterstützt	unterstützt	unterstützt
SetTagValue, Seite 717	synchron	unterstützt	unterstützt	unterstützt

19.20.2 ExecuteAlarmAck

Diese Funktion quittiert einen aktiven Alarm mit dem angegebenen Tag. Der Vorteil dieser Funktion ist, dass bei Verwendung vom Thin Client der Alarm-Task den Benutzernamen und die Station, von der aus der Alarm bestätigt wurde, speichert.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ExecuteAlarmAck	Tag-Datenbank	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ExecuteAlarmAck( "strTagName", optStrComment, optStrAlarmType )
ExecuteAlarmAck("strTagName",{ | ,optStrComment{ | ,optStrAlarmType } })
```

Eingaben

strTagName

Name des Tags, dessen Alarm quittiert werden soll



Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

optStrComment

Ein optionaler Kommentar zum Senden an den Alarm-Task, zusammen mit Benutzername und Station.

optStrAlarmType

Wenn auf dem angegebenen Tag mehr als ein Alarm aktiv ist, kann festgelegt werden, welcher Alarm bestätigt werden soll (z.B. Hi, Lo, HiHi, LoLo). Andernfalls bestätigt die Funktion den zuletzt aktivierten Alarm.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich ausgeführt.
-1	Ungültige Anzahl Parameter.
-2	Ungültiger Tag-Name.
-3	Ausgeführt, aber ohne Warten auf Bestätigung vom Alarm-Task. Siehe Hinweis.

☞ Wenn diese Funktion zum Bestätigen eines Alarms verwendet wird, wartet sie in der Regel auf Bestätigung durch den Alarm-Task und wird dann als Anzeige für die erfolgreiche Ausführung auf den Wert "0" zurückgesetzt. In einigen Fällen kann das Warten auf die Bestätigung allerdings dazu führen, dass sich die Laufzeitumgebung aufhängt. Wenn dies passiert und die Funktion korrekt mit gültigen Parametern aufgerufen wurde, wird sie wie vorgesehen ausgeführt, wartet aber nicht auf Bestätigung.

Beispiele Bestätigen des aktiven Alarms Hi auf Tag **A** mit dem Kommentar **Alarm Hi auf Tag A**:

ExecuteAlarmAck ("A", "Hi alarm on tag A", "Hi")

19.20.3 ForceTagChange

Erzwingt das Schreiben eines Werts in einen Tag in die Datenbank, so dass der neue Wert auch dann als Wertänderung erkannt wird, wenn er sich gegenüber dem alten Wert nicht geändert hat.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ForceTagChange	Tag-Datenbank	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
ForceTagChange( "strTagName", numValue )
```

Eingaben

strTagName

Der Name des Tags, für den der neue Wert erzwungen wird.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

numValue

Der neue Wert zum Schreiben in den angegebenen Tag.

Rückgabewert Diese Funktion gibt keinen Wert zurück.

Beispiele

Tag-Name	Ausdruck
n/a	ForceTagChange("TagA", 5)

19.20.4 GetTagValue

Ruft den Wert des angegebenem Tags aus der Projektdatenbank ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetTagValue	Tag-Datenbank	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetTagValue( "strTagName", optNumRefresh )
```

Eingaben

strTagName

Der Name des Tags, dessen Wert abgerufen werden soll.

Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

optNumRefresh

Der Tag, der als Auslöser zum Aktualisieren der Funktion verwendet werden soll. Wenn sich der Wert des angegebenen Tags ändert, wird die Funktion erneut ausgeführt.

Normalerweise wird eine Funktion nur dann ausgeführt, wenn das Objekt, auf dem sie konfiguriert ist, sich verändert, z.B. wenn ein Drucktastenobjekt, Seite 193 angeklickt wird. Zum Ausführen der Funktion in regelmäßigen Zeitabständen kann einer der System-Tags, Seite 121 des Projekts wie "Sekunde", "Minute" oder "Stunde" verwendet werden.

Rückgabewert

Diese Funktion gibt nur den Wert des in **strTagName** festgelegten Tags zurück. Wenn der angegeben Tag nicht vorhanden ist, gibt die Funktion den Wert 0 zurück.

Der in **optNumRefresh** festgelegte Tag-Wert beeinflusst nicht den Rückgabewert der Funktion.

Beispiele

Tag-Name	Ausdruck
requiredTag = 15	GetTagName("requiredTag", Second) // Return = 15
pointerTag = "Required Tag" requiredTag = 15 optNum = Second	GetTagName(pointerTag, optNum) // Return = 15

19.20.5 SetTagValue

Legt den Wert des angegebenem Tags in der Projektdatenbank fest.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SetTagValue	Tag-Datenbank	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
SetTagValue( "strTagName", TagValue )
```

Eingaben

strTagName

Der Name des festzulegenden Tags.

Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

TagValue

Der neue Wert zum Setzen im angegebenen Tag.

Rückgabewert

Wert	Beschreibung
−1	Ungültiger Tag-Name
0	Kein Fehler

Beispiele

Tag-Name	Ausdruck
TagA	SetTagValue("TagA", "Hello") // Return = Hello
TagA	SetTagValue("TagA", 123) // Return = 123
TagA, TagB= 15	SetTagValue("TagA", TagB) // Return = 15

19.21 Schleifenfunktionen

19.21.1 Schleifenfunktionen – Allgemeines

Diese Funktionen dienen der Implementierung von sich wiederholenden, inkrementierenden Schleifen in einem Skript.

Loop

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
For, Seite 718 & "Next"	entf.	unterstützt	unterstützt	nicht unterstützt

19.21.2 For … Next

Implementiert eine **For ... Next** Schleife in einem Mathematik-Arbeitsblatt oder einem Dynamikobjekt. Der von der Schleife betroffene Skriptabschnitt beginnt mit dem Funktionsaufruf **For()** und endet mit der Angabe von **Next**. Das Element **Next** führt zurück zum Anfang der Schleife.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
For … Next	Loop	entf.	unterstützt	unterstützt	unterstützt

Syntax

```
For( numInitialValue, numFinalValue, numStep ) … Next
```

Eingaben

numInitialValue

Der erste Schritt (das erste Inkrement) der Schleife.

numFinalValue

Der letzte Schritt (das letzte Inkrement) der Schleife.

numStep

Der Schritt (das Inkrement) der Schleife.

Rückgabewert

Gibt den Schritt, auf dem die Schleife aktuell ausgeführt wird, zurück.

Beispiele

Tag-Name	Ausdruck
Tag	For(1, 5, 1)
Next	

☞ Jede **For**-Funktion muss mit einem **Next**-Element kombiniert werden. Wie im Beispiel gezeigt muss das Element **Next** im Tag-Feld des Mathematik-Arbeitsblatts platziert werden.

19.22 ODBC-Funktionen

19.22.1 ODBC-Funktionen – Allgemeines

Diese Funktionen dienen der Interaktion mit einer externen Datenbank über Open Database Connectivity (ODBC).

☞ Diese Funktionen werden zur Unterstützung von Datenbankschnittstellen aus älteren System zur Verfügung gestellt. In den meisten Fällen empfehlen wir, die neuere Datenbank-/ERP-Verbindungsverwaltung, Seite371 und die entsprechenden Funktionen, Seite 772 zu verwenden.

ODBC

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
ODBCBeginTrans, Seite 720	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCBindCol, Seite 721	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCCanAppend, Seite 722	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCCanTransact, Seite 722	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCCanUpdate, Seite 723	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCClose, Seite 723	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCCommitTrans, Seite 724	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCDelete, Seite 724	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCExecuteSQL, Seite 725	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCInsert, Seite 725	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCIsBOF, Seite 726	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCIsDeleted, Seite 727	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCIsEOF, Seite 727	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCIsFieldNULL, Seite 728	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Anhang: WinStudio-interne Skriptsprache

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
ODBCIsFieldNullable, Seite 728	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCMove, Seite 729	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCMoveFirst, Seite 729	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCMoveLast, Seite 730	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCMoveNext, Seite 730	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCMovePrev, Seite 731	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCOpen, Seite 732	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCQuery, Seite 733	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCRollback, Seite 733	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCSetFieldNULL, Seite 734	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCSetFilter, Seite 734	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCSetSort, Seite 735	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCUnbindCol, Seite 736	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt
ODBCUpdate, Seite 736	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

19.22.2 ODBCBeginTrans

Beginnt eine Transaktion mit der verbundenen Datenquelle.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCBeginTrans	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCBeginTrans(numHandler)
```

Eingaben

numHandler
Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler
2	Datenbank nicht geöffnet
3	Fehler beim Beginn der Transaktion

Beispiele

Tag-Name	Ausdruck
Tag	ODBCBeginTrans(5)

19.22.3 ODBCBindCol

Verbindet eine Spalte mit einem Tag.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCBindCol	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCBindCol( numHandler, strColName, strColType, strTagName )
```

Eingaben

numHandler

Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

strColName

Der Spaltenname in der Datenbank.

strColType

Der SQL-Datentyp (aus der folgenden Liste):

- SQL_BIT
- SQL_TINYINT
- SQL_LONGVARCHAR
- SQL_CHAR
- SQL_VARCHAR
- SQL_DECIMAL
- SQL_NUMERIC
- SQL_DATE
- SQL_TIME
- SQL_TIMESTAMP
- SQL_DOUBLE
- SQL_REAL
- SQL_SMALLINT
- SQL_INTEGER

strTagName

Der Name des mit der Spalte zu verbindenden Tags.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler
2	Ungültiger Parametertyp
3	In einem der Parameter befindet sich eine leere Zeichenkette
4	ColType enthält einen ungültigen Typ

☞ Nach jedem Verbinden von Spalten muss die Funktion ODBCQuery, Seite 733 aufgerufen werden.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCBindCol(5, "OrderDate", "SQL_DATE", "Order_Date")

Weitere Informationen unter: ODBCUnbindCol(), Seite 736

19.22.4 ODBCCanAppend

Prüft, ob der Datenbank neue Datensätze hinzugefügt werden dürfen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCCanAppend	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCCanAppend(numHandler)
```

Eingaben

numHandler

Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Datenbank erlaubt kein Hinzufügen neuer Datensätze
Nicht null	Datenbank erlaubt Hinzufügen neuer Datensätze

Beispiele

Tag-Name	Ausdruck
Tag	ODBCCanAppend(5)

19.22.5 ODBCCanTransact

Prüft, ob die Datenbank Transaktionen zulässt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Avg	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCCanTransact(numHandler)
```

Eingaben

numHandler
Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Datenbank lässt keine Transaktionen zu.
Nicht null	Datenbank lässt Transaktionen zu.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCCanTransact(2)

19.22.6 ODBCCanUpdate

Prüft, ob die Datenbank aktualisiert werden kann.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCCanUpdate	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCCanUpdate(numHandler)
```

Eingaben

numHandler
Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Datenbank lässt keine Aktualisierungen zu.
Nicht null	Datenbank lässt Aktualisierungen zu.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCCanUpdate(6)

19.22.7 ODBCClose

Schließt eine Verbindung zur Datenbank.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCClose	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCClose(numHandler)
```

Eingaben

numHandler
Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler

Beispiele

Tag-Name	Ausdruck
Tag	ODBCClose(5)

19.22.8 ODBCCommitTrans

Schließt eine Transaktion ab. Rufen Sie diese Funktion nach dem Abschluss von Transaktionen auf.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCCommitTrans	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCCommitTrans(numHandler)
```

Eingaben

numHandler

Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler
2	Datenbank nicht geöffnet
3	Fehler beim Abschließen der Transaktion

Beispiele

Tag-Name	Ausdruck
Tag	ODBCCommitTrans(1)

19.22.9 ODBCDelete

Löscht den aktuellen Datensatz.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCDelete	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCDelete(numHandler)
```

Eingaben

numHandler

Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler
2	Datenbank nicht geöffnet
3	Fehler beim Löschen



Nach erfolgreichem Löschen muss eine der „Move“-Funktionen ausdrücklich aufgerufen werden (d.h., ODBCMove, Seite 729, ODBCMoveFirst, Seite 729, ODBCMoveLast, Seite 730, ODBCMoveNext, Seite 730, ODBCMovePrev, Seite 731), um den gelöschten Datensatz zu verlassen.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCDelete(5)

19.22.10 ODBCExecuteSQL

Führt eine SQL-Anweisung direkt aus.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCExecu-teSQL	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausge-führt

Syntax

```
ODBCExecuteSQL( numHandler, strSqlCommand )
```

Eingaben

numHandler

Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

strSqlCommand

Eine gültige SQL-Anweisungen.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler
2	Datenbank nicht geöffnet
3	Ungültiger Parameter
4	Fehler beim Ausführen des SQL-Befehls



Diese Funktion gibt keine Datensätze zurück, ungeachtet der ausgeführten Anweisung.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCExecuteSQL(3, '')
Tag	ODBCExecuteSQL(4, '')

19.22.11 ODBCInsert

Fügt einen neuen Datensatz in die Datenbank ein.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCInsert	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausge-führt

Syntax

```
ODBCInsert(numHandler)
```

Eingaben

numHandler
Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler
2	Datenbank nicht geöffnet
3	Fehler beim Einfügen

☞ Diese Funktion verwendet den Wert der durch die Funktion ODBCBindCol, Seite 721 verknüpften Tag zum Erstellen des neuen Datensatzes.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCInsert(7)

19.22.12 ODBCIsBOF

Gibt zurück, ob über den ersten Datensatz der Datensatzreihe hinausgegangen wurde. (Rufen Sie diese Funktion vor dem Blättern durch die Datensätze auf.)

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCIsBOF	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCIsBOF(numHandler)
```

Eingaben

numHandler
Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Datensatz gefunden
nicht null	Datensatzreihe enthält keine Datensätze oder Rückwärtsbewegung, über den ersten Datensatz hinaus

Diese Funktion kann auch zusammen mit der Funktion ODBCIsEOF, Seite 727 verwendet werden, um zu bestimmen, ob die Datensatzreihe Datensätze enthält oder leer ist. Direkt nach Aufruf der Funktion ODBCQuery, Seite 733 und wenn die Datensatzreihe keine Datensätze enthält, gibt **ODBCIsBOF** nicht Null zurück. Wenn eine Datensatzreihe mit mindestens einem Datensatz geöffnet wird, ist der erste Datensatz der aktuelle Datensatz und **ODBCIsBOF** gibt eine Null ("0") zurück. Wenn der erste Datensatz der aktuelle Datensatz ist und ODBCMovePrev, Seite 731 aufgerufen wird, gibt **ODBCIsBOF** anschließend einen Wert ungleich null aus.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCIsBOF(1)

19.22.13 ODBCIsDeleted

Gibt an, ob der aktuelle Datensatz gelöscht wurde.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCIsDeleted	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCIsDeleted(numHandler)
```

Eingaben

numHandler
Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Datensatzreihe ist nicht auf einem gelöschten Datensatz positioniert.
nicht null	Datensatzreihe ist auf einem gelöschten Datensatz positioniert.

Wenn Sie zu einem Datensatz gehen und diese Funktion nicht Null ausgibt, müssen Sie zu einem anderen Datensatz wechseln, bevor Sie weitere Vorgänge durchführen können.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCIsDeleted(8)

19.22.14 ODBCIsEOF

Gibt an, ob über den letzten Datensatz der Datensatzreihe hinausgegangen wurde. (Rufen Sie diese Funktion beim Blättern durch die Datensätze auf.)

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCIsEOF	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCIsEOF(numHandler)
```

Eingaben

numHandler
Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Datensatz gefunden.
nicht null	Datensatzreihe enthält keine Datensätze oder Vorwärtsbewegung über den letzten Datensatz hinaus.

Diese Funktion kann auch zusammen mit der Funktion ODBCIsBOF, Seite 726 verwendet werden, um zu bestimmen, ob die Datensatzreihe Datensätze enthält oder leer ist. Direkt nach Aufruf der Funktion ODBCQuery, Seite

733 und wenn die Datensatzreihe keine Datensätze enthält, gibt **ODBCIsBOF** nicht Null zurück. Wenn eine Datensatzreihe mit mindestens einem Datensatz geöffnet wird, ist der erste Datensatz der aktuelle Datensatz und **ODBCIsEOF** gibt eine Null ("0") zurück. Wenn der letzte Datensatz der aktuelle Datensatz ist und ODBCMoveNext, Seite 730 aufgerufen wird, gibt **ODBCIsEOF** anschließend einen Wert ungleich null aus.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCIsEOF(5)

19.22.15 ODBCIsFieldNULL

Gibt an, ob ein bestimmtes Feld in einer Datensatzreihe als "NULL" markiert wurde.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCIsFieldNULL	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCIsFieldNULL(numHandler,strColName)
```

Eingaben

numHandler

Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

strColName

Der Spaltenname.

Rückgabewert

Wert	Beschreibung
0	Das angegebene Feld ist nicht als Null markiert.
Nicht null	Das angegebene Feld ist nicht Null markiert.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCIsFieldNULL(7, "CustomerName")
Tag	ODBCIsFieldNULL(3, "CompanyName")

19.22.16 ODBCIsFieldNullable

Prüft, ob ein angegebenes Feld auf einen "NULL"-Wert gesetzt werden kann.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCIsFieldNullable	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCIsFieldNullable(numHandler,strColName)
```

Eingaben

numHandler

Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

strColName

Der Spaltenname.

Rückgabewert

Wert	Beschreibung
0	Das angegebene Feld ist nicht als nullbar markiert.
Nicht null	Das angegebene Feld ist als nullbar markiert.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCIsFieldNullable(1, "Price")
Tag	ODBCIsFieldNullable(1, "Model")

19.22.17 ODBCMove

Bewegt den aktuellen Datensatzzeiger innerhalb einer Datensatzreihe vor oder zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCMove	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCMove(numHandler,numOffset)
```

Eingaben

numHandler

Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

numOffset

Die Anzahl der Zeilen für die Vorwärts- oder Zurückbewegung:

- Positive Werte bewegen den Zeiger vorwärts, zum Ende der Datensatzreihe hin.
- Negative Werte bewegen den Zeiger rückwärts, zum Anfang der Datensatzreihe hin.
- Der Wert "0" aktualisiert den aktuellen Datensatz.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler
2	Datenbank nicht geöffnet
3	Bewegungsfehler

Beispiele

Tag-Name	Ausdruck
Tag	ODBCMove(2, 3)
Tag	ODBCMove(8, 2)

19.22.18 ODBCMoveFirst

Bewegt den Zeiger zum ersten Datensatz in der Datensatzreihe.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCMoveFirst	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCMoveFirst(numHandler)
```

Eingaben

numHandler
Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	erfolgreich
1	Ungültiger Handler.
2	Datenbank nicht geöffnet.
3	Bewegungsfehler.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCMoveFirst(4)

19.22.19 ODBCMoveLast

Bewegt den Zeiger zum letzten Datensatz in der Datensatzreihe.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCMoveLast	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCMoveLast(numHandler)
```

Eingaben

numHandler
Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler
2	Datenbank nicht geöffnet
3	Bewegungsfehler

Beispiele

Tag-Name	Ausdruck
Tag	ODBCMoveLast(7)

19.22.20 ODBCMoveNext

Bewegt den Zeiger zum nächsten Datensatz in der Datensatzreihe.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCMoveNext	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCMoveNext(numHandler)
```

Eingaben

numHandler
Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler
2	Datenbank nicht geöffnet
3	Ende der Datensatzreihe erreicht
4	Bewegungsfehler

Beispiele

Tag-Name	Ausdruck
Tag	ODBCMoveNext(9)

19.22.21 ODBCMovePrev

Bewegt den Zeiger zum nächsten Datensatz in der Datensatzreihe.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCMovePrev	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCMovePrev(numHandler)
```

Eingaben

numHandler
Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler
2	Datenbank nicht geöffnet
3	Anfang der Datensatzreihe erreicht
4	Bewegungsfehler

Beispiele

Tag-Name	Ausdruck
Tag	ODBCMovePrev(2)

Anhang: WinStudio-interne Skriptsprache

19.22.22 ODBCOpen

Öffnet eine Verbindung zur Datenbank und gibt einen numerischen Handler zur Verwendung durch andere ODBC-Funktionen zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCOpen	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCOpen( strDsn, strUser, strPassw, strTable, strFilter, strSort )
```

Eingaben

strDsn

Der Name der Datenquelle.

strUser

Benutzername.

strPassw

Passwort.

strTable

Name der Datenbanktabelle

strFilter

Der SQL-SATZ **WHERE**.

strSort

Der SQL-SATZ **ORDER BY**.

Rückgabewert

Wert	Beschreibung
n	Bei erfolgreicher Ausführung wird der Handler zur Identifizierung der Datenbank zurückgegeben.
−1	Ungültiger Parameter
−2	DSN oder TableName enthalten eine leere Zeichenkette

☞ Diese Funktion liest oder schreibt keine Daten; sie erstellt nur einen Handler zum Handhaben der Datenbank. Die Spalten müssen mithilfe der Funktion ODBCBindCol, Seite 721 gebunden werden und dann muss die Funktion ODBCQuery, Seite 733 zum Abrufen des ersten Datensatzes aufgerufen werden.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCOpen("MyDSNFile", "Alex", "", "Table1", "Name='Mayer'", "Name ASC")
Tag	ODBCOpen("DSNFileName", "Robert", "Robot", "Table1", "", "")

19.22.23 ODBCQuery

Ruft den aktuell ausgewählten Datensatz aus einer Datenbank ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCQuery	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCQuery(numHandler)
```

Eingaben

numHandler
Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler
2	Keine Spalten gebunden
3	Datenbank kann nicht geöffnet werden
4	Datenbank kann nicht gestartet werden
5	Abfragefehler

☞ Wenn die Spaltenbindung, Seite 721 oder die Filter, Seite 734 und die Sortierung, Seite 735 verändert werden, muss diese Funktion erneut aufgerufen werden.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCQuery(6)

19.22.24 ODBCRollBack

Verwrift die während eines Vorgangs vorgenommenen Änderungen

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCRollBack	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCRollBack(numHandler)
```

Eingaben

numHandler
Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler

Wert	Beschreibung
2	Datenbank nicht geöffnet
3	Fehler beim Verwerfen der Änderungen

Beispiele

Tag-Name	Ausdruck
Tag	ODBCRollback(4)

19.22.25 ODBCSetFieldNULL

Markiert ein Felddaten-Member in der Datensatzreihe als "NULL" (spezifisch ohne Wert) oder als "nicht NULL".

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCSetField-NULL	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCSetFieldNULL( numHandler, strColName, numValue )
```

Eingaben

numHandler

Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

strColName

Der Spaltenname.

numValue

Ein numerischer Tag zur Festlegung der Felddaten als NULL wenn 0 und als nicht NULL wenn nicht Null.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler
2	Datenbank nicht geöffnet
3	Ungültiger Parameter
4	Ungültiger Spaltenname

Beispiele

Tag-Name	Ausdruck
Tag	ODBCSetFieldNULL(2, "Price", 1)
Tag	ODBCSetFieldNULL(4, "CompanyName", 0)

19.22.26 ODBCSetFilter

Beschränkt die in einer Datenbank ausgewählten Datensätze.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCSetFilter	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCSetSort(numHandler,strSort)
```

Eingaben

numHandler

Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

strFilter

Der SQL-SATZ **WHERE**.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler
2	Ungültiger Parameter

☞ Diese Funktion kann hilfreich sein für die Auswahl einer untergeordneten Datensatzreihe, z.B. „alle Vertriebsmitarbeiter in Kalifornien“ (**"state = 'CA'"**). Denken Sie daran, nach Aufruf dieser Funktion ODBCQuery, Seite 733 aufzurufen.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCSetFilter(3, "Name='Morgan'")

19.22.27 ODBCSetSort

Sortiert die in einer Datenbank ausgewählten Datensätze.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCSetSort	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCSetSort(numHandler,strSort)
```

Eingaben

numHandler

Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

strSort

Der SQL-SATZ **ORDER BY**.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler
2	Ungültiger Parametertyp

☞ Diese Eigenschaft kann zum Sortieren von Datensätzen in einer oder mehreren Spalten verwendet werden. Denken Sie daran, nach Aufruf dieser Funktion ODBCQuery, Seite 733 aufzurufen.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCSetSort(5, "Name DESC")

19.22.28 ODBCUnbindCol

Löst die Bindung einer Spalte, die vorher durch die Funktion "ODBCBindCol" verknüpft wurde, auf.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCUnbindCol	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCUnbindCol(numHandler,strColName)
```

Eingaben

numHandler

Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

strColName

Der Spaltenname.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler
2	Ungültiger Parametertyp
3	Spalte nicht gebunden

Beispiele

Tag-Name	Ausdruck
Tag	ODBCUnbindCol(7, "Name")

19.22.29 ODBCUpdate

Aktualisiert den aktuellen Datensatz.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
ODBCUpdate	ODBC	synchron	unterstützt	nicht unterstützt	auf dem Server ausgeführt

Syntax

```
ODBCUpdate(numHandler)
```

Eingaben

numHandler

Der von der Funktion ODBCOpen, Seite 732 zurückgegebene Handler.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiger Handler

Wert	Beschreibung
2	Datenbank nicht geöffnet
3	Aktualisierungsfehler

☞ Diese Funktion verwendet die Werte der durch die Funktion ODBCBindCol, Seite 721 verknüpften Tags zum Aktualisieren des aktuellen Datensatzes.

Beispiele

Tag-Name	Ausdruck
Tag	ODBCUpdate(1)

19.23 E-Mail-Funktionen

19.23.1 E-Mail-Funktionen – Allgemeines

Diese Funktionen dienen dem Konfigurieren und Senden von E-Mails aus einem Projekt heraus.

E-Mail

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
CnfEmail, Seite 737	synchron	unterstützt	unterstützt	unterstützt
GetStatusSendEmailExt, Seite 740	synchron	unterstützt	unterstützt	unterstützt
SendEmail, Seite 740	synchron	unterstützt	unterstützt	unterstützt
SendEmailExt, Seite 741	asynchron	unterstützt	unterstützt	unterstützt

19.23.2 CnfEmail

Diese Funktion konfiguriert die von anderen Funktionen im Projekt verwendeten E-Mail-Einstellungen zum Senden von E-Mails, wie z.B. Alarmarbeitsblätter und die Funktionen **SendEmail** und **SendEmailExt**.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
CnfEmail	E-Mail	synchron	unterstützt	unterstützt, außer für Verschlüsselung	unterstützt

Syntax

```
CnfEmail(strSMTP, strFrom, strPOP3, strUser, strPassword, optNumTimeout, optNumAuthType, optStrSMTPUser,
        optStrSMTPPassword)

CnfEmail(strSMTP,strFrom,strPOP3,strUser,strPassword{ | ,optNumTimeout{ | ,{ optNumAuthType | 1 | 2 }
        { | ,optStrSMTPUser,optStrSMTPPassword } } })
```

Eingaben

strSMTP

Host-Name oder IP-Adresse des ausgehenden E-Mail-Servers, auch als SMTP-Server bezeichnet. Wenn der Server keine der Standard-SMTP-Ports verwendet, kann eine Port-Nummer angegeben werden.

☞ Auf Windows-CE-Geräten muss eine IP-Adresse angegeben werden.

strFrom

Die E-Mail-Adresse, von der die E-Mails gesendet werden und auf der E-Mails empfangen werden können. Dies sollte eine gültige Adresse auf dem POP3-Server sein (siehe **strPOP3** unten).

strPOP3

Host-Name oder IP-Adresse des eingehenden E-Mail-Servers, auch als POP3-Server bezeichnet. Wenn der Server keine der Standard-POP3-Ports verwendet, kann eine Port-Nummer angegeben werden.

☞ Auf Windows-CE-Geräten muss eine IP-Adresse angegeben werden.

strUser

Der Benutzername zum Anmelden am POP3-Server.

strPassword

Das Passwort zum Anmelden am POP3-Server.

optNumTimeout

Die Timeout-Begrenzung (in Sekunden) für das Senden von E-Mails. Wenn vom SMTP-Server innerhalb dieser Zeitspanne keine Antwort kommt, wird der Vorgang abgebrochen.

Dieser Parameter ist optional; wird keine Timeout-Begrenzung angegeben, versucht das Projekt solange zu senden, bis es eine Antwort erhält. Es sollte allerdings eine Timeout-Begrenzung angegeben werden, damit das Projekt sich nicht aufhängen kann.

optNumAuthType

optStrSMTPUser

optStrSMTPPassword

Standardmäßig benötigen SMTP-Server keine Authentifizierung für ausgehende E-Mails. Wenn Ihr Server eine Authentifizierung benötigt, setzen Sie **optNumAuthType** auf "1" (unverschlüsselt) oder "2" (über TLS/SSL verschlüsselt) und geben dann Benutzername und Passwort an. (Wenn Ihr Benutzername und Passwort für SMTP und POP3 gleich sind, können **optStrSMTPUser** und **optStrSMTPPassword** ausgelassen werden.) Das Projekt verwendet dann automatisch die Werte aus **strUser** und **strPassword**.)

☞ Verschlüsselung mittels TLS/SSL ist nicht für Projekte verfügbar, die auf Windows-CE-Zielsystemen laufen.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiges Format für strSMTP
2	Ungültiges Format für strFrom
3	Ungültiges Format für strPOP3
4	Ungültiges Format für strUser
5	Ungültiges Format für strPassword
6	Ungültiges Format für optNumTimeout
7	Falsche Anzahl Parameter
8	Fehler beim Abruf der Host-IP-Adresse (ungültiger POP3-Server)
9	Fehler beim Verbinden mit dem POP3-Server
10	Fehler beim Senden des Benutzernamens
11	Fehler beim Senden des Passworts
12	Gewählter Authentifizierungsmodus wird vom SMTP-Server nicht unterstützt
13	Ungültiger SMTP-Benutzername
14	Authentifizierung fehlgeschlagen



Die von dieser Funktion erstellte E-Mail-Konfiguration funktioniert nur in dem Windows-Prozess, in dem die Funktion aufgerufen wurde.

Wenn zum Beispiel ein Schaltflächenobjekt in einem Bild platziert und das Objekt dann zum Aufruf dieser Funktion beim Drücken eingestellt wird, funktioniert die resultierende E-Mail-Konfiguration nur auf der Client Station, auf der das Bild angezeigt und die Schaltfläche gedrückt wird.

Sie funktioniert weder auf anderen Client-Stationen noch auf der Serverstation, weil der auf der Client-Station ausgeführte Projekt-Viewer nur mit dem auf der Serverstation ausgeführten Datenserver Daten austauscht (d.h. Änderungen vonTag-Werten). Ein Server kann nicht direkt Funktionen auf einem anderen aufrufen; es können nur Trigger verwendet werden, die den Aufruf von Funktionen auf dem anderen Server auslösen.

Es ist zu beachten, dass dies auch gilt, wenn Client-Station und Server-Station sich auf dem gleichen physischen Gerät befinden, weil Projekt-Viewer und Datenserver unter Windows zwei separate Prozesse sind.

Wenn eine E-Mail-Konfiguration für die Hintergrund-Tasks Ihres Projekts gelten soll – z.B. zum Senden von E-Mails, wenn Alarme aktiv werden – muss entweder der Dialog "E-Mail-Einstellungen" zum Konfigurieren der Standardeinstellungen für das gesamte Projekt verwendet werden ODER diese Funktion an einer Stelle wie z.B. dem Startup-Skript, Seite 369, einer Skriptgruppe, Seite 368 oder einem Berechnungsarbeitsblatt, Seite 367 aufgerufen werden.

Beispiele *CnfEmail - Beispiele*

```
CnfEmail("smtp.company.com","Robert@company.com","pop.company.com","RobertH","Shades556",100)

CnfEmail("smtp.company.com:4455","Robert@company.com","pop.company.com:9900","RobertH","Shades556",5,1)

CnfEmail("195.11.22.33:4455","Robert@company.com","195.66.77.88:9900","RobertH","Shades556",5,2,"JohnS",
        "abcd1234")
```

19.23.3 GetStatusSendEmailExt

Gibt den Status der mit der Funktion "SendEmailExt" zuletzt gesendeten E-Mail zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
GetStatusSendE-mailExt	E-Mail	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
GetStatusSendEmailExt({ | optTagName })
```

Eingaben

optTagName
"Optionaler" Tag, der bewirkt, dass die Funktion Ihren Rückgabewert aktualisiert. Dieser Parameter ist optional, muss aber konfiguriert werden, wenn diese Funktion für eine Objektanimation, Seite 243 (z.B. Textdatenverknüpfung, Position) konfiguriert wird.

Rückgabewert

Wert	Beschreibung
−2	Falsche Version der Bibliothek **INDMail.DLL**.
−1	Die Bibliothek **INDMail.DLL** ist korrupt.
0	Die Funktion **SendEmailExt** wird nicht ausgeführt.
1	Senden der letzten E-Mail dauert noch an. Funktion **SendEmailExt** kann nicht ausgeführt werden.
2	Letzte E-Mail wurde erfolgreich versendet. Funktion **SendEmailExt** kann wieder ausgeführt werden.
3	Fehler beim Senden der letzten E-Mail. Funktion **SendEmailExt** erneut ausführen.

Beispiele

Tag-Name	Ausdruck
Tag	GetStatusSendEmailExt(Second)
Tag	GetStatusSendEmailExt()

19.23.4 SendEmail

Diese Funktion sendet eine E-Mail-Nachricht an einen Empfänger (ohne Dateianhang).

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SendEmail	E-Mail	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
SendEmail( strSubject, strMessage, strTo )
```

Eingaben

strSubject

Der Betreff der E-Mail.

strMessage

Nachrichtentext der E-Mail, bis zu 255 Zeichen lang.

strTo

Die E-Mail-Adresse des Empfängers.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ungültiges Format für strSubject
2	Ungültiges Format für strMessage
3	Ungültiges Format für strTo
4	Falsche Anzahl Parameter
5	Fehler Start-Socket
6	Fehler beim Abruf der Host-IP-Adresse (ungültiger SMTP-Server)
7	Fehler beim Verbinden mit dem SMTP-Server
8	Fehler beim Senden des HELO-Befehls (d.h. Initialisierung)
9	Fehler beim Senden des MAIL-Befehls (d.h. Adresse „Von“)
10	Fehler beim Senden des RCPT-Befehls (d.h. Adresse „An“)
11	Fehler beim Senden von DATA (d.h. Nachrichtentext)
12	Fehler beim Senden des SMTP-Authentifizierungsbefehls
13	Ungültiger Benutzername
14	Ungültiges Passwort

☞ Vor dem Versenden der E-Mail müssen die E-Mail-Einstellungen des Projekts konfiguriert werden, entweder durch Verwendung des Dialogs **E-Mail-Einstellungen** oder durch Aufruf der Funktion **CnfEmail**. Fehlerhafte Konfigurationen werden im Rückgabewert (siehe oben) angezeigt..

Die Funktion **SendEmail** kann nicht verwendet werden, wenn Verschlüsselung über TLS/SSL ausgewählt ist oder eine E-Mail mit Unicode-Zeichen versendet wird. Dafür steht die Funktion **SendEmailExt** zur Verfügung.

Beispiele

SendEmail - Beispiele

```
SendEmail( "Hi!", "How are you?", "rogers@pnd.net" )

SendEmail( statusSummary, statusDetail, adminAddress )
```

19.23.5 SendEmailExt

Versendet eine E-Mail-Nachricht mit Dateianhängen.

Anhang: WinStudio-interne Skriptsprache

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SendEmailExt	E-Mail	asynchron	unterstützt	unterstützt	unterstützt

Syntax

```
SendEMailExt( strSubject, strMessage, strTo, optStrCc, optStrBcc, optStrFile1, … , optStrFileN )
```

Eingaben

strSubject

Der Betreff der E-Mail (bis zu 255 Zeichen).

strMessage

Die E-Mail-Nachricht (bis zu 255 Zeichen).

strTo

Die Adresse des Empfängers. Es kann mehr als ein Empfänger angegeben werden; die Adressen werden durch ein Semikolon (";") getrennt.

optStrCc

Die Empfängeradressen für Kopien der Nachricht. Es kann mehr als ein Empfänger angegeben werden; die Adressen werden durch ein Semikolon (";") getrennt.

Dieser Parameter ist optional. Wenn nachfolgende Parameter verwendet werden sollen, kann hier ein Null-String ("""") angegeben werden.

optStrBcc

Die Empfängeradressen für "Blindkopien" der Nachricht. Es kann mehr als ein Empfänger angegeben werden; die Adressen werden durch ein Semikolon (";") getrennt.

Dieser Parameter ist optional. Wenn nachfolgende Parameter verwendet werden sollen, kann hier ein Null-String ("""") angegeben werden.

optStrFile (1-N)

Vollständige Dateipfade und -namen für Dateianhänge.

Rückgabewert

Wert	Beschreibung
−4	Einige der Dateianhänge wurden nicht gefunden.
−3	Falsche Parameteranzahl (mindesten drei Parameter sind notwendig).
−2	Falsche Version der Bibliothek **INDMail.DLL** .
−1	Die Bibliothek **INDMail.DLL** ist korrupt.
0	Erfolgreich
1	Funktion kann nicht ausgeführt werden, weil die letzte E-Mail noch nicht versendet wurde.
2	Interner Fehler

Vor dem Senden von E-Mails muss zunächst die Funktion CnfEmail, Seite 737 zum Konfigurieren der E-Mail-Einstellungen verwendet werden. Fehlerhafte Konfigurationen werden im Rückgabewert (siehe oben) angezeigt.

Beispiele

Tag-Name	Ausdruck
Tag	SendEmailExt("Subject", "Message", "Sam@universe.com", "", "", "C:\Projects eport.txt")
Tag	SendEmailExt("Subject", "Message", "David@Ohio.net", "Ted@Austin.com", "Bart@Springfield.gov", "C:\TechRef51.doc")

19.24 Einwahlfunktionen

19.24.1 Einwahlfunktionen – Allgemeines

Diese Funktionen dienen dem Konfigurieren eines Computermodems (falls vorhanden) und dem Aufbau von Einwahlverbindungen zu anderen Rechnern.

Diese Funktionen werden unter Windows 7 nicht unterstützt.

Einwahl

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
DialError, Seite 744	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt
DialGetClientIP, Seite 750	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt
DialGetServerIP, Seite 751	synchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt
DialStatus, Seite 753	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt
DialUp, Seite 755	asynchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt
DialUpToCE, Seite 756	asynchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt
FindAllDevices, Seite 758	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt
FindModem, Seite 758	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt
HangUp, Seite 759	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt
PhoneDialUp, Seite 759	asynchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt
PhoneDisableListen, Seite 760	synchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt

Anhang: WinStudio-interne Skriptsprache

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
PhoneEnableListen, Seite 761	synchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt
PhoneHangUp, Seite 761	asynchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt
PhoneStatus, Seite 762	synchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt

19.24.2 DialError

Gibt die Fehlermeldungen zu jeder Verbindung zurück.

☞ Diese Funktion wird unter Windows 7 nicht unterstützt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DialError	Einwahl	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt

Syntax

```
DialError(numType, strPhonebookEntryOrModem, "optError", optRefresh)
```

Eingaben

numType

Numerischer Parameter, der den Inhalt für den Parameter **strPhonebookEntryorModem** festlegt.

- 0: Name Telefonbuch
- 1: Name Modem
- 2: Name Direktverbindung

strPhonebookEntryOrModem

Name des Telefonbuchs, des Modems oder der Direktverbindung zum Herstellen der Verbindung. Der Parameter **numType** legt fest, welche der Methoden verwendet wird.

optError

"Optional" Der Name des String-Tags, der die Fehlermeldung erhält.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

optRefresh

"Optionaler" Tag, der bewirkt, dass die Funktion Ihren Rückgabewert aktualisiert. Dieser Parameter ist optional, muss aber konfiguriert werden, wenn diese Funktion in einer Objektanimation, Seite 243 (z.B. Textdatenverknüpfung, Position) verwendet wird.

Rückgabewert

Wert	Beschreibung
0	OK
−1	Fehler: INDRas.DLL nicht gefunden.
−2	Fehler: INDRas.DLL beschädigt.
−3	Fehler: ungültige Anzahl Parameter (mindestens 2).
-4	Ungültiger Wert für den Parameter **numType** (0 oder 1).
-5	Telefonbuch oder Modem nicht vorhanden.
600	Ein Vorgang steht noch aus.
601	Port-Handle ist ungültig.
602	Port ist bereits geöffnet.
603	Puffer des Anrufers zu klein.
604	Falsche Informationen angegeben.
605	Port-Information kann nicht gesetzt werden.
606	Port ist nicht verbunden
607	Ereignis ist ungültig.
608	Gerät nicht vorhanden.
609	Gerätetyp nicht vorhanden.
610	Puffer ist ungültig.
611	Die Route ist nicht erreichbar.
612	Die Route ist nicht zugewiesen.
613	Ungültige Komprimierung angegeben.
614	Keine Puffer verfügbar.
615	Port wurde nicht gefunden.
616	Asynchrone Anfrage steht noch an.
617	Port oder Gerät wird bereits getrennt.
618	Port ist nicht geöffnet.
619	Port ist getrennt.
620	Keine Endpunkte vorhanden.
621	Telefonbuchdatei kann nicht geöffnet werden.
622	Telefonbuchdatei kann nicht geladen werden.
623	Telefonbuchdatei kann nicht geöffnet werden.
624	Telefonbuchdatei kann nicht geschrieben werden.
625	Ungültige Informationen in Telefonbuchdatei gefunden.
626	Zeichenkette kann nicht geladen werden.
627	Schlüssel kann nicht gefunden werden.
628	Port wurde getrennt.
629	Datenverbindung wurde vom Remote-Gerät getrennt.

Anhang: WinStudio-interne Skriptsprache

Wert	Beschreibung
630	Port wurde aufgrund eines Hardwarefehlers getrennt.
631	Port wurde vom Benutzer getrennt.
632	Falsche Strukturgröße.
633	Der Port wird bereits verwendet oder ist nicht für Remote-Zugriff-Auswahl konfiguriert.
634	Ihr Rechner kann auf dem Remote-Netzwerk nicht registriert werden.
635	Unbekannter Fehler.
636	Falsches Gerät am Port angeschlossen.
637	Die Zeichenkette konnte nicht konvertiert werden.
638	Zeitüberschreitung bei Anfrage.
639	Kein asynchrones Netzwerk verfügbar.
640	Ein NetBIOS-Fehler ist aufgetreten.
641	Der Server kann keine NetBIOS-Ressourcen zur Unterstützung des Clients zuweisen.
642	Einer Ihrer NetBIOS-Namen ist bereits auf dem Remote-Netzwerk registriert.
643	Ein Netzwerkadapter auf dem Server ist ausgefallen.
644	Sie werden keine Pop-Up-Netzwerkmeldungen erhalten.
645	Interner Authentifizierungsfehler.
646	Dieses Konto hat zu dieser Uhrzeit keine Anmeldeberechtigung.
647	Das Konto ist deaktiviert.
648	Das Passwort ist abgelaufen.
649	Das Konto verfügt über keine Berechtigung für Remote-Zugriff.
650	Der Remote-Zugriff-Server antwortet nicht.
651	Ihr Modem (oder anderes Verbindungsgerät) meldet einen Fehler.
652	Unbekannte Antwort vom Gerät.
653	Ein vom Gerät benötigtes Makro im **.INF**-Dateibereich des Geräts nicht gefunden.
654	Ein Befehl oder eine Antwort im **.INF**-Dateiabschnitt des Geräts bezieht sich auf ein nicht definiertes Makro.
655	Das Makro <message wurde im **.INF**-Dateiabschnitt des Geräts nicht gefunden.
656	Das Makro <deafultoff im **.INF**-Dateiabschnitt des Geräts enthält ein nicht definiertes Makro.
657	Die **.INF**-Datei des Geräts konnte nicht geöffnet werden.
658	Der Gerätename der **.INF**-Date des Geräts bzw. **.INI**-Datei des Mediums ist zu lang.
659	Die **.INF**-Datei des Mediums bezieht sich auf einen unbekannten Gerätenamen.

Wert	Beschreibung
660	Die **.INF**-Datei des Geräts enthält keine Antworten für den Befehl.
661	In der **.INF**-Datei des Geräts fehlt ein Befehl.
662	Es wurde versucht, ein Makro zu setzen, das nicht im **.INF**-Dateibereich des Geräts aufgeführt ist.
663	Die **.INI**-Datei des Mediums bezieht sich auf einen unbekannten Gerätetyp.
664	Speicher kann nicht zugewiesen werden.
665	Der Port ist für Remote-Zugriff nicht konfiguriert.
666	Ihr Modem (bzw. anderes Verbindungsgerät) funktioniert nicht.
667	**.INI**-Datei des Mediums kann nicht gelesen werden.
668	Verbindung wurde unterbrochen.
669	Der Verwendungsparameter in der **.INI**-Datei des Mediums ist ungültig.
670	Abschnittsname kann nicht aus der **.INI**-Datei des Mediums gelesen werden.
671	Gerätetyp kann nicht aus der **.INI**-Datei des Mediums gelesen werden.
672	Gerätename kann nicht aus der **.INI**-Datei des Mediums gelesen werden.
673	Verwendung kann nicht aus der **.INI**-Datei des Mediums gelesen werden.
674	Maximale BPS-Verbindungsgeschwindigkeit kann nicht aus der **.INI**-Datei des Mediums gelesen werden.
675	Maximale BPS-Übertragungsgeschwindigkeit kann nicht aus der **.INI**-Datei des Mediums gelesen werden.
676	Leitung besetzt.
677	Anruf wurde von einer Person anstelle eines Modems angenommen.
678	Keine Antwort.
679	Kein Träger gefunden.
680	Kein Wählton.
681	Allgemeiner Fehler vom Gerät gemeldet.
682	ERROR_WRITING_SECTIONNAME
683	ERROR_WRITING_DEVICETYPE
684	ERROR_WRITING_DEVICENAME
685	ERROR_WRITING_MAXCONNECTBPS
686	ERROR_WRITING_MAXCARRIERBPS
687	ERROR_WRITING_USAGE
688	ERROR_WRITING_DEFAULTOFF
689	ERROR_READING_DEFAULTOFF
690	ERROR_EMPTY_INI_FILE

Anhang: WinStudio-interne Skriptsprache

Wert	Beschreibung
691	Zugriff verweigert weil Benutzername und/oder Passwort in der Domäne ungültig sind.
692	Hardwareausfall am Port oder angeschlossenen Gerät.
693	ERROR_NOT_BINARY_MACRO
694	ERROR_DCB_NOT_FOUND
695	ERROR_STATE_MACHINES_NOT_STARTED
696	ERROR_STATE_MACHINES_ALREADY_STARTED
697	ERROR_PARTIAL_RESPONSE_LOOPING
698	Ein Antwort-Schlüsselname in der **.INF**-Datei des Geräts hat ein unerwartetes Format.
699	Pufferüberlauf durch Geräteantwort.
700	Der erweiterte Befehl in der **.INF**-Datei des Geräts ist zu lang.
701	Das Gerät hat auf eine BPS-Geschwindigkeit umgestellt, die vom COM-Treiber nicht unterstützt wird.
702	Unerwartete Geräteantwort erhalten.
703	Die Anwendung erlaubt keinen Benutzereingriff, die Verbindung erfordert Benutzereingriff zum erfolgreichen Ausführen
704	ERROR_BAD_CALLBACK_NUMBER
705	ERROR_INVALID_AUTH_STATE
706	ERROR_WRITING_INITBPS
707	Diagnostikanzeige X.25.
708	Das Konto ist abgelaufen.
709	Fehler bei Passwortänderung in der Domäne. Das Passwort ist evtl. zu kurz oder zu lang oder stimmt mit bereits verwendeten Passwörtern überein.
710	Bei der Kommunikation mit Ihrem Modem wurden serielle Überlauffehler gefunden.
711	RasMan-Initialisierungsfehler. Ereignisprotokoll prüfen.
712	Initialisierung Biplex-Port. Einige Sekunden warten und erneut einwählen.
713	Keine aktiven ISDN-Leitungen verfügbar.
714	Keine ISDN-Kanäle für den Anruf verfügbar.
715	Zu viele Fehler wegen schlechter Telefonleitungsqualität aufgetreten.
716	Die Remote-Access-IP-Konfiguration ist nicht verwendbar.
717	Keine IP-Adressen verfügbar im statischen Pool der Remote-Access-IP-Adressen.
718	Zeitüberschreitung beim Warten auf eine gültige Antwort vom Remote-PPP-Peer.
719	PPP vom Remote-Gerät beendet.

Wert	Beschreibung
720	Keine PPP-Steuerprotokolle konfiguriert.
721	Remote-PPP-Peer antwortet nicht.
722	PPP-Paket ist ungültig.
723	Die Telefonnummer mit Vorwahl und Durchwahl ist zu lang.
724	Das IPX-Protokoll kann auf dem Port nicht herauswählen, weil das Gerät ein IPX-Router ist.
725	Das IPX-Protokoll kann auf dem Port nicht einwählen, weil der IPX-Router nicht installiert ist
726	Das IPX-Protokoll kann nicht zum Auswählen auf mehr als einem Port verwendet werden.
727	Kein Zugriff auf **TCPCFG.DLL**.
728	An Remote-Zugriff angebundener IP-Adapter kann nicht gefunden werden.
729	SLIP kann nur verwendet werden, wenn das IP-Protokoll installiert wurde.
730	Computerregistrierung unvollständig.
731	Das Protokoll ist nicht konfiguriert.
732	Die PPP-Verhandlung konvergiert nicht.
733	Das PPP-Steuerprotokoll für dieses Netzwerkprotokoll ist auf dem Server nicht verfügbar.
734	Das PPP-Verknüpfungs-Steuerprotokoll wurde beendet.
735	Die angeforderte Adresse wurde vom Server abgelehnt.
736	Der Remote-Rechner hat das Steuerungsprotokoll beendet.
737	Loopback gefunden.
738	Der Server hat keine Adresse zugewiesen.
739	Das vom Remote-Server benötigte Authentifizierungsprotokoll kann die verschlüsselte Passwort-Wiedereinwahl von Windows NT mit ausdrücklicher Passworteingabe nicht verwenden.
740	Ungültige TAPI-Konfiguration.
741	Der lokale Rechner unterstützt den erforderlichen Verschlüsselungstyp nicht.
742	Der Remote-Rechner unterstützt den erforderlichen Verschlüsselungstyp nicht.
743	Der Remote-Rechner erfordert Verschlüsselung.
744	Vom Remote-Server zugewiesene IPX-Netzwerknummer kann nicht verwendet werden. Ereignisprotokoll prüfen.
745	ERROR_INVALID_SMM
746	ERROR_SMM_UNINITIALIZED
747	ERROR_NO_MAC_FOR_PORT
748	ERROR_SMM_TIMEOUT

Wert	Beschreibung
749	ERROR_BAD_PHONE_NUMBER
750	ERROR_WRONG_MODULE
751	Ungültige Rückrufnummer. In der Nummer sind nur die Zeichen "0" bis "9", "T", "P", "W", "(", ")", "-", "@" und Leerzeichen erlaubt.
752	Bei der Verarbeitung eines Skripts ist ein Syntaxfehler aufgetreten.
753	Die Verbindung konnte nicht getrennt werden, weil sie vom Multi-Protocol Router erstellt wurde.

Beispiele

Tag-Name	Ausdruck
Tag	DialError(0, "Office DialUp")
Tag	DialError(1, "USRobotics_SportsterFaxModem", "StatusMessage", second)
Tag	DialError(2, "DirectDial", "DialupError")

Weitere Informationen unter: FindModem(), Seite 758

19.24.3 DialGetClientIP

Ruft die IP-Adresse einer Remote Access Service (RAS) Client-Station ab.

☞ Diese Funktion wird unter Windows 7 nicht unterstützt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DialGetClientIP	Einwahl	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt

Syntax

```
DialGetClientIP({ numType | 0 | 1 | 2 },strPhoneBookOrModem,"tagClientIP"{ | ,optTagRefresh })
```

Eingaben

numType

Die Art der von **strPhoneBookOrModem** angegebenen Information: "0" ist ein Telefonbuchname, "1" ist ein Modemname und "2" ist ein Direktverbindungsname.

strPhoneBookOrModem

Name des Telefonbuchs, des Modems oder der Direktverbindung zum Herstellen der Verbindung.

tagClientIP

Der Name eines String-Tags, der die IP-Adresse erhält.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

optTagRefresh

Der Name eines Tags, der bei jeder Wertänderung des Tags die Funktion zum Aktualisieren des Rückgabewerts auslöst.

Dieser Parameter optional. Er muss aber angegeben werden, wenn die Funktion in einer Objektanimation wie z.B. eine Text-Daten-Verknüpfung oder eine Position verwendet wird.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte zurück:

Wert	Beschreibung
-5	**DialGetClientIP** wurde in **IndRAS.dll** nicht gefunden.
-4	Ungültiger Typ, **numType** prüfen.
-3	Falsche Anzahl Parameter.
-2	**DialStatus** wurde in **IndRAS.dll** nicht gefunden.
-1	**IndRAS.dll** wurde nicht geladen.
n	Von **DialStatus** zurückgegebener Statuscode.

Die IP-Adresse wird in den von **tagClientIP** festgelegten Tag geschrieben.

☞ Diese Funktion muss auf der RAS-Client-Station ausgeführt werden und **nicht** auf der Serverstation. (Im Prinzip erhält die Station ihre eigene IP-Adresse.)

Beispiele

Verwenden des Telefonbuchnamens "Office DialUp" und Schreiben der resultierenden IP-Adresse in **ClientIPTag**:

DialGetClientIP(0, "Office DialUp", "ClientIPTag")

Verwenden des Modemnamens "USRobotics_SportsterFaxModem" zum Schreiben der resultierenden IP-Adresse in **ClientIPAddress** und Aktualisieren zu jeder Sekunde (d.h. jedes Mal, wenn sich der System-Tag **Second** ändert:

DialGetClientIP(1, "USRobotics_SportsterFaxModem", "ClientIPAddress", Second)

Verwenden des Direktverbindungsnamens "DirectDial" und Schreiben der resultierenden IP-Adresse in **IPAdd**:

DialGetClientIP(2, "DirectDial", "IPAdd")

19.24.4 DialGetServerIP

"DialGetServerIP" ist eine interne Skript-Funktion zum Abrufen der IP-Adresse einer Remote Access Service (RAS) Serverstation.

☞ Diese Funktion wird unter Windows 7 nicht unterstützt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DialGetServerIP	Einwahl	synchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt

Syntax

```
DialGetServerIP({ numType | 0 | 1 | 2 },strPhoneBookOrModem,"tagServerIP"{ | ,optTagRefresh })
```

Eingaben

numType

Die Art der von **strPhoneBookOrModem** angegebenen Information: "0" ist ein Telefonbuchname, "1" ist ein Modemname und "2" ist ein Direktverbindungsname.

strPhoneBookOrModem

Name des Telefonbuchs, des Modems oder der Direktverbindung zum Herstellen der Verbindung.

tagServerIP

Der Name eines String-Tags, der die IP-Adresse erhält.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

optTagRefresh

Der Name eines Tags, der bei jeder Wertänderung des Tags die Funktion zum Aktualisieren des Rückgabewerts auslöst.

Dieser Parameter optional. Er muss aber angegeben werden,wenn die Funktion in einer Objektanimation wie z.B. eine Text-Daten-Verknüpfung oder eine Position verwendet wird.

Rückgabewert

Diese Funktion gibt die folgenden möglichen Werte zurück:

Wert	Beschreibung
-5	**DialGetServerIP** wurde in **IndRAS.dll** nicht gefunden.
-4	Ungültiger Typ, **numType** prüfen.
-3	Falsche Anzahl Parameter.
-2	**DialStatus** wurde in **IndRAS.dll** nicht gefunden.
-1	**IndRAS.dll** wurde nicht geladen.
n	Von **DialStatus** zurückgegebener Statuscode.

Die IP-Adresse wird in den von **tagServerIP** festgelegten Tag geschrieben.

☞ Diese Funktion muss auf der RAS-Client-Station ausgeführt werden und "nicht" auf der Serverstation. (Im Prinzip ruft die Station die IP-Adresse von dem Server ab, mit dem sie verbunden ist.)

Beispiele

Verwenden des Telefonbuchnamens "Office DialUp" und Schreiben der resultierenden IP-Adresse in **ServerIPTag**:

DialGetServerIP(0, "Office DialUp", "ServerIPTag")

Verwenden des Modemnamens "USRobotics_SportsterFaxModem" zum Schreiben der resultierenden IP-Adresse in **ServerIPAddress** und Aktualisieren zu jeder Sekunde (d.h. jedes Mal, wenn sich der System-Tag **Second** ändert:

DialGetServerIP(1, "USRobotics_SportsterFaxModem", "ServerIPAddress", Second)

Verwenden des Direktverbindungsnamens "DirectDial" und Schreiben der resultierenden IP-Adresse in **IPAdd**:

DialGetServerIP(2, "DirectDial", "IPAdd")

19.24.5 DialStatus

Ermittelt den Status der Einwahlverbindung.

☞ Diese Funktion wird unter Windows 7 nicht unterstützt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DialStatus	Einwahl	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt

Syntax

```
DialStatus( numType, strPhonebookEntryOrModem, "optStatus", optRefresh )
```

Eingaben

numType

Numerischer Parameter, der den Inhalt für den Parameter **strPhonebookEntryorModem** festlegt.

- 0: Name des Telefonbuchs
- 1: Name des Modems
- 2: Name der Direktverbindung

strPhonebookEntryOrModem

Name des Telefonbuchs, des Modems oder der Direktverbindung zum Herstellen der Verbindung. Der Parameter **numType** legt fest, welche der Methoden verwendet wird.

optStatus

"Optional" Der Name des String-Tags, der die Statusmeldung erhält.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

optRefresh

"Optionaler" Tag, der bewirkt, dass die Funktion Ihren Rückgabewert aktualisiert. Dieser Parameter ist optional, muss aber konfiguriert werden, wenn diese Funktion in einer Objektanimation, Seite 243 (z.B. Textdatenverknüpfung, Position) verwendet wird.

Rückgabewert

Wert	Beschreibung
-5	Telefonbuch oder Modem nicht vorhanden
-4	Ungültiger Wert für den Parameter **numType** ("0" oder "1"
-3	Fehler: Ungültige Anzahl Parameter (mindestens 2)
-2	Fehler: **INDRAS.DLL** beschädigt

Anhang: WinStudio-interne Skriptsprache

Wert	Beschreibung
-1	Fehler: **INDRAS.DLL** nicht gefunden
0	Öffne Port...
1	Port wurde erfolgreich geöffnet.
2	Verbindungsaufbau mit dem Gerät...
3	Das Gerät hat erfolgreich eine Verbindung hergestellt.
4	Alle Geräte in der Gerätekette wurden erfolgreich verbunden.
5	Überprüfe Benutzernamen und das Passwort...
6	Es ist ein Authentifizierungsfehler aufgetreten.
7	Erneuten Gültigkeitsprüfungsversuch mit neuem Benutzer angefordert.
8	Server hat eine Rückrufnummer angefordert.
9	Der Client hat eine Passwortänderung angefragt
10	Registrierung Ihres Rechners im Netzwerk...
11	Die Berechnungsphase für die Verbindungsgeschwindigkeit beginnt...
12	Die Authentifizierungsanfrage wird bestätigt.
13	Reauthentifizierung (nach Rückruf) beginnt.
14	Der Client hat die Authentifizierung erfolgreich abgeschlossen.
15	Die Leitung wird gleich für den Rückruf getrennt.
16	Verzögerung für Zurücksetzen des Modems für Rückruf.
17	Warten auf eingehenden Anruf vom Server.
18	Hochrechnungsergebnisdaten verfügbar.
19	Benutzerauthentifizierung wird eingeleitet oder neu versucht.
20	Client wurde zurückgerufen und setzt die Authentifizierung gleich fort.
21	Anmelden am Netzwerk...
22	Untereintrag wurde verbunden.
23	Untereintrag wurde getrennt
24	Terminalzustand unterstützt von **RASPHONE.EXE**.
25	Zustand erneute Authentifizierung unterstützt von **RASPHONE.EXE**.
26	Callbackzustand unterstützt von **RASPHONE.EXE**.
27	Zustand Passwortänderung unterstützt von **RASPHONE.EXE**.
8192	Verbindung zum Remote-Server erfolgreich aufgebaut!
8193	Getrennt.

Beispiele

Tag-Name	Ausdruck
Tag	DialStatus(0, "Office DialUp")
Tag	DialStatus(1, "USRobotics_SportsterFaxModem", "StatusMessage", second)
Tag	DialStatus(2, "DirectDial", "DialupError")

Weitere Informationen unter: FindModem(), Seite 758

19.24.6 DialUp

Baut eine Einwahlverbindung auf.



Diese Funktion wird unter Windows 7 nicht unterstützt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DialUp	Einwahl	asynchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt

Syntax

```
DialUp( numType, strPhonebookEntryOrModem, strUserName, strPassword, optStrDomain, strPhoneNumber )
```

Eingaben

numType

Ein numerischer Merker, der den Inhalt für den Parameter **strPhonebookEntryorModem** festlegt.

- 0: Name des Telefonbuchs
- 1: Name des Modems
- 2: Name der Direktverbindung

strPhonebookEntryOrModem

Name des Telefonbuchs, des Modems oder der Direktverbindung zum Herstellen der Verbindung. Der Parameter **numType** legt fest, welche der Methoden verwendet wird.

strUserName

Benutzername für die Anmeldung.

strPassword

Passwort für die Anmeldung.

optStrDomain

"Optional" Der Domänenname für die Anmeldung.

strPhoneNumber

Die zu wählende Telefonnummer (wird nur verwendet, wenn Parameter numType=1).

Rückgabewert

Wert	Beschreibung
0	OK: Einwahl gestartet
−1	Fehler: **INDRAS.DLL** nicht gefunden
−2	Fehler: **INDRAS.DLL** beschädigt
−3	Fehler: Ungültige Anzahl Parameter (mindestens 5)
-4	Ungültiger Wert für den Parameter **numType** (0 oder 1)

Wert	Beschreibung
-5	Ungültiger Wert für den Parameter **strPhonebookEntryOrModem** (String)
-6	Telefonbuch oder Modem nicht vorhanden
-7	Telefonbuch oder Modem in Benutzung
-8	Abhängig vom Parameter **numType**: • Wenn **numType** = 0: Eigenschaften konnten nicht vom Telefonbuch gelesen werden. • Wenn **numType** = 1: Mehr als 1000 Verbindungen gleichzeitig aktiviert.
-9	Vorläufiges Telefonbuch konnte nicht erstellt werden.

☞ Der RAS-Server des Betriebssystems führt die Einwahl für Windows-PC-Stationen automatisch aus.

Beispiele

Tag-Name	Ausdruck
Tag	DialUp(0, "OfficeDialup", "Guest", "Password")
Tag	DialUp(1, "USRobotics_SportsterFaxModem", "HR12378", "HRPass", "15125554321")
Tag	DialUp(2, "DirectDial", "Rberton", "MyPassword", "156.48.25.0")

Weitere Informationen unter: FindModem(), Seite 758

19.24.7 DialUpToCE

Führt das Programm "DialUpToCE" aus, das die notwendigen Informationen für Rückruf an den Server durch **CERasSvr.exe** sendet.

☞ Diese Funktion wird unter Windows 7 nicht unterstützt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DialUpToCE	Einwahl	asynchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt

Syntax

```
DialUpToCE( numModem, strDialPhone, strMyNumber, strUser, strPassword, optStrDomain, optAutoDial, optAutoClose )
```

Eingaben

numModem

Das Modem für die Einwahl in die Windows-CE-Remote-Station.

strDialPhone

Die Telefonnummer der Windows-CE-Remote-Station.

strMyNumber

Die an die Windows-CE-Remote-Station gesendete Telefonnummer. **CE-RasSvr.exe** ruft diese Telefonnummer zurück.

strUser

Der Benutzername zum Senden an die Windows-CE-Remote-Station. **CE-RasSvr.exe** verwendet diesen Namen nach dem Rückruf zum Verbinden mit dem Windows XP/Vista/7-Rechner.

strPassword

Das Passwort zum Senden an die Windows-CE-Remote-Station. **CE-RasSvr.exe** verwendet dieses Passwort nach dem Rückruf zum Verbinden mit dem Windows XP/Vista/7-Rechner.

optStrDomain

"Optional" Der Domänenname für die Anmeldung.

optAutoDial

"Optionaler" Tag, für den folgende Werte möglich sind:

- 1: Löst die **DialupToCE**-Verbindung automatisch, wenn die Funktion ausgeführt wird
- 0: Fordert vor dem automatischen Lösen der **DialupToCE**-Verbindung bei Ausführung der Funktion eine Bestätigung an

optAutoClose

"Optionaler" Tag, für den folgende Werte möglich sind:

- 1: Schließt den Dialog **DialUpToCE** automatisch nach Anwahl der Windows-CE-Remote-Station
- 0: Lässt den Dialog **DialUpToCE** geöffnet

Rückgabewert

Wert	Beschreibung
0	Fehlgeschlagen, konnte "DialUpToCE" nicht anrufen.
1	Erfolgreich, "DialUpToCE" wurde ausgeführt.

☞ Das Programm **DialUpToCE** wurde zur Einwahl in eine Remote-Windows-CE-Station entwickelt. Da Windows CE v.3.00 keinen RAS-Server zur Verfügung stellt, muss vor dem Annehmen eines Anrufs das Programm **CERasSvr.exe** und ein Rückruf zum Windows XP/Vista/7-Rechner mit den von der Funktion **DialUpToCE** verwendeten Parametern ausgeführt werden. Der RAS-Server-Dienst muss auf dem Windows XP/Vista/7-Rechner zum Rückruf vom Windows-Embedded-Gerät und Einstellen der TCP/IP-Verbindung konfiguriert werden.

Beispiele

Tag-Name	Ausdruck
Tag	DialUpToCE(0, "12344321", "98765432", "Administrator", "MyPass")
Tag	DialUpToCE(0, "12344321", "98765432", "Administrator", "MyPass", "", 1, 1)

19.24.8 FindAllDevices

Gibt eine Liste aller verfügbaren Modems und direkten Verbindungsschnittstellen (COM-Ports) auf der lokalen Station zurück.

☞ Diese Funktion wird unter Windows 7 nicht unterstützt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
FindAllDevices	Einwahl	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt

Syntax

```
FindAllDevices( "tagArray" )
```

Eingaben

tagArray

Name eines String-Bereichs-Tags für die Liste aller Modems und direkten Verbindungsschnittstellen.

Rückgabewert

Gibt die Anzahl der gefundenen Modems und/oder Schnittstellen zurück.

Beispiele

Tag-Name	Ausdruck
Tag	FindAllDevices("SerialConnections[1]")

19.24.9 FindModem

Gibt eine Liste aller verfügbaren Modems in der lokalen Station zurück.

☞ Diese Funktion wird unter Windows 7 nicht unterstützt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
FindModem	Einwahl	synchron	unterstützt (siehe Hinweis)	unterstützt	unterstützt

Syntax

```
FindModem( "tagArray" )
```

Eingaben

tagArray

Name eines String-Bereichs-Tags für die Liste der verfügbaren Modems.

Rückgabewert

Gibt die Anzahl der gefundenen Modems zurück.

☞ Diese Funktion kann zum Abruf des seriellen Schnittstellennamens für eine Einwahlverbindung über ein Modem verwendet werden. Die ermittelten Informationen können für den Parameter "strPhonebookEntryOrModem" in der Funktionen DialError, Seite 744, DialStatus, Seite 753, DialUp, Seite 755 und HangUp, Seite 759 verwendet werden.

Beispiele

Tag-Name	Ausdruck
Tag	FindModem("Modems[1]")

19.24.10 HangUp

Beendet eine Einwahlverbindung.

☞ Diese Funktion wird unter Windows 7 nicht unterstützt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
HangUp	Einwahl	synchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt

Syntax

```
HangUp( numType, strPhonebookEntryOrModem )
```

Eingaben

numType

Ein numerischer Merker, der die Bedeutung des Parameters **strPhonebookEntryorModem** festlegt.

- 0: Name des Telefonbuchs
- 1: Name des Modems
- 2: Name der Direktverbindung

strPhonebookEntryOrModem

Name des Telefonbuchs, des Modem oder der Direktverbindung zum Herstellen der Verbindung. Der Parameter **numType** legt fest, welche der Methoden verwendet wird.

Rückgabewert

Wert	Beschreibung
0	OK.
-1	Fehler: **INDRAS.DLL** nicht gefunden
-2	Fehler: **INDRAS.DLL** beschädigt
-3	Ungültiger Wert für den Parameter **numType** ("0" oder "1"
-4	Telefonbuch oder Modem nicht vorhanden
-5	Keine konfigurierten Modems vorhanden

Beispiele

Tag-Name	Ausdruck
Tag	HangUp(0, "OfficeDialup")
Tag	HangUp(1, "USRobotics_SportsterFaxModem")
Tag	HangUp(2, "DirectDial", "Rberton", "MyPassword")

Weitere Informationen unter: FindModem(), Seite 758

19.24.11 PhoneDialUp

Wählt eine Telefonnummer mit dem Telephony Application Program Interface (TAPI) an.

☞ Diese Funktion wird unter Windows 7 nicht unterstützt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
PhoneDialUp	Einwahl	asynchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt

Syntax

```
PhoneDialUp( strPhoneNumber, optStrModemName )
```

Eingaben

strPhoneNumber

Telefonnummer, die die Funktion anruft.

optStrModemName

Name des zum Wählen verwendeten Modems. Wird kein Modem angegeben, verwendet die Funktion das erste im Betriebssystem gefundene Modem.

Rückgabewert

Wert	Beschreibung
0	OK (Wählen ausgelöst)
−1	Ungültige Anzahl Parameter
−3	Bibliothek INDTAPI.DLL nicht gefunden
-4	Funktion PhoneDialUp() wird von der aktuellen INDTAPI.DLL-Bibliothek nicht unterstützt

Beispiele

Tag-Name	Ausdruck
	PhoneDialUp ("512-123-4567")
	PhoneDialUp (StringPhoneNumberTag)
	PhoneDialUp (StringPhoneNumberTag, StringModemNameTag)

19.24.12 PhoneDisableListen

Beendet die Empfangsbereitschaft des Modems für eingehende Anrufe.

☞ Diese Funktion wird unter Windows 7 nicht unterstützt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
PhoneDisableListen	Einwahl	synchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt

Syntax

```
PhoneDisableListen( optStrModemName )
```

Eingaben

optStrModemName

Name des zum Wählen verwendeten Modems. Wird kein Modem angegeben, verwendet die Funktion das erste im Betriebssystem gefundene Modem.

Rückgabewert

Wert	Beschreibung
1	OK (keine Bereitschaft für eingehende Anrufe)
−1	Bibliothek INDTAPI.DLL nicht gefunden
−2	Funktion PhoneDisableListen () wird von der aktuellen **INDTAPI.DLL**-Bibliothek nicht unterstützt

Beispiele

Tag-Name	Ausdruck
	PhoneDisableListen()
	PhoneDisableListen("Hayes Compatible Modem on COM1")
	PhoneDisableListen(StringModemNameTag)

Weitere Informationen unter: PhoneEnableListen(), Seite 761

19.24.13 PhoneEnableListen

Nimmt die Empfangsbereitschaft des Modems für eingehende Anrufe wieder auf.

Diese Funktion wird unter Windows 7 nicht unterstützt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
PhoneEnableListen	Einwahl	synchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt

Syntax

```
PhoneEnableListen( optStrModemName )
```

Eingaben

optStrModemName

Name des zum Wählen verwendeten Modems. Wird kein Modem angegeben, verwendet die Funktion das erste im Betriebssystem gefundene Modem.

Rückgabewert

Wert	Beschreibung
1	OK (Bereitschaft für eingehende Anrufe)
0	Fehler beim Ausführen der Funktion PhoneEnableListen()
−1	Bibliothek INDTAPI.DLL nicht gefunden
−2	Funktion PhoneEnableListen() wird von der aktuellen INDTAPI.DLL-Bibliothek nicht unterstützt

Beispiele

Tag-Name	Ausdruck
	PhoneEnableListen()
	PhoneEnableListen("Hayes Compatible Modem on COM1")
	PhoneEnableListen(StringModemNameTag)

Weitere Informationen unter: PhoneDisableListen(), Seite 760

19.24.14 PhoneHangUp

Beendet eine zuvor mit der Funktion "PhoneDialUp" hergestellte Verbindung..

☞ Diese Funktion wird unter Windows 7 nicht unterstützt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
PhoneHangUp	Einwahl	synchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt

Syntax

```
PhoneHangUp( optStrModemName )
```

Eingaben

optStrModemName

Name des zum Wählen verwendeten Modems. Wird kein Modem angegeben, verwendet die Funktion das erste im Betriebssystem gefundene Modem.

Rückgabewert

Wert	Beschreibung
1	OK (Einwahl wurde unterbrochen)
−1	Bibliothek INDTAPI.DLL nicht gefunden
−2	Funktion PhoneHangUp () wird von der aktuellen INDTAPI.DLL-Bibliothek nicht unterstützt

Beispiele

Tag-Name	Ausdruck
	PhoneHangUp()
	PhoneHangUp("Hayes Compatible Modem on COM1")
	PhoneHangUp(StringModemNameTag)

19.24.15 PhoneStatus

Überprüft den Status der aktuellen Verbindungen.

☞ Diese Funktion wird unter Windows 7 nicht unterstützt.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
PhoneStatus	Einwahl	synchron	unterstützt (siehe Hinweis)	nicht unterstützt	unterstützt

Syntax

```
PhoneStatus ( "strStatus", optStrModemName )
```

Eingaben

strStatus

Name des Tags, der den Statusbeschreibungstext erhält

optStrModemName

Name des zum Wählen verwendeten Modems. Wird kein Modem angegeben, verwendet die Funktion das erste im Betriebssystem gefundene Modem

Rückgabewert

Wert	Beschreibung
0	Bereit zum Tätigen eines Anrufs
1	Anruf wurde abgebrochen
2	Leitung klingelt
3	Wählton
4	Anruf wählen
5	Anruf ist im Ablauf
6	Rückruf
7	Leitung besetzt
8	Leitung ist frei
9	Getrennt

Beispiele

Tag-Name	Ausdruck
Tag Status Code	PhoneStatus("String Tag Status")
Tag Status Code	PhoneStatus("String Tag Status", "Hayes Compatible Modem on COM1")
Tag Status Code	PhoneStatus("String Tag Status", "StringModemNameTag")

19.25 ActiveX- und .NET-Control-Funktionen

19.25.1 ActiveX- und .NET-Control-Funktionen – Allgemeines

Diese Funktionen dienen zum Ausführen von Funktionen in ActiveX- und .NET-Objekten, die in WinStudio-Bildern eingebettet sind, sowie zum Abrufen und Ändern von Eigenschaftswerten dieser Objekte.

ActiveX Control und .NET Control

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
XGet, Seite 763	asynchron	unterstützt	nur ActiveX Controls	unterstützt
XRun, Seite 764	asynchron	unterstützt	nur ActiveX Controls	unterstützt
XSet, Seite 765	asynchron	unterstützt	nur ActiveX Controls	unterstützt

19.25.2 XGet

Diese Funktion ruft den aktuellen Wert einer Eigenschaft eines ActiveX- und .NET-Objekts ab.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
XGet	ActiveX und .NET Control	asynchron	unterstützt	nur ActiveX-Objekte (siehe Hinweis)	unterstützt

Syntax

```
XGet ( strName , strProperties )
```

Eingaben

strName

Der eindeutige Name des ActiveX- und .NET-Objekts, der im Feld **Name** des Dialogs "Objekteigenschaften" konfiguriert wurde.

strProperties

Der Name der Eigenschaft, deren Wert abgerufen werden soll. Die verfügbaren Eigenschaften werden im Dialog "Konfiguration" (für ActiveX-Objektet) oder "Members" (für ein .NET-Objekte) angezeigt.

Rückgabewert Gibt den Wert der angegebenen Eigenschaft zurück.

☞ Diese Funktion kann nicht in Tasks, Seite 61 oder im Skript Globale Prozeduren, Seite 799 verwendet werden. Außerdem werden .NET Controls unter Windows CE nicht unterstützt.

Beispiele Ruft den aktuellen Wert der Farbeigenschaft auf dem ActiveX-Objekt „ActXRec“ ab:

XGet("ActXRec", "Color")

19.25.3 XRun

Diese Funktion führt eine Methode auf einem ActiveX- oder .NET-Objekt aus.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
XRun	ActiveX und .NET Control	asynchron	unterstützt	nur ActiveX-Objekte (siehe Hinweis)	unterstützt

Syntax

```
XRun ( strName , strMethod , Parameter1 ,ParameterN)
```

Eingaben

strName

Der eindeutige Name des ActiveX- und .NET-Objekts, der im Feld **Name** des Dialogs "Objekteigenschaften" konfiguriert ist.

strMethod

Die auszuführende Methode. Die verfügbaren Methoden werden im Dialog "Konfiguration" (für ActiveX-Objekte) oder "Members" (für ein .NET-Objekte) angezeigt.

Parameter(1…N)

Verschiedene Datentypen, die für die Ausführung der Methode notwendig sind. Die Anzahl der Parameter kann zwischen 0 und 255 liegen und hängt von der angegebenen Methode ab. Die Datentypen (z.B. Boolean, Integer, Real oder String) von referenzierenden Tags müssen den Parametern in der Methode entsprechen.

Rückgabewert Liefert das Ergebnis der Methodenausführung zurück. Nicht alle Methoden geben Ergebnisse zurück.

Diese Funktion kann nicht in Tasks, Seite 61 oder im Skript Globale Prozeduren, Seite 799 verwendet werden. Außerdem werden .NET-Objekte unter Windows CE nicht unterstützt.

Beispiele

Ausführen der Methode XPos auf dem ActiveX-Objekt „ActXCir“ mit vier Originalwerten für die Methode:

XRun("ActXCir", "XPos", FALSE, 12, 4.6, "Das ist mein Text.")

Ausführen der Methode XPos auf dem ActiveX-Steuerelemente „ActXCir“ mit vier referenzierenden Tags für die Methode:

XRun("ActXCir", "XPos", TagA, TagB, TagC, TagD)

19.25.4 XSet

Diese Funktion legt den Wert einer Eigenschaft in einem ActiveX- und .NET-Objekt fest.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
XSet	ActiveX und .NET Control	asynchron	unterstützt	nur ActiveX-Objekte (siehe Hinweis)	unterstützt

Syntax

```
XSet(strName,strProperties,Value)
```

Eingaben

strName

Der eindeutige Name des ActiveX- und .NET-Objekts, der im Feld **Name** des Dialogs **Objekteigenschaften** konfiguriert ist.

strProperties

Die Eigenschaft, deren Wert festgelegt werden soll. Die verfügbaren Eigenschaften werden im Dialog **Konfiguration** (für ActiveX-Objekte) oder **Members** (für .NET-Objekte) angezeigt.

Wert

Beliebiger Wert, der die vorgewählte Eigenschaft im ActiveX- bzw. .NET-Objekt überschreibt.

Rückgabewert

Diese Funktion gibt keinen Wert zurück.

Diese Funktion kann nicht in Tasks, Seite 61 oder im Skript Globale Prozeduren, Seite 799 verwendet werden. Außerdem werden .NET-Objekte unter Windows CE nicht unterstützt.

Beispiele

Eigenschaft "Display" des ActiveX-Objekts „ActXDisplay“ auf „Status Normal“ ändern:

XSet("ActXDisplay", "Display", "Status Normal")

19.26 Funktionen zur Ereignisprotokollierung

19.26.1 Funktionen zur Ereignisprotokollierung – Allgemeines

Diese Funktionen dienen dem Senden von Ereignissen und Kommentaren an die Ereignisprotokollierung.

Anhang: WinStudio-interne Skriptsprache

Ereignisprotokollierung

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
SendEvent, Seite 766	synchron	unterstützt	unterstützt	unterstützt

19.26.2 SendEvent

Sendet ein Ereignis an die Ereignisprotokollierung.

Die Funktion generiert einen Eintrag in der Ereignisprotokollierungsdatei. Wenn die Option **Kommentar** aktiviert ist, wird der Benutzer aufgefordert, nach Ausführen der Funktion SendEvent() einen Kommentar einzugeben. Dieser Kommentar wird in der Ereignisprotokolldatei gespeichert.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SendEvent	Ereignisprotokol-lierung	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
SendEvent( strEvent, optBoolFlag, optStrComment )
```

Eingaben

strEvent

Text zum Speichern in der Ereignisprotokollierung.

optBoolFlag

Wenn der Parameter weggelassen wird oder 0 (null) ist, hat das Ereignis keinen Kommentar. Andernfalls wird mit dem Ereignis ein Kommentar gespeichert.

optStrComment

Kommentartext der beim Auslösen der Funktion in die Ereignisprotokolldatei eingetragen wird (wenn optBoolFlag =1). Dieser Parameter ist optional. Ohne Textvorgabe wird der Benutzer zur Eingabe eines Kommentars über einen Standard-Dialog aufgefordert.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
1	Ereignisprotokollierung im Dialog **Ereigniseinstellungen** deaktiviert
2	Ereignisprotokollierung aktiviert, aber benutzerdefinierte Meldungen sind im Dialog **Ereigniseinstellungen** deaktiviert

☞ Diese Funktion ist synchron. Daher endet die Ausführung der Funktion erst, nachdem die Ereignisdaten (einschließlich des optionalen Kommentars) in der Datenbankdatei gespeichert wurden. Es wird empfohlen, diese Funktion nur dann in einer Hintergrund-Task zu konfigurieren (z.B. "Mathematik" oder "Scheduler"), wenn der Kommentar nicht verwendet oder direkt in der Funktion (ohne Dialogeingabe) eingetragen wird.

Beispiele

Tag-Name	Ausdruck
Tag	SendEvent("Valve Open") // Saves the event message.
Tag	SendEvent("Valve Open Oven No. " + OvenID) // Saves the event message concatenated with the value of the OvenID tag.
Tag	SendEvent("Valve Open" , 1) // Displays the dialog where the operator can type comments.
Tag	SendEvent("Valve Open" , 1 , TagComment) // Saves the event message with the comment configured in the TagComment tag.

19.27 FTP-Funktionen

19.27.1 FTP-Funktionen – Allgemeines

Diese Funktionen dienen dem Konfigurieren der FTP-Einstellungen sowie dem Hoch- und Herunterladen von Dateien zu/von einem Remote-Server.

FTP

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
CnfFTP, Seite 767	synchron	unterstützt	unterstützt	unterstützt
FTPGet, Seite 768	asynchron	unterstützt	unterstützt	unterstützt
FTPPut, Seite 770	asynchron	unterstützt	unterstützt	unterstützt
FTPStatus, Seite 771	synchron	unterstützt	unterstützt	unterstützt

19.27.2 CnfFTP

Diese Funktion konfiguriert die von anderen Funktionen im Projekt verwendeten FTP-Einstellungen, z.B. die Funktionen "FTPGet" und "FTPPut".

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
CnfFTP	FTP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
CnfFTP(strServer, optStrUser, optStrPwn, optNumPassiveMode, optNumPort)
```

Eingaben

strServer

Die Adresse des FTP-Servers.

optStrUser

Der Benutzername für das FTP-Konto.

Dieser Parameter ist optional; wird kein Wert angegeben, wird standardmäßig der Benutzername „anonymous“ verwendet.

optStrPwd

Das Passwort für das FTP-Konto.

Dieser Parameter ist optional; wird kein Wert angegeben, bleibt das Passwortfeld leer.

optNumPassiveMode

Ein numerischer Merker zur Festlegung, ob der passive FTP-Modus aktiviert ist. (Passiv-FTP kann zum Umgehen einiger Firewall-Konfigurationen verwendet werden.)

Wert	Beschreibung
0	Passiver FTP-Modus ist deaktiviert.
1	Passiver FTP-Modus ist aktiviert.

Dieser Parameter ist optional; wird kein Wert angegeben, wird der passive FTP-Modus standardmäßig deaktiviert.

optNumPort

Die TCP/IP-Portnummer.

Dieser Parameter ist optional; wird kein Wert angegeben, wird standardmäßig Port 21 verwendet.

Rückgabewert

Wert	Beschreibung
0	Erfolgreich
-1	Ungültige Parameteranzahl
-2	Ungültiger Servername
-3	Ungültiger Benutzername



Diese Funktion stellt keine tatsächliche Verbindung zum Server her, so dass diese Fehlercodes keine Aussage über die Verbindungsqualität darstellen. Sie zeigen lediglich an, ob die FTP-Einstellungen erfolgreich konfiguriert wurden.



Diese Funktion muss vor Verwendung der Funktionen FTPGet, Seite 768 und FTPPut, Seite 770 einmalig aufgerufen werden, um Dateien übertragen zu können.

Beispiel

Tag-Name	Ausdruck
Tag	CnfFTP("ftp.mycompany.com", "admin", "12345", 1) // Configures the FTP server using passive mode.

19.27.3 FTPGet

Ruft eine Datei vom Remote-Server ab und speichert diese auf dem lokalen Rechner.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
FTPGet	FTP	asynchron	unterstützt	unterstützt	unterstützt

Syntax

```
FTPGet( strRemoteFile, strLocalFile, numTransferType, numOverwrite )
```

Eingaben

strRemoteFile

Der vollständige Pfad und Name der gewünschten Datei auf dem Remote-Server, mit der Syntax **"/dateipfad/dateiname.erweiterung"**. Einige FTP-Server unterscheiden zwischen Groß- und Kleinschreibung, es sollte also stets auf korrekte Groß- und Kleinschreibung geachtet werden.

strLocalFile

Vollständiger Pfad und Name, unter dem die Datei auf dem lokalen Rechner gespeichert werden soll, mit der Syntax **"C:\dateiname.erweiterung“**.

numTransferType

Numerischer Parameter zur Festlegung der Dateiübertragungsart (siehe Tabelle). Dieser Parameter ist optional; wird kein Wert angegeben, wird standardmäßig die Übertragungsart "Unbekannt" (0) verwendet.

Wert	Beschreibung
0	Unbekannt
1	ASCII
2	Binär
10	Unbekannt, ohne Zwischenspeicherung
11	ASCII, ohne Zwischenspeicherung
12	Binär, ohne Zwischenspeicherung

☞ Es wir empfohlen, Option "10" zu verwenden. Diese ermittelt automatisch das Format (ASCII oder Binär) der Remote-Datei und legt die Übertragungsart entsprechend fest. Sie zwingt das Laufzeitsystem außerdem zum Herunterladen der Datei vom tatsächlichen FTP-Server und nicht vom dazwischenliegenden Proxy- oder Cache-Server.

numOverWrite

Numerischer Wert zur Festlegung, ob die lokale Datei (festgelegt durch **strLocalFile**) überschrieben werden darf, wenn sie bereits existiert. Dieser Parameter ist optional; der Standardwert ist 0.

Wert	Beschreibung
0	Nicht überschreiben – Fehler ausgeben, wenn Datei bereits vorhanden ist.
1	Überschreiben.

Rückgabewert

Wert	Beschreibung
1	FTP-Verbindung konnte nicht geöffnet werden
0	Erfolgreich
-1	Ungültige Anzahl Parameter

Wert	Beschreibung
-2	Unbekannter Systemfehler
-3	Ungültige Remote-Datei
-4	Ungültige lokale Datei
-5	Ungültige Übertragungsart

Vor dem Aufrufen dieser Funktion müssen mit Hilfe der Funktion CnfFTP, Seite 767 die FTP-Einstellungen konfiguriert werden (d.h. Serveradresse und Anmeldung).

Da die Funktion asynchron ausgeführt wird, muss zur Überprüfung des Übertragungsergebnisses die Funktion FTPStatus, Seite 771 aufgerufen werden.

Beispiel

Tag-Name	Ausdruck
Tag	FTPGet("\Reports\040303.txt", "C:\Report.txt") // Retrieves the file **040303.txt** in the **Reports** directory, from the FTP server that was previously specified by CnfFTP() page 767, It then saves the file locally at **C:\Report.txt**.

19.27.4 FTPPut

Überträgt eine Datei auf einen Remote-Server.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
FTPPut	FTP	asynchron	unterstützt	unterstützt	unterstützt

Syntax

```
FTPPut( strLocalFile, strRemoteFile, numTransferType )
```

Eingaben

strLocalFile

Vollständig qualifizierter Name der lokalen Datei (z.B. **"C:\datei.erweiterung"**)

strRemoteFile

Vollständig qualifizierter Name der Remote-Datei (z.B. **"/ordner/datei.erweiterung"**). Es ist zu beachten, dass einige FTP-Server zwischen Groß- und Kleinschreibung unterscheiden. Darauf ist beim Eingeben des Dateinamens zu achten.

numTransferType

- 0 - unbekannt
- 1 - ASCII
- 2 - binär

Standard ist 0.

Rückgabewert

Wert	Beschreibung
1	FTP-Thread konnte nicht erstellt werden
0	Erfolgreich
−1	Ungültige Anzahl Parameter
−2	Unbekannter Systemfehler
−3	Ungültige Remote-Datei
-4	Ungültige lokale Datei
-5	Ungültige Übertragungsart

☞ Vor dem Aufrufen dieser Funktion müssen die FTP-Einstellungen mit Hilfe der Funktion CnfFTP, Seite 767 konfiguriert werden (d.h. Serveradresse und Anmeldung).

Da die Funktion asynchron ausgeführt wird, muss zur Überprüfung des Übertragungsergebnisse die Funktion FTPStatus, Seite 771 aufgerufen werden.

Beispiel

Tag-Name	Ausdruck
Tag	FTPPut("C:\Report.txt", "\Reports\040303.txt") // Retrieves the file **C:\Report.txt** with the name **040303.txt**in the folder **Reports** in the FTP Server.

19.27.5 FTPStatus

Diese Funktion gibt den aktuellen Status einer mit "FTPGet" oder "FTPPut" gestarteten Dateiübertragung zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
FTPStatus	FTP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
FTPStatus("strStatusTag")
```

Eingaben

strStatusTag

Name des String-Tags, der bei Rückmeldung der Funktion die Statusbeschreibung (im Klartext) erhält.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert

Wert	Beschreibung
1	Transaktion erfolgreich ausgeführt
2	Name auflösen
3	Name aufgelöst
4	Verbindung mit Server herstellen

5	Verbindung mit Server hergestellt
6	Verbindung schließen
7	Verbindung geschlossen
8	Anfrage senden
9	Anfrage gesendet
10	Antwort erhalten
11	Zwischenantwort erhalten
12	Antwort erhalten
13	Anfrage beendet
0	Es wird keine Transaktion ausgeführt
−2	Ungültige **opttagErrorDescription**
-6	Fehler beim Öffnen der Verbindung (mehr Details siehe Status-String)
-7	Fehler beim Aufbau der Verbindung (mehr Details siehe Status-String)
-8	Fehler beim Empfangen der Datei (mehr Details siehe Status-String)
-9	Übertragung steht an

Beispiel

Tag-Name	Ausdruck
StatusCode	FTPStatus("StatusDescription") // Retrieves the status of a current transfer. The return code is stored in the **StatusCode** tag and the description in the **StatusDescription** tag.

19.28 Datenbank-/ERP-Funktionen

19.28.1 Datenbank-/ERP-Funktionen – Allgemeines

Diese Funktionen dienen der Interaktion mit externen Datenbanken und ERP-Systemen unter Verwendung von SQL-ähnlichen Befehlen.

DB/ERP

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
DBCursorClose, Seite 773	synchron	unterstützt	unterstützt	unterstützt
DBCursorCurrentRow, Seite 776	synchron	unterstützt	unterstützt	unterstützt
DBCursorGetValue, Seite 776	synchron	unterstützt	unterstützt	unterstützt
DBCursorMoveTo, Seite 777	synchron	unterstützt	unterstützt	unterstützt
DBCursorNext, Seite 778	synchron	unterstützt	unterstützt	unterstützt
DBCursorOpen, Seite 779	synchron	unterstützt	unterstützt	unterstützt
DBCursorOpenSQL, Seite 781	synchron	unterstützt	unterstützt	unterstützt
DBCursorPrevious, Seite 783	synchron	unterstützt	unterstützt	unterstützt
DBCursorRowCount, Seite 783	synchron	unterstützt	unterstützt	unterstützt
DBDelete, Seite 784	synchron	unterstützt	unterstützt	unterstützt

Funktion	Ausführung	Windows PC	Windows CE	Thin Client
DBExecute, Seite 785	synchron	unterstützt	unterstützt	unterstützt
DBInsert, Seite 787	synchron	unterstützt	unterstützt	unterstützt
DBSelect, Seite 788	synchron	unterstützt	unterstützt	unterstützt
DBUpdate, Seite 790	synchron	unterstützt	unterstützt	unterstützt
SyncAlarm, Seite 791	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt
SyncAlarmStatus, Seite 792	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
SyncEvent, Seite 793	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt
SyncEventStatus, Seite 793	synchron	unterstützt	unterstützt	auf dem Server ausgeführt
SyncTrend, Seite 794	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt
SyncTrendStatus, Seite 795	synchron	unterstützt	unterstützt	auf dem Server ausgeführt

19.28.2 DBCursorClose

Schließt den Cursor und gibt die Ergebnisreihe frei.

☞ Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Befehlen vertraut machen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DBCursorClose	Datenbank/ERP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DBCursorClose( numCur, "optStrErrorTag" )
```

Eingaben

numCur

Der Cursor-Handle für die Ergebnisreihe. Der Cursor-Handle wird von DBCursorOpen, Seite 779 oder DBCursorOpenSQL, Seite 781 zurückgegeben.

optStrErrorTag

Der Name eines String-Tags, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.

Dieser Parameter ist optional.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert Bei Fehlern wird eine negative Zahl ausgegeben. Andernfalls wird 0 ausgegeben.

☞ Wenn der Cursor geschlossen ist, wird er zerstört und kann nicht wiederverwendet werden. Es muss mithilfe von DBCursorOpen, Seite 779 oder DBCursorOpenSQL, Seite 781 ein neuer Cursor erstellt werden.

Beispiele Verwendung in einem Berechnungsarbeitsblatt:

Tag-Name	Ausdruck
nErrorCode	DBCursorClose(nCursor)

19.28.3 DBCursorColumnCount

Ruft die Gesamtzahl der Spalten in einer SQL-Ergebnisreihe ab.

☞ Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Befehlen vertraut machen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DBCursorColumnCount	Datenbank/ERP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DBCursorColumnCount( numCur, "optStrErrorTag" )
```

Eingaben

numCur

Der Cursor-Handle für die Ergebnisreihe. Der Cursor-Handle wird von DBCursorOpen, Seite 779 oder DBCursorOpenSQL, Seite 781 zurückgegeben.

optStrErrorTag

Der Name eines String-Tags, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.

Dieser Parameter ist optional.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert Gibt die Anzahl der Spalten zurück. Bei Fehlern wird eine negative Zahl ausgegeben.

☞ Siehe auch DBCursorRowCount, Seite 783.

Beispiele Verwendung in einem Berechnungsarbeitsblatt:

Tag-Name	Ausdruck
nErrorC-ode	DBCursorColumnCount(nCursor)

19.28.4 DBCursorColumnInfo

Ruft Informationen über eine Spalte in einer SQL-Ergebnisreihe ab. Die Spalte wird durch eine Nummer und nicht durch einen Namen festgelegt, so dass diese Funktion zum Abrufen unbekannter Spaltennamen verwendet werden kann.

☞ Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Befehlen vertraut machen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DBCursorColum-nInfo	Datenbank/ERP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DBCursorColumnInfo(numCur, numColumn, numTypeInfo, "optStrErrorTag")
```

Eingaben

numCur

Der Cursor-Handle für die Ergebnisreihe. Der Cursor-Handle wird von DBCursorOpen, Seite 779 oder DBCursorOpenSQL, Seite 781 zurückgegeben.

numColumn

Die Anzahl der Spalten, über die Informationen benötigt werden. Es ist zu beachten, dass eine Ergebnisreihe ggf. nur einige der Spalten in der Original-Datenbank enthält.

numTypeInfo

Die Art der Informationen, die aus der Spalte abgerufen werden sollen:

Wert	Beschreibung
0	Spaltenname
1	Datentyp Spalte

optStrErrorTag

Der Name eines String-Tags, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.

Dieser Parameter ist optional.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert Bei Fehlern wird eine negative Zahl ausgegeben. Andernfalls wird 0 ausgegeben.

Beispiele Verwendung in einem Mathematik-Arbeitsblatt:

Tag-Name	Ausdruck
ErrorCode	DBCursorColumnInfo(nCursor, 2, 0) // Gets the column name of the second column in the result set.

19.28.5 DBCursorCurrentRow

Gibt die Zeilennummer der aktuellen Zeile (d.h. der Cursor-Position) in einer SQL-Ergebnisreihe zurück.

☞ Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Befehlen vertraut machen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DBCursorCurrent-Row	Datenbank/ERP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DBCursorCurrentRow( numCur, "optStrErrorTag" )
```

Eingaben

numCur

Der Cursor-Handle für die Ergebnisreihe. Der Cursor-Handle wird von DBCursorOpen, Seite 779 oder DBCursorOpenSQL, Seite 781 zurückgegeben.

optStrErrorTag

Der Name eines String-Tags, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.

Dieser Parameter ist optional.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert Gibt die Nummer der aktuellen Zeile zurück. Bei Fehlern wird eine negative Zahl ausgegeben.

Beispiele Verwendung in einem Mathematik-Arbeitsblatt:

Tag-Name	Ausdruck
nRow	DBCursorCurrentRow(nCursor)

19.28.6 DBCursorGetValue

Gibt den Wert der angegebenen Spalte der aktuellen Zeile (d.h. der Cursor-Position) in einer SQL-Ergebnisreihe zurück.

☞ Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Befehlen vertraut machen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DBCursorGetValue	Datenbank/ERP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DBCursorGetValue(numCur, strColumn, "optStrErrorTag")
```

Eingaben

numCur

Der Cursor-Handle für die Ergebnisreihe. Der Cursor-Handle wird von DBCursorOpen, Seite 779 oder DBCursorOpenSQL, Seite 781 zurückgegeben.

strColumn

Spaltenname.

optStrErrorTag

Der Name eines String-Tags, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.

Dieser Parameter ist optional.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert

Gibt den Wert der angegebenen Spalte zurück. Wenn der Wert NULL oder der Cursor ungültig ist, wird eine leere Zeichenkette mit der Qualität BAD ausgegeben.

Beispiele

Verwendung in einem Mathematik-Arbeitsblatt:

Tag-Name	Ausdruck
Tag	DBCursorGetValue(nCursor, "Column1")

19.28.7 DBCursorMoveTo

Bewegt den Cursor zur angegebenen Spalte in der SQL-Ergebnisreihe und kopiert die Werte dieser Zeilen in die abgebildeten Tags. Wenn die angegebene Zeile nicht vorhanden ist – d.h. wenn sie außerhalb des Bereichs der Ergebnisreihe liegt – gibt die Funktion einen Fehlercode aus und lässt die abgebildeten Tags unverändert.

☞ Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Befehlen vertraut machen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DBCursorMoveTo	Datenbank/ERP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DBCursorMoveTo( numCur, numRow, "optStrErrorTag" )
```

Eingaben

numCur

Der Cursor-Handle für die Ergebnisreihe. Der Cursor-Handle wird von DBCursorOpen, Seite 779 oder DBCursorOpenSQL, Seite 781 zurückgegeben.

numRow

Die Zeile der Ergebnisreihe, zu der der Cursor bewegt wird.

optStrErrorTag

Der Name eines String-Tags, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.

Dieser Parameter ist optional.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert Bei Fehlern wird eine negative Zahl ausgegeben. Andernfalls wird 0 ausgegeben.

Beispiele Verwendung in einem Mathematik-Arbeitsblatt:

Tag-Name	Ausdruck
nErrorCode	DBCursorMoveTo(nCursor, 4) // Moves the cursor to the fourth row of the result set and copies those values.

19.28.8 DBCursorNext

Bewegt den Cursor zur nächsten Zeile in der SQL-Ergebnisreihe und kopiert die Werte dieser Zeilen in die abgebildeten Tags. Wenn keine nächste Zeile vorhanden ist – d.h. wenn die aktuelle Zeile die letzte ist – gibt die Funktion einen Fehlercode aus und lässt die abgebildeten Tags unverändert.

☞ Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Befehlen vertraut machen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DBCursorNext	Datenbank/ERP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DBCursorNext( numCur, "optStrErrorTag" )
```

Eingaben

numCur

Der Cursor-Handle für die Ergebnisreihe. Der Cursor-Handle wird von DBCursorOpen, Seite 779 oder DBCursorOpenSQL, Seite 781 zurückgegeben.

optStrErrorTag

Der Name eines String-Tags, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.

Dieser Parameter ist optional.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert

Bei Fehlern wird eine negative Zahl ausgegeben. Andernfalls wird 0 ausgegeben.

Beispiele

Verwendung in einem Mathematik-Arbeitsblatt:

Tag-Name	Ausdruck
nErrorCode	DBCursorNext(nCursor)

19.28.9 DBCursorOpen

Wählt eine Reihe von Zeilen und Spalten aus einer Datenbanktabelle aus, initialisiert den Cursor in der ersten Zeile der Ergebnisreihe, kopiert die Zeilenwerte in abgebildete Tags und gibt dann einen Cursor-Handle aus, der von anderen DB-/ERP-Funktionen referenziert werden kann.

☞ Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Befehlen vertraut machen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DBCursorOpen	Datenbank/ERP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DBCursorOpen(strDBConn, strTable, optStrCondition, optStrColumns, optStrTags, optStrOrder, "optStrErrorTag")
```

Eingaben

strDBConn

Name der Datenbankverbindung. Verbindungen sind im Ordner Datenbank/ERP, Seite 371 konfiguriert.

strTable

Name der Datenbanktabelle.

optStrCondition

Eine Zeichenkette zur Angabe, welche Tabellenzeilen auszuwählen sind. Dies entspricht dem SQL-Satz WHERE und die Zeichenkette sollte der gleichen Syntax folgen.

Dieser Parameter ist optional. Werden keine Zeilen angegeben, werden alle Tabellenzeilen ausgewählt.

optStrColumns

Eine Zeichenkette zur Angabe, welche Tabellenspalten auszuwählen sind. Die Liste der Spaltennamen sollte durch Kommas getrennt sein.

Dieser Parameter ist optional. Werden keine Spalten angegeben, werden alle Tabellenspalten ausgewählt.

optStrTags

Eine Zeichenkette zur Angabe der Projekt-Tags, auf die die Spalten abgebildet werden. Die Liste der Tag-Namen sollte durch Kommas getrennt sein und die gleiche Reihenfolge haben wie die von **optStrColumns** festgelegten Spalten. Wenn der Cursor durch die Ergebnisreihe bewegt wird, werden die Werte in der aktuellen Zeile in diese Tags kopiert.

Dieser Parameter ist optional. Sind keine Tags angegeben, werden keine Werte kopiert.

optStrOrder

Die Sortierreihenfolge für die Zeilen. Dies entspricht dem SQL-Satz ORDER BY und die Zeichenkette sollte der gleichen Syntax folgen.

Dieser Parameter ist optional. Ist keine Reihenfolge angegeben, bleiben die Zeilen in der Standardreihenfolge der Tabelle.

optStrErrorTag

Der Name eines String-Tags, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.

Dieser Parameter ist optional.



Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert

Gibt einen numerischen Wert als Darstellung des Cursor-Handles aus. Bei Fehlern wird eine negative Zahl ausgegeben.



Diese Funktion ist gleich der Anweisung SQL SELECT, allerdings unterteilt sie die Sätze der Anweisung in separate Funktionsparameter. Wenn Sie sich mit SQL auskennen und eine eigene SELECT-Anweisung komplett selbst erstellen möchten, kann stattdessen DBCursorOpenSQL, Seite 781 verwendet werden.

Siehe auch DBCursorClose, Seite 773.

Beispiele

Verwendung in einem Berechnungsarbeitsblatt:

Tag-Name	Ausdruck
nCursor	DBCursorOpen("DB1", "Table1", "Column1 > 3", "Column1, Column2", "Tag1, Tag2", "Column1, Column2 DESC", "TagError") // Opens Table1 of DB1 and selects all rows where Column1 has a value greater than 3. Column1 is mapped to Tag1, and Column2 is mapped to Tag2. Rows are ordered first by Column1, then by Column2, in descending order. Error messages are written to TagError.

19.28.10 DBCursorOpenSQL

Wählt eine Reihe von Zeilen und Spalten aus einer Datenbanktabelle aus, initialisiert den Cursor in der ersten Zeile der Ergebnisreihe, kopiert die Zeilenwerte in abgebildete Tags und gibt dann einen Cursor-Handle aus, der von anderen DB-/ERP-Funktionen referenziert werden kann. (Diese Funktion entspricht einer SQL-Anweisung SELECT.)



Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Befehlen vertraut machen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DBCursorOpenSQL	Datenbank/ERP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DBCursorOpenSQL( strDBConn, strSQL, optStrTags, "optStrErrorTag" )
```

Eingaben

strDBConn

Name der Datenbankverbindung. Verbindungen sind im Ordner Datenbank/ERP, Seite 371 konfiguriert.

strSQL

Eine Zeichenkette als vollständige, syntaktisch korrekte SQL-Anweisung SELECT.

☞ In SQL werden geschweifte Klammern ("{}") üblicherweise zum Einschließen eines Ausdrucks verwendet, der vor Ausführung der gesamten SQL-Anweisung ausgewertet werden muss. Zum Beispiel:

```
SELECT * INTO inmates FROM OPENROWSET
('MSDASQL','Driver={Microsoft Text Driver (*.txt; *.csv)};
DEFAULTDIR=C:\;Extensions=CSV;
','SELECT * FROM flat.csv')
```

In "WinStudio" werden geschweifte Klammern allerdings zum Referenzieren von Tags in Textfeldern, die normalerweise nicht ausgewertet werden, verwendet (z.B. in der Beschriftung eines Schaltflächenobjekts). Wird eine SQL-Anweisung mit solch einem Ausdruck an **DBCursorOpenSQL** gegeben, versucht das Projekt, den Ausdruck als Tag-Bezug auszuwerten und die Funktion schlägt fehl. Um eine SQL-Anweisung so zu geben, dass das Projekt den Ausdruck korrekt auswerten kann, erstellen Sie einen neuen String-Tag mit dem Text des Ausdrucks und referenzieren dann den Tag in der SQL-Anweisung. Zum Beispiel:

```
$AuxTag = "{Microsoft Text Driver (*.txt; *.csv)}"
$DBCursorOpenSQL( "inmates", "SELECT * INTO inmates FROM
OPENROWSET ('MSDASQL','Driver={AuxTag};
DEFAULTDIR=C:\;Extensions=CSV;
','SELECT * FROM flat.csv')")
```

optStrTags

Eine Zeichenkette zur Auflistung der Projekt-Tags, auf die die Spalten abgebildet werden. Die Liste der Tag-Namen sollte durch Kommas getrennt sein und die gleiche Reihenfolge haben wie die von dem Satz WHERE von **StrSQL** festgelegten Spalten. Wenn der Cursor durch die Ergebnisreihe bewegt wird, werden die Werte in der aktuellen Zeile in diese Tags kopiert.

Dieser Parameter ist optional. Sind keine Tags angegeben, werden keine Werte kopiert.

optStrErrorTag

Der Name eines String-Tags, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.

Dieser Parameter ist optional.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert

Gibt den Cursor-Handle als numerischen Wert zurück. Bei Fehlern wird eine negative Zahl ausgegeben.

Siehe auch DBCursorClose, Seite 773.

Beispiele Verwendung in einem Mathematik-Arbeitsblatt:

Tag-Name	Ausdruck
nCursor	DBCursorOpenSQL("DB1", "SELECT Column1, Column2 FROM Table1 WHERE Column1 > 3 ORDER BY Column1, Column2 DESC", "Tag1, Tag2")

19.28.11 DBCursorPrevious

Bewegt den Cursor zur vorhergehenden Zeile in der Ergebnisreihe und kopiert die Werte dieser Zeilen in die abgebildeten Tags. Wenn keine vorhergehende Zeile vorhanden ist – d.h. wenn die aktuelle Zeile die erste ist – gibt die Funktion einen Fehlercode aus und lässt die abgebildeten Tags unverändert.

Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Befehlen vertraut machen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DBCursorPrevious	Datenbank/ERP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DBCursorPrevious( numCur, "optStrErrorTag" )
```

Eingaben

numCur

Der Cursor-Handle für die Ergebnisreihe. Der Cursor-Handle wird von DBCursorOpen, Seite 779 oder DBCursorOpenSQL, Seite 781 zurückgegeben.

optStrErrorTag

Der Name eines String-Tags, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.

Dieser Parameter ist optional.

Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert Bei fehlerfreier Ausführung wird 0 zurückgegeben. Bei Fehlern wird eine negative Zahl zurückgegeben.

Beispiele Verwendung in einem Mathematik-Arbeitsblatt:

Tag-Name	Ausdruck
ErrorCode	DBCursorPrevious(nCursor)

19.28.12 DBCursorRowCount

Ruft die Gesamtzahl der Zeilen in einer SQL-Ergebnisreihe ab.

Siehe auch DBCursorColumnCount(), Seite 774.

Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Befehlen vertraut machen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DBCursorRow-Count	Datenbank/ERP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DBCursorRowCount( numCur, "optStrErrorTag" )
```

Eingaben

numCur

Der Cursor-Handle für die Ergebnisreihe. Der Cursor-Handle wird von DBCursorOpen, Seite 779 oder DBCursorOpenSQL, Seite 781 zurückgegeben.

optStrErrorTag

Der Name eines String-Tags, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.

Dieser Parameter ist optional.

Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert

Gibt die Anzahl der Zeilen zurück. Bei Fehlern wird eine negative Zahl zurückgegeben.

Siehe auch DBCursorColumnCount, Seite 774.

Beispiele

Verwendung in einem Mathematik-Arbeitsblatt:

Tag-Name	Ausdruck
nRow-Count	DBCursorRowCount(nCursor)

19.28.13 DBDelete

Löscht ausgewählte Zeilen aus einer Datenbanktabelle. (Diese Funktion entspricht einer SQL-Anweisung DELETE.)

Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Befehlen vertraut machen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DBDelete	Datenbank/ERP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DBDelete(strDBConn, strTable, strCondition, "optStrErrorTag")
```

Eingaben

strDBConn

Name der Datenbankverbindung. Verbindungen sind im Ordner Datenbank/ERP, Seite 371 konfiguriert.

strTable

Name der Datenbanktabelle.

strCondition

Eine Zeichenkette zur Angabe, welche Tabellenzeilen auszuwählen sind. Dies entspricht dem SQL-Satz WHERE und die Zeichenkette sollte der gleichen Syntax folgen.

Zum Löschen aller Zeilen in der Tabelle geben Sie als Bedingungsanweisung ein Leerzeichen ein (" ").

optStrErrorTag

Der Name eines String-Tags, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.

Dieser Parameter ist optional.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert Gibt die Anzahl der aus der Tabelle gelöschten Zeilen zurück. Bei Fehlern wird eine negative Zahl ausgegeben.

Beispiele Verwendung in einem Mathematik-Arbeitsblatt:

Tag-Name	Ausdruck
nRowsDeleted	DBDelete("DB1", "Table1", "Column1 > 1000", "TagError") // Deletes all rows in Table1 where the value of Column1 is greater than 1000. The returned value (i.e., the number of rows deleted) is written to **TagError.**
Tag	DBDelete("DB1", "Table1", " ") // Deletes all rows of Table1.

19.28.14 DBExecute

"DBExecute" ist eine interne Skript-Funktion zur Ausführung einer benutzerdefinierten SQL-Anweisung auf einer externen Datenbank. Wenn die Anweisung eine Abfrage ist (z.B. SELECT), werden die Datenbankwerte in festgelegte Array-Tags kopiert.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DBExecute	Datenbank/ERP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DBExecute(strDBConn,strSQL{ | ,optStrTags,optNumMaxRows{ | ,optStrErrorTag }})
```

Eingaben

strDBConn

Name der Datenbankverbindung. Verbindungen sind im Ordner **Datenbank/ERP** im "Projektexplorer" konfiguriert.

strSQL

Eine vollständige, syntaktisch korrekte SQL-Anweisung.



In SQL werden geschweifte Klammern ("{}") üblicherweise zum Einschließen eines Ausdrucks verwendet, der vor Ausführung der gesamten SQL-Anweisung ausgewertet werden muss. Zum Beispiel:

```
SELECT * INTO inmates FROM OPENROWSET
('MSDASQL','Driver={Microsoft Text Driver (*.txt; *.csv)};
DEFAULTDIR=C:\;Extensions=CSV;','SELECT * FROM flat.csv')
```

In "WinStudio" werden geschweifte Klammern allerdings zum Referenzieren von Projekt-Tags in Textfeldern, die normalerweise nicht ausgewertet werden, verwendet (z.B. in der Beschriftung eines Schaltflächenobjekts). Wird eine SQL-Anweisung mit solch einem Ausdruck an "DBExecute" gegeben, versucht das Projekt, den Ausdruck als Tag-Bezug auszuwerten und die Funktion schlägt fehl. Um eine SQL-Anweisung so zu übergeben, dass das Projekt den Ausdruck korrekt auswerten kann, erstellen Sie einen neuen String-Tag mit dem Text des Ausdrucks und referenzieren dann diesen Tag in der SQL-Anweisung. Zum Beispiel:

```
AuxTag = "{Microsoft Text Driver (*.txt; *.csv)}"
DBExecute( "inmates", "SELECT * INTO inmates FROM
OPENROWSET ('MSDASQL','Driver={AuxTag};
DEFAULTDIR=C:\;Extensions=CSV;
','SELECT * FROM flat.csv')")
```

optStrTags

Eine komma-getrennte Liste der Namen eines Array-Tags in Ihrem Projekt, auf die die Spalten einer SQL-SELECT-Ergebnisreihe abgebildet werden. Die Datenbankwerte werden in diese Array-Tags kopiert, wobei die erste Zeile der Ergebnisreihe in Array-Index 0 kopiert wird. Es ist sicherzustellen, dass die Arrays groß genug sind, um alle Zeilen der Ergebnisreihe aufzunehmen.

Dieser Parameter wird nur benötigt, wenn **strSQL** eine SQL-SELECT-Anweisung enthält. Für alle anderen Arten von Anweisungen wird dieser Parameter ignoriert und kann ausgelassen werden. Wenn allerdings die Syntax der Funktion zur Weiterführung zu **optStrErrorTag** aufrechterhalten werden muss, wird diesem Parameter eine leere Zeichenkette zugewiesen ("").

optNumMaxRows

Die maximale Anzahl der aus einer SQL-SELECT-Ergebnisreihe zu kopierenden Zeilen. In den meisten Fällen wird zum Kopieren aller Zeilen eine Zahl eingegeben, die größer als die erwartete Anzahl der Zeilen in der Ergebnisreihe ist.

Dieser Parameter wird nur benötigt, wenn **strSQL** eine SQL-SELECT-Anweisung enthält. Für alle anderen Arten von Anweisungen wird dieser Parameter ignoriert und kann ausgelassen werden. Wenn allerdings die Syntax der Funktion zur Weiterführung zu **optStrErrorTag** aufrechterhalten werden muss, wird diesem Parameter der Wert 0 zugewiesen.

optStrErrorTag

Der Name eines String-Tags, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.

Dieser Parameter ist optional.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert

Diese Funktion gibt die Gesamtzahl der von der SQL-Anweisung betroffenen Zeilen zurück. Wenn ein Fehler auftritt, wird eine negative Zahl ausgegeben.

Bitte beachten Sie, dass dies der von der Funktion selbst zurückgegebene Wert ist. Im Falle einer SQL-SELECT-Anweisung werden die Datenbankwerte in die durch **optStrTags** festgelegten Array-Tags kopiert.

☞ Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Anweisungen vertraut machen.

Beispiele

Programm:

```
DBExecute( "DB1", "INSERT INTO Table1(Column1,Column2) values(1,1)" )

DBExecute("DB1","SELECT max(Column1),max(Column2) FROM Table1","Tag1,Tag2",1,"
```

19.28.15 DBInsert

Fügt eine neue Zeile in eine Datenbanktabelle ein. (Diese Funktion entspricht einer SQL-Anweisung INSERT.)

☞ Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Befehlen vertraut machen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DBInsert	Datenbank/ERP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DBInsert( strDBConn, strTable, strValues, optStrColumns, "optStrErrorTag" )
```

Eingaben

strDBConn

Name der Datenbankverbindung. Verbindungen sind im Ordner **Datenbank/ERP** im "Projektexplorer" konfiguriert.

strTable

Name der Datenbanktabelle.

strValues

Eine Zeichenkette mit den Werten, die in die neue Zeile geschrieben werden sollen. Diese Werteliste sollte komma-getrennt sein und String-Werte müssen in Apostrophe gesetzt werden.

optStrColumns

Ein Zeichenkette mit einer Liste der Spalten, in die die Werte geschrieben werden. Die Liste der Spalten-Namen sollte durch Kommas getrennt sein und die gleiche Reihenfolge haben wie die von **strValues** festgelegten Werte.

Dieser Parameter ist optional. Sind keine Spalten angegeben, werden die Werte in die Standardspaltenreihenfolge der Datenbanktabelle geschrieben.

optStrErrorTag

Der Name eines String-Tags, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.

Dieser Parameter ist optional.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert Bei Fehlern wird eine negative Zahl ausgegeben. Andernfalls wird 1 ausgegeben.

Beispiele Verwendung in einem Mathematik-Arbeitsblatt:

Tag-Name	Ausdruck
nErrorCode	DBInsert("DB1", "Table1", "1, 'one'", "Column1, Column2")

19.28.16 DBSelect

"DBSelect" ist eine interne Skript-Funktion, die eine Ergebnisreihe aus einer externen Datenbank auswählt (ähnlich einer SQL-Anweisung SELECT), die Spalten auf Array-Tags in Ihrem Projekt abbildet und die Werte aus der Ergebnisreihe in die Array-Tags kopiert.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DBSelect	Datenbank/ERP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DBSelect(strDBConn,strTable,strTags,strColumns,strCondition,strOrder{ | ,optNumMaxRows{ | ,optStrErrorTag}})
```

Eingaben

strDBConn

Name der Datenbankverbindung. Verbindungen sind im Ordner **Datenbank/ERP** im "Projektexplorer" konfiguriert.

strTable

Der Name der Datenbanktabelle, aus der die Ergebnisreihe ausgewählt wird.

strTags

Eine komma-getrennte Liste der Namen eines Array-Tags in Ihrem Projekt, auf die die Spalten der Datenbanktabelle abgebildet werden. Die Datenbankwerte werden in diese Array-Tags kopiert, wobei die erste Zeile der Ergebnisreihe in Index 0 kopiert wird. Es ist sicherzustellen, dass die Arrays groß genug sind, um alle Zeilen der Ergebnisreihe aufzunehmen.

strColumns

Eine komma-getrennte Liste mit den aus der Datenbanktabelle auszuwählenden Spalten. Die Reihenfolge der Liste sollte der Liste in **strTags** entsprechen.

Um alle Spalten der Tabelle in ihrer ursprünglichen Reihenfolge auszuwählen, muss diesem Parameter eine leere Zeichenkette ("""") zugewiesen werden.

strCondition

Eine Anweisung zur Angabe, welche Tabellenzeilen aus der Datenbank auszuwählen sind. Dies entspricht dem SQL-Satz WHERE und muss der gleichen Syntax folgen.

Um alle Zeilen in der Tabelle auszuwählen, wird diesem Parameter eine leere Zeichenkette ("") zugewiesen.

strOrder

Eine Anweisung zur Festlegung der Sortierreihenfolge für die Zeilen. Dies entspricht dem SQL-Satz ORDER BY und muss der gleichen Syntax folgen.

Um die Zeilen in ihrer ursprünglichen Reihenfolge zu belassen, wird diesem Parameter eine leere Zeichenkette ("") zugewiesen.

optNumMaxRows

Die maximale Anzahl der zu kopierenden Zeilen. In den meisten Fällen wird zum Kopieren aller Zeilen eine Zahl eingegeben, die größer als die erwartete Anzahl der Zeilen in der Ergebnisreihe ist.

Dieser Parameter ist optional; wird kein Wert angegeben, wird nur die erste Zeile der Ergebnisreihe kopiert.

optStrErrorTag

Der Name eines String-Tags, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.

Dieser Parameter ist optional.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert

Diese Funktion gibt die Gesamtzahl der Zeilen in der Ergebnisreihe zurück. Wenn ein Fehler auftritt, wird eine negative Zahl ausgegeben.

Es ist zu beachten, dass dies der von der Funktion selbst zurückgegebene Wert ist; die Datenbankwerte werden in die durch **strTags** festgelegten Array-Tags kopiert.

☞ Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Anweisungen vertraut machen.

Beispiele

Programm:

```
DBSelect( "DB1", "Table1", "Array1,Array2", "Column1,Column2", "", "" )
DBSelect("DB1","Table1","Array1,Array2","Column1,Column2","Column2 < Column1","Column1",4,"TagError")
```

19.28.17 DBUpdate

Wählt eine Ergebnisreihe aus und schreibt den gleichen Wert in alle Zeilen einer festgelegten Spalte. (Diese Funktion entspricht einer SQL-Anweisung UPDATE.)

☞ Diese Funktion emuliert Datenbankoperationen in SQL (Structured Query Language). Vor der Verwendung dieser Funktion sollten Sie sich mit Struktur und Ausführung von SQL-Befehlen vertraut machen.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
DBUpdate	Datenbank/ERP	synchron	unterstützt	unterstützt	unterstützt

Syntax

```
DBUpdate( strDBConn, strTable, strValues, strColumns, optStrCondition, "optStrErrorTag" )
```

Eingaben

strDBConn

Name der Datenbankverbindung. Die verfügbaren Verbindungen werden im Dialog "Datenbankverbindungen" angezeigt, der über das Kontextmenü des Ordners **Tasks ▸ Externe Datenbanken** geöffnet werden kann.

strTable

Name der Datenbanktabelle.

strValues

Eine Zeichenkette mit den Werten, die in die Spalten geschrieben werden sollen. Diese Werteliste sollte komma-getrennt sein und String-Werte müssen in Apostrophe gesetzt werden.

strColumns

Ein Zeichenkette mit einer Liste der Spalten, in die die Werte geschrieben werden. Die Liste der Spalten-Namen sollte durch Kommas getrennt sein und die gleiche Reihenfolge haben wie die von **strValues** festgelegten Werte.

optStrCondition

Eine Zeichenkette zur Angabe, welche Tabellenzeilen auszuwählen sind. Dies entspricht dem SQL-Satz WHERE und die Zeichenkette sollte der gleichen Syntax folgen.

Dieser Parameter ist optional. Werden keine Zeilen angegeben, werden alle Tabellenzeilen ausgewählt.

optStrErrorTag

Der Name eines String-Tags, der beim Auftreten von Fehlern während der Laufzeit detaillierte Fehlermeldungen erhält.

Dieser Parameter ist optional.

☞ Der Tag-Name muss wie im Syntaxdiagramm gezeigt in Anführungszeichen stehen, andernfalls versucht das Projekt, den Wert des angegebenen Tags abzurufen.

Rückgabewert Gibt die Anzahl der aktualisieren Zeilen zurück. Bei Fehlern wird eine negative Zahl ausgegeben.

Beispiele Verwendung in einem Berechnungsarbeitsblatt:

Tag-Name	Ausdruck
Tag	DBUpdate("DB1", "Table1", "'X'", "Column2", "Column1 = 1", "TagError") // In Table1 of DB1, for all rows where Column1 equals 1, writes "X" to Column2.

19.28.18 SyncAlarm

Synchronisiert die Alarm-Datenbank.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SyncAlarm	Datenbank/ERP	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
SyncAlarm( optStrStartDate, optStrEndDate )
```

Eingaben

optStrStartDate

Zeichenkette mit dem Startdatum. Ist dieser Parameter nicht angegeben, wird das aktuelle Datum verwendet.

optStrEndDate

Zeichenkette mit dem Enddatum. Wird dieser Parameter nicht angegeben, verwendet die Funktion den gleichen Wert wie das Startdatum.

Rückgabewert

Wert	Beschreibung
1	Synchronisation konnte nicht gestartet werden; die Datenbank wird wahrscheinlich gerade synchronisiert.
0	Erfolgreich
-1	Ungültige Gruppennummer
-2	Format ist nicht auf „Datenbank“ eingestellt.
-4	Angegebenes Startdatum ist ungültig.
-5	Angegebenes Enddatum ist ungültig.
-6	Startdatum ist größer als Enddatum.

☞ Diese Funktion wird asynchron ausgeführt, damit sie nicht das Ergebnis des Synchronisation zurückgibt. Um diese Information zu erhalten, verwenden Sie die Funktion SyncAlarmStatus, Seite 792.

Beispiele

Tag-Name	Ausdruck
Tag	SyncAlarm() // Synchronizes the database using the current date
Tag	SyncAlarm("10/20/2004") // Synchronizes the database only for the day 10/20/2004
Tag	SyncAlarm("10/20/2004", "10/28/2004") // Synchronizes the database from 10/20/2004 to 10/28/2004

19.28.19 SyncAlarmStatus

Gibt den Status einer zuvor aufgerufenen Funktion "SyncAlarm" zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SyncAlarmStatus	Datenbank/ERP	synchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
SyncAlarmStatus()
```

Diese Funktion hat keine Parameter.

Rückgabewert

Wert	Beschreibung
3	Synchronisation abgeschlossen.
2	Synchronisation fehlgeschlagen
1	Synchronisation wird ausgeführt
0	Es wird keine Synchronisation ausgeführt.
-1	Format ist nicht auf „Datenbank“ eingestellt.

Beispiele

Tag-Name	Ausdruck
Tag	SyncAlarmStatus()

19.28.20 SyncEvent

Synchronisiert die Ereignis-Datenbank.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SyncEvent	Datenbank/ERP	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
SyncEvent( optStrStartDate, optStrEndDate )
```

Eingaben

optStrStartDate

Zeichenkette mit dem Startdatum. Ist dieser Parameter nicht angegeben, wird das aktuelle Datum verwendet.

optStrEndDate

Zeichenkette mit dem Enddatum. Wird dieser Parameter nicht angegeben, verwendet die Funktion den gleichen Wert wie das Startdatum.

Rückgabewert

Wert	Beschreibung
1	Synchronisation konnte nicht gestartet werden; die Datenbank wird wahrscheinlich gerade synchronisiert.
0	Erfolgreich
-1	Ungültige Gruppennummer
-2	Format ist nicht auf „Datenbank“ eingestellt.
-4	Angegebenes Startdatum ist ungültig.
-5	Angegebenes Enddatum ist ungültig.
-6	Startdatum ist größer als Enddatum.



Diese Funktion wird asynchron ausgeführt, damit sie nicht das Ergebnis des Synchronisation zurückgibt. Um diese Information zu erhalten, verwenden Sie die Funktion SyncEventStatus, Seite 793.

Beispiele

Tag-Name	Ausdruck
Tag	SyncEvent() // Synchronizes the database using the current date
Tag	SyncEvent("10/20/2004") // Synchronizes the database only for the day 10/20/2004
Tag	SyncEvent("10/20/2004", "10/28/2004") // Synchronizes the database from 10/20/2004 to 10/28/2004

19.28.21 SyncEventStatus

Gibt den Status einer zuvor aufgerufenen Funktion "SyncEvent" zurück.

Anhang: WinStudio-interne Skriptsprache

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SyncEventStatus	Datenbank/ERP	synchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
SyncEventStatus()
```

Diese Funktion hat keine Parameter.

Rückgabewert

Wert	Beschreibung
3	Synchronisation abgeschlossen.
2	Synchronisation fehlgeschlagen
1	Synchronisation wird ausgeführt
0	Es wird keine Synchronisation ausgeführt.
-1	Format ist nicht auf „Datenbank“ eingestellt.

Beispiele

Tag-Name	Ausdruck
Tag	SyncEventStatus()

19.28.22 SyncTrend

Synchronisiert die Trend-Datenbank.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
Avg	Datenbank/ERP	asynchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
SyncTrend( numGroup, optStrStartDate, optStrEndDate )
```

Eingaben

numGroup

Nummer der Trendgruppe (Nummer des Trend-Arbeitsblatts).

optStrStartDate

Zeichenkette mit dem Startdatum. Ist dieser Parameter nicht angegeben, wird das aktuelle Datum verwendet.

optStrEndDate

Zeichenkette mit dem Enddatum. Wird dieser Parameter nicht angegeben, verwendet die Funktion den gleichen Wert wie das Startdatum.

Rückgabewert

Wert	Beschreibung
1	Synchronisation konnte nicht gestartet werden; die Datenbank wird wahrscheinlich gerade synchronisiert.
0	Erfolgreich
-1	Ungültige Gruppennummer
-2	Format ist nicht auf „Datenbank“ eingestellt.

Wert	Beschreibung
-4	Angegebenes Startdatum ist ungültig.
-5	Angegebenes Enddatum ist ungültig.
-6	Startdatum ist größer als Enddatum.



Diese Funktion wird asynchron ausgeführt, damit sie nicht das Ergebnis des Synchronisation zurückgibt. Um diese Information zu erhalten, verwenden Sie die Funktion SyncTrendStatus, Seite 795.

Beispiele

Tag-Name	Ausdruck
Tag	SyncTrend(1) // Synchronizes the group 1 database using the current date
Tag	SyncTrend(1, "10/20/2004") // Synchronizes the group 1 database only for the day 10/20/2004
Tag	SyncTrend(1, "10/20/2004", "10/28/2004") // Synchronizes the group 1 database from 10/20/2004 to 10/28/2004

19.28.23 SyncTrendStatus

Gibt den Status einer zuvor aufgerufenen Funktion "SyncTrend" zurück.

Funktion	Gruppe	Ausführung	Windows	Embedded	Thin Client
SyncTrendStatus	Datenbank/ERP	synchron	unterstützt	unterstützt	auf dem Server ausgeführt

Syntax

```
SyncTrendStatus(numGroup)
```

Eingaben

numGroup
Nummer der Trendgruppe (Trend-Arbeitsblatt).

Rückgabewert

Wert	Beschreibung
3	Synchronisation abgeschlossen.
2	Synchronisation fehlgeschlagen
1	Synchronisation wird ausgeführt
0	Es wird keine Synchronisation ausgeführt.
-1	Format ist nicht auf „Datenbank“ eingestellt.

Beispiele

Tag-Name	Ausdruck
Tag	SyncTrendStatus(1)

20 Anhang: VBScript-Sprache

20.1 Übersicht VBScript

VBScript ist eine einfache, standardisierte und flexible Skriptsprache, mit der logische Operationen und Algorithmen innerhalb Ihres Projekts implementiert werden können.

"WinStudio" implementiert Visual Basic Scripting Edition 5.5 oder höher.

Vorteile der Sprache "VBScript":

- Syntax, Operatoren und Funktionen.
- Die Fähigkeit zum Erstellen neuer Variablen und Prozeduren (Funktionen und/oder Subroutinen).
- Zugriff auf Eigenschaften, Methoden und/oder Ereignisse aus COM-Objekten, einschließlich ActiveX-Steuerelemente.
- Die Fähigkeit zum Ausführen der logischen Operationen auf jeder Plattform, die VBScript unterstützt, einschließlich PCs auf Basis von Microsoft Windows (als "WinStudio"-Serverstation ausgeführt), Microsoft-Windows-Embedded-Geräten (über CEView) und Internet Explorer (über den Thin Client).



Wenn Sie nicht sicher sind, ob das auf Ihrem Gerät geladene Image VBScript unterstützt, wenden Sie sich bitte an den Hardwarehersteller. Der Hardwarehersteller muss die Unterstützung für VBScript auf dem Windows-CE-Gerät freigeben, damit CEView die in der VBScript-Sprache auf dem Gerät konfigurierten Skripte ausführen kann.

Das Ziel dieser Dokumentation ist die Bereitstellung eines Überblicks für die Verwendung von VBScript-Kommandos in"Rexroth WinStudio". Außerdem kann sie zum schnellen Nachschlagen für die am häufigsten verwendeten Merkmale der Sprache verwendet werden. Eine vollständige Beschreibung der Sprache sowie der Schnittstellen und Funktionen erhalten Sie von Microsoft. (Zum Zeitpunkt der Erstellung dieser Dokumentation stand die VBScript-Dokumentation direkt auf dem Microsoft Developer Network zur Verfügung. Allerdings befindet sich dieser Link außerhalb unserer Kontrolle und kann sich ohne Vorankündigung ändern.)

20.2 VBScript-Schnittstellen in der Software

20.2.1 VBScript-Schnittstellen in der Software, Allgemeines

Die folgende Tabelle zeigt eine Zusammenfassung der von "Rexroth WinStudio" unterstützen VBScript-Schnittstellen:

Schnittstelle	Bereich für Prozeduren und Variablen	Ausführung	Funktionalität
Globale Prozeduren	Grafiken und Tasks	-	Deklarieren von Prozeduren
Grafikskript	Nur Grafikskript-Schnittstelle	Server (Viewer) + Thin Clients	Deklarieren von Variablen Deklarieren von Prozeduren Ausführung
Bild-Skript	Bild, in dem das Skript konfiguriert ist	Server (Viewer) + Thin Clients	Deklarieren von Variablen Deklarieren von Prozeduren Ausführung

Anhang: VBScript-Sprache

Schnittstelle	Bereich für Prozeduren und Variablen	Ausführung	Funktionalität
Befehlsanimation	Objekt, in der das Skript konfiguriert ist	Server (Viewer) + Thin Clients	Deklarieren von Variablen Ausführung
ActiveX Ereignisse	Objekt, in der das Skript konfiguriert ist	Server (Viewer) + Thin Clients	Deklarieren von Variablen Ausführung
Start-Skript	Alle Skript-Blätter aus Tasks	Server (BGTask)	Deklarieren von Variablen Deklarieren von Prozeduren Ausführung
Skript-Gruppen	Nur Skript-Gruppen	Server (BGTask)	Deklarieren von Variablen Ausführung

Das folgende Bild zeigt den Bereich jeder VBScript-Schnittstelle und die Reihenfolge, in der diese von "WinStudio" ausgeführt werden:

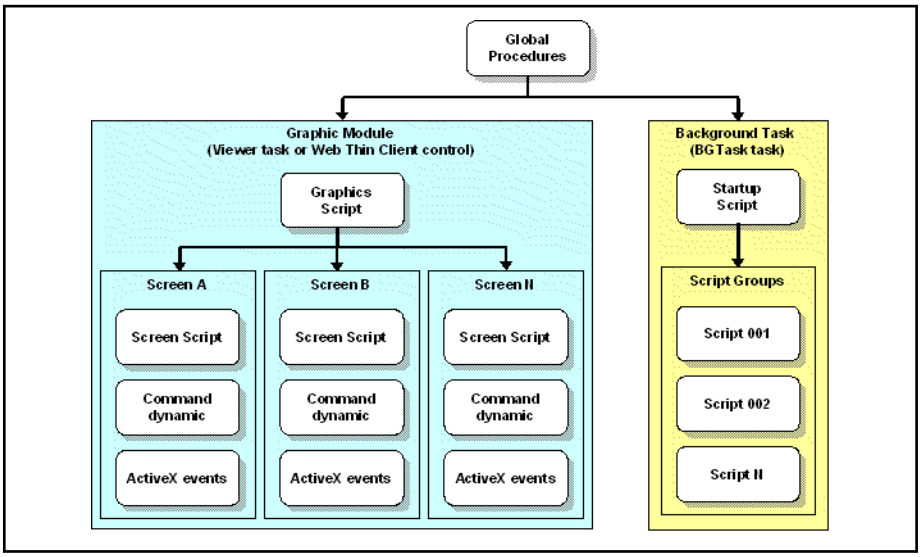


Abb.20-1: Bereich von VBScript-Schnittstellen

Die Darstellung zeigt, dass die globalen Prozeduren von Grafikmodul und Hintergrundtask gemeinsam verwendet werden. Allerdings stammen die anderen VBScript-Schnittstellen entweder vom Grafikmodul oder vom Hintergrundtask und es werden zwischen ihnen keine Variablen oder Prozeduren gemeinsam verwendet. Sie sind voneinander unabhängig.

Obwohl das Grafik-Skript von "WinStudio" vor den Bild-Skripten ausgeführt wird, stehen die in den Grafik-Skript-Schnittstellen deklarierten Prozeduren und Variablen für in den Bildern konfigurierten Skript-Schnittstellen NICHT zur Verfügung. Zum Implementieren von Prozeduren, die für alle Bilder verfügbar sein müssen, müssen Sie diese in den globalen Prozeduren deklarieren.

Wenn Sie Ihren Code in einer VBScript-Schnittstelle schreiben, können Sie auf jeden Tag aus der "WinStudio"-Datenbank oder jede Funktion aus der eingebauten Skriptsprache zugreifen, indem Sie wie in den unten gezeigten Beispielen das „$“-Zeichen vor den Namen des Tags oder der Funktion setzen:

Code in einer VBScript-Schnittstelle

```
$Time 'Returns the value of the tag Time from the tags database
$MyTag 'Returns the value of the tag MyTag from the tags database
$Open("main") 'Executes the Open() built-in function to open the "main" screen
```

Somit können Sie in VB-Skripten auf interne "WinStudio"-Funktionen, "WinStudio"-Tags, VBScript-Funktionen, VBScript-Variablen, ActiveX-Eigenschaften, Methoden oder Ereignisse zugreifen. Die "WinStudio"-Tags werden von allen Modulen aus "WinStudio" gemeinsam verwendet, einschließlich des Grafikmoduls und des Hintergrundtasks.

20.2.2 Globale Prozeduren

Arbeitsblätter im Ordner "Globale Prozeduren" können zur Erstellung einer Bibliothek mit VBScript-Funktionen und Subroutinen auf Seite 828 verwendet werden, deren Funktionen von jeder anderen Skriptschnittstelle in "WinStudio" aufgerufen werden können. Die hier deklarierten Prozeduren werden während der Laufzeit nie direkt ausgeführt; sie müssen ausdrücklich von einem anderen Skript aufgerufen werden.

HINWEIS

"WinStudio" hindert Sie nicht daran, zwei oder mehrere Funktionen mit dem gleichen Namen zu deklarieren.

(Dies schließt aus externen Dateien importierte Funktionen ein; siehe „Importieren von Funktionen aus einer externen Datei“ unten.) Wenn Sie dies allerdings tun, kann Ihr Projekt sich während der Laufzeit unerwartet verhalten. Stellen Sie sicher, dass alle Funktionen korrekt und eindeutig benannt sind.

Sie können beliebig viele Arbeitsblätter für globale Prozeduren erstellen und jedes Arbeitsblatt kann beliebig viele Funktionen und Subroutinen enthalten.

Zum Erstellen von Funktionen in einem Arbeitsblatt für globale Prozeduren gehen Sie folgendermaßen vor:

1. Öffnen Sie das Standardarbeitsblatt **Hauptprozeduren** im Ordner **WinStudio ▸ Tasks ▸ Global Prozeduren**.

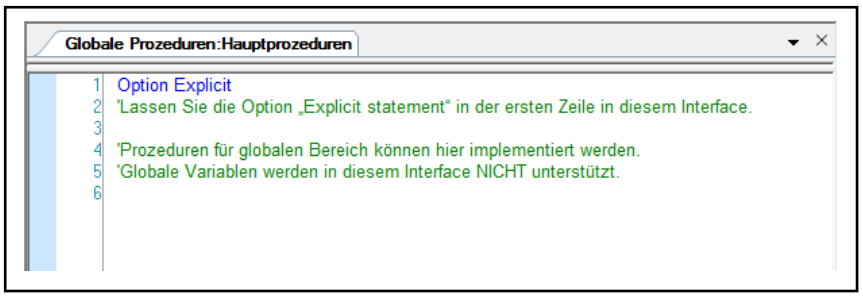


Abb.20-2: Arbeitblatt "Globale Prozeduren - Hauptprozeduren"

   In diesem Arbeitsblatt können Sie beliebige Prozeduren deklarieren. Dieses Arbeitsblatt ist fester Bestandteil jedes Projekts und kann nicht gelöscht werden. Zur Strukturierung Ihrer Prozeduren können Sie jedoch zusätzliche Arbeitsblätter wie im folgenden Schritt beschrieben erstellen.

2. Öffnen Sie das Kontextmenü des Ordners **Globale Prozeduren** und wählen Sie den Eintrag "Neues Skript".

   Führen Sie einen Doppelklick auf das eingefügte Arbeitsblatt aus, um dieses im Editor zu öffnen.



Abb.20-3: Ein neu eingefügtes Arbeitsblatt Globale Prozeduren

3. Deklarieren Sie Ihre Funktion und Subroutinen in der Schnittstelle.

Zum Beispiel:

```
'Procedures with global scope can be implemented here
'Global variables are NOT supported in this interface

Sub MyMessage(message)
  MsgBox(message,0)
End Sub

Function MyAdd(number1, number2)
  MyAdd = number1 + number2
  Call MyMessage("The sum is" & MyAdd & ".")
End Function
```

Sie können in jeder Prozedur lokale Variablen deklarieren, aber Sie können keine globalen Variablen in dieser Schnittstelle deklarieren. In den meisten Fällen sollten statt dessen Tags verwendet werden.

4. Speichern Sie Ihre Änderungen.

Passwortschutz für Arbeitsblätter

Alle zusätzlich erstellten Arbeitsblätter für Globale Prozeduren können Sie mit einem Passwort schützen, um zu verhindern, dass sie von anderen Benutzern bearbeitet oder analysiert werden können. Zum Schützen eines Arbeitsblatts gehen Sie folgendermaßen vor:

1. Führen Sie im Ordner "Globale Prozeduren" einen Rechtsklick auf das gewünschte Arbeitsblatt aus und wählen Sie den Eintrag **Passwortschutz** aus dem Kontextmenü. Der Dialog "Passwortschutz" wird angezeigt.
2. Geben Sie Ihr Passwort in das Feld **Neues Passwort** ein.
3. Geben Sie Ihr Passwort erneut in das Feld **Passwort bestätigen** ein.
4. Klicken Sie auf **OK**.

Anschließend werden Sie bei jedem Öffnen des Arbeitsblatts aufgefordert, Ihr Passwort einzugeben.

20.2.3 Grafikmodul

Grafikskript

Das Arbeitsblatt "Grafikskript " dient zur Deklaration von Funktionen, die bei bestimmten Ereignissen (s.u.) des Graphikmoduls ausgeführt werden sollen. Um das Grafikskript zu öffnen, führen Sie einen Doppelklick auf das Element "Grafik-Skript" im Ordner "Skripte" aus.

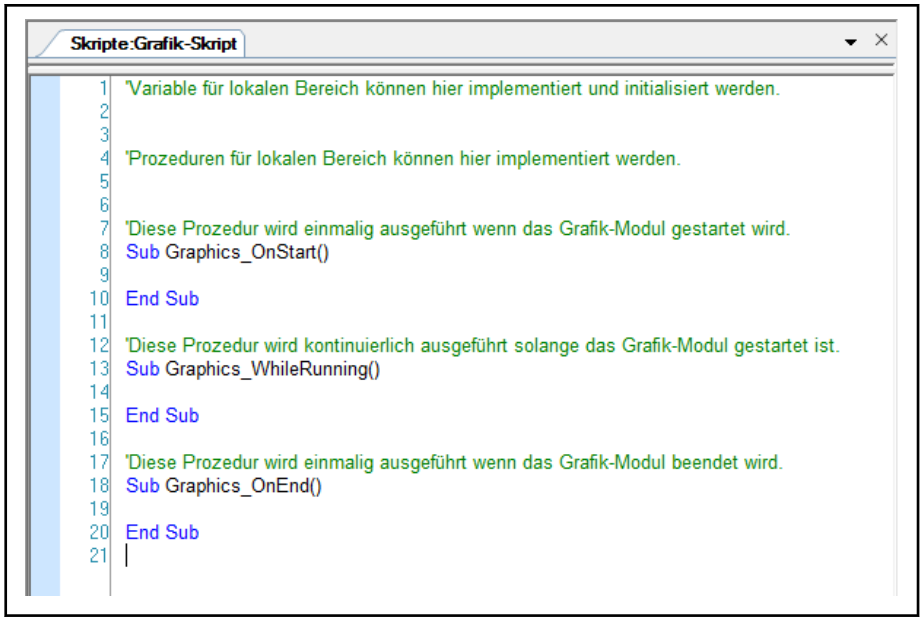


Abb.20-4: Projektexplorer: Schnittstelle Grafikskript

Dieses Arbeitblatt enthält bereits vordefinierte Funktionen, die bei folgenden Ereignissen ausgeführt werden:

- **Graphics_OnStart()**: Der innerhalb dieser Subroutine konfigurierte Code wird automatisch beim Start des Grafikmoduls einmalig ausgeführt. Diese Schnittstelle ist hilfreich zur Initialisierung von Variablen oder zum Ausführen von Logiken, die einmalig beim Start der Laufzeitsystems ausgeführt werden müssen.
- **Graphics_WhileRunning()**: Der innerhalb dieser Subroutine konfigurierte Code wird automatisch während der Ausführung des Grafikmoduls kontinuierlich ausgeführt. Die Geschwindigkeit, mit der diese Subroutine aufgerufen wird, hängt von der Leistungsfähigkeit der Plattform, auf dem das Projekt ausgeführt wird, ab.
- **Graphics_OnEnd()**: Der innerhalb dieser Subroutine konfigurierte Code wird automatisch beim Schließen des Grafikmoduls einmalig ausgeführt.
- **Graphics_OnScreenResize(width, height)**: Der innerhalb dieser Subroutine konfigurierte Code wird automatisch beim Verändern der Bildauflösung der Laufzeitstation einmalig ausgeführt. Die neue Höhe und Breite des Bildes (in Pixel) werden als Parameter an die Subroutine weitergegeben.

☞ Die Subroutine **Graphics_OnScreenResize()** ist für Projekte bestimmt, die auf CEView- oder Windows-Embedded-Geräten ausgeführt werden, die zwischen Hochformat- und Querformatanzeige wechseln können. Sie wird für Projekte, die auf einem Windows-PC oder im Thin Client laufen, "nicht" unterstützt.

☞ Die Namen der vordefinierten Subroutinen dürfen nicht verändert werden. Andernfalls kann das System diese nicht automatisch ausführen.

Beispiel:

```
'Variables with local scope can be declared and initialized here
Dim MyDate
MyDate = Date()
```

```
Dim MyValue
MyValue = 100

'Procedures with local scope can be implemented here
Function MyNewProcedure(nCount)
    MyNewProcedure = nCount + 1
End Function

Function AreaRec(side1, side2)
  AreaRec = side1 * side2
End Function

Sub CheckHiLimit(myValue, myHiLimit)
  If myValue > myHiLimit Then
    MsgBox("Value out of range")
  End If
End Sub

'This procedure is executed just once when the graphic module is started
Sub Graphics_OnStart()
  MsgBox("Welcome to the system!")
End Sub

'This procedure is executed continuously while the graphic module is running
Sub Graphics_WhileRunning()
  If $UserName = "Guest" Then
    $MyFlag = 0
  End If
End Sub

'This procedure is executed just once when the graphic module is closed
Sub Graphics_OnEnd()
  $LogOff()
End Sub
```

Für die Ausführung des Grafikskripts sind folgende Hinweise zu beachten:

Auf dem Server (auf dem "Rexroth WinStudio" oder CEView ausgeführt werden):

- Das Grafikmodul ist der Viewer-Task.
- Die Subroutine **Graphics_OnStart()** wird beim Starten des Viewer-Tasks einmal auf dem Server ausgeführt.
- Die Subroutine **Graphics_WhileRunning()** wird während der Ausführung des Viewer-Tasks kontinuierlich auf dem Server ausgeführt. Die Subroutine **Graphics_OnEnd()** wird beim Schließen des Viewer-Tasks einmal auf dem Server ausgeführt.

Auf dem Thin Client oder Secure Viewer:

- Das Grafikmodul ist das ISSymbol- Steuerelement.
- Die Subroutine **Graphics_OnStart()** wird nach erfolgreicher Anmeldung einmal auf dem Thin Client ausgeführt.
- Die Subroutine **Graphics_WhileRunning()** wird während des Hostings des ISSymbol-Steuerelements durch den Webbrowser kontinuierlich auf dem Thin Client ausgeführt.
- Die Subroutine **Graphics_OnStart()** wird beim Schließen des Webbrowsers einmal auf dem Thin Client ausgeführt (oder wenn das Hosting des ISSymbol-Steuerelements durch den Webbrowser beendet ist).

Die Ausführung der Grafikskriptsubroutinen auf dem Server ist komplett unabhängig von der Ausführung auf den Thin-Client- und Secure-Viewer-Stationen.

Aufrufen von Grafikskriptprozeduren in anderen VB-Skript-Schnittstellen

Die drei vordefinierten Suchroutinen sind ausschließlich für das Grafikmodul bestimmt und werden nur bei den oben genannten Ereignissen ausgeführt. Im Grafikskript können Sie jedoch weitere Funktionen deklarieren, die von

anderen Schnittstellen (z.B. in Bildskripten oder Befehlobjekten) mithilfe der Syntax **Graphics.prozedur_name** aufgerufen werden können.

Bei der im obigen Beispiel deklarierten Funktion **MyNewProcedure** könnte mit der folgenden Zeile ein Schaltflächenobjekt auf dem WinStudio-Bild platziert und dann eine Befehlsanimation darauf angewendet werden:

$NewTag = Graphics.MyNewProcedure($oldTag)

Bildskript

Zum Bearbeiten des Bildskripts eines WinStudio-Bilds öffnen Sie dieses im Editor und wählen dann im Kontextmenü des Editors den Eintrag **Bild-Skript** aus.

Bildskripte können zum Ausführen von Logiken bei den folgenden Ereignissen verwendet werden, basierend auf vordefinierten Subroutinen:

- **Screen_OnOpen()**: Der innerhalb dieser Subroutine konfigurierte Code wird automatisch beim Öffnen des Bildes einmalig ausgeführt.
- **Screen_WhileOpen()**: Der innerhalb dieser Subroutine konfigurierte Code wird automatisch kontinuierlich ausgeführt, während das Bild geöffnet ist. Die Geschwindigkeit, mit der diese Subroutine aufgerufen wird, hängt von der Leistungsfähigkeit der Plattform, auf dem das Projekt ausgeführt wird, ab.
- **Screen_OnClose()**: Der innerhalb dieser Subroutine konfigurierte Code wird beim Schließen des Bildes einmalig ausgeführt.
- **Screen_IsClosedByReplace()**: Diese Prozedur legt fest, ob das Bild beim Öffnen eines anderen Bildes automatisch geschlossen wird (dies entspricht dem Standardverhalten). Gibt die Funktion den Wert "0" oder "FALSE" zurück, wird das automatische Schließen deaktiviert. Erhält die Funktion einen positiven Wert (z.B. "1") oder "TRUE" oder wird die Prozedur nicht deklariert, so wird das automatische Schließen aktiviert.

Die in dieser Schnittstelle deklarierten Variablen und Prozeduren stehen für die VBScript-Schnittstellen des Bildes, in dem das Bildskript konfiguriert ist, zur Verfügung.

HINWEIS **Die Namen der zuvor beschriebenen vorkonfigurierten Subroutinen dürfen "nicht" verändert werden.**

Andernfalls kann das System diese nicht automatisch aufrufen.

- Die Ausführung der Bildskript-Subroutinen auf dem Server ist komplett unabhängig von der Ausführung auf den Thin-Client-Stationen. Das bedeutet, dass diese Subroutinen asynchron ausgeführt werden.
- Die in dieser Bildskriptschnittstelle deklarierten Prozeduren bzw. Variablen haben lokale Gültigkeit.

  Sie können nur aus dem bestimmten Bild aufgerufen werden, in dem sie deklariert sind.

Beispiel:

```
'Variables available on this screen can be declared and initialized here
Dim Counter

'Procedures available on this screen can be implemented here
Function AreaCircle(radius)
  AreaCircle = Sqr(radius) * $Pi()
```

```
End Function

Sub CheckLoLimit (myValue, myLoLimit)
  If myValue < myLoLimit Then
    MsgBox("Value out of range")
  End If
End Sub

'This procedure determines whether the screen is automatically closed
Function Screen_IsClosedByReplace()
  Screen_IsClosedByReplace = $ReplaceModeTag
End Function

'This procedure is executed just once when this screen is open
Sub Screen_OnOpen()
  MsgBox("The screen was open!")
End Sub

'This procedure is executed continuously while this screen is open
Sub Screen_WhileOpen()
  If Counter < 100 Then
    Counter = Counter + 1
  Else
    Counter = 0
  End If
  $SimulationTag = Counter
End Sub

'This procedure is executed just once when this screen is closed
Sub Screen_OnClose()
  MsgBox("The screen will be closed!")
End Sub
```

Befehlsanimation

Zum Bearbeiten von Skripten in Befehlsanimationen gehen Sie wie folgt vor:

1. Wählen Sie das Objekt aus.
2. Klicken Sie in der Symbolleiste "WinStudio-Dynamikfunktionen" auf das Symbol **Kommando ein-/ausschalten.**.

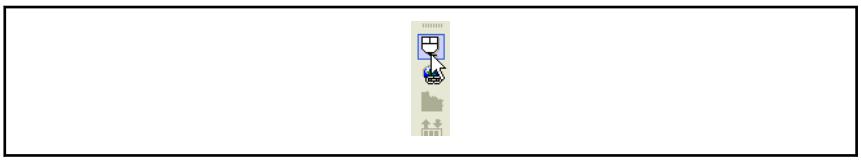


Abb.20-5: Symbol Befehlsanimationen

3. Führen Sie einen Doppelklick auf das Objekt aus. Der Dialog "Objekteigenschaften" für die Befehlsanimation wird angezeigt.

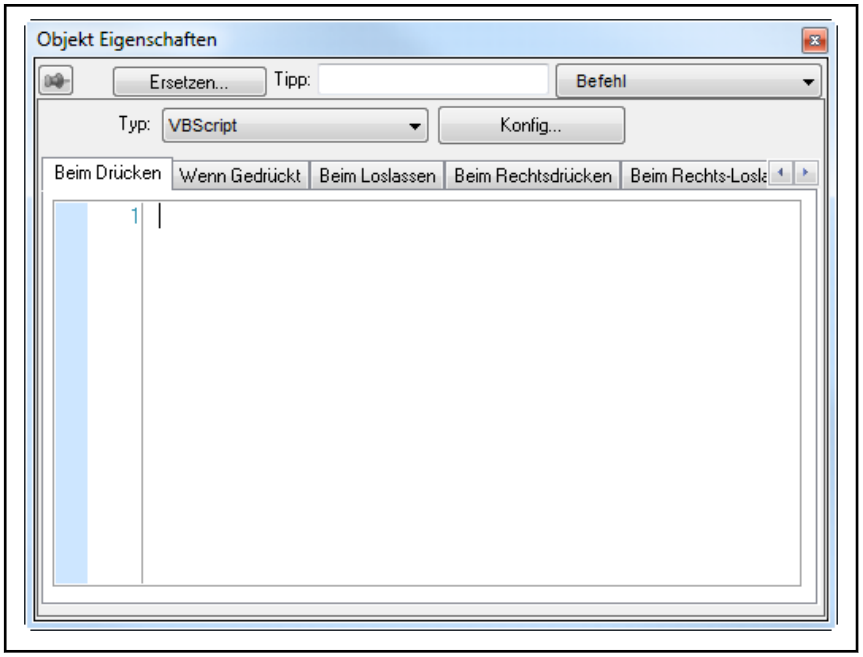


Abb.20-6: Dialog Objekteigenschaften für die Befehlsanimation

4. Wählen Sie **VBScript** als **Typ** aus.
5. Um das Skript in einem größeren Editorfenster zu bearbeiten oder um weitere Optionen einzustellen, klicken Sie auf die Schaltfläche **Konfig...**.

Verwenden Sie diese Schnittstelle zum Ausführen von Funktionen, wenn der Benutzer während der Laufzeit auf das Objekt klickt oder die für Objekt konfigurierte Tastenkombination drückt.

In dieser Schnittstelle deklarierte Variablen stehen nur für diese Schnittstelle zur Verfügung (lokaler Bereich). Sie stehen also nicht für andere Objekte im Projekt zur Verfügung. In dieser Schnittstelle können keine Prozeduren implementiert werden. Es können aber Prozeduren aufgerufen werden, die in den Globalen Prozeduren oder im Bildskript desselben Bildes, in dem sich das Bildobjekt befindet, implementiert sind.

☞ Weitere Informationen unter Befehlsanimation auf Seite 243.

Beispiel:

```
'The script below will be executed when the user clicks on the object
'where this animation is configured
$MyValue = InputBox("Please enter the new set-point", "Set-point")
```

ActiveX-Ereignisse

Zum Bearbeiten der Schnittstelle für ActiveX-Ereignisse wählen Sie die Option Skript auf der Registerkarte Ereignisse des im Bild eingefügten ActiveX-Objekts.

1. Klicken Sie auf das Symbol "ActiveX-Objekt" im Werkzeugfenster "WinStudio Screen Tools".

Anhang: VBScript-Sprache

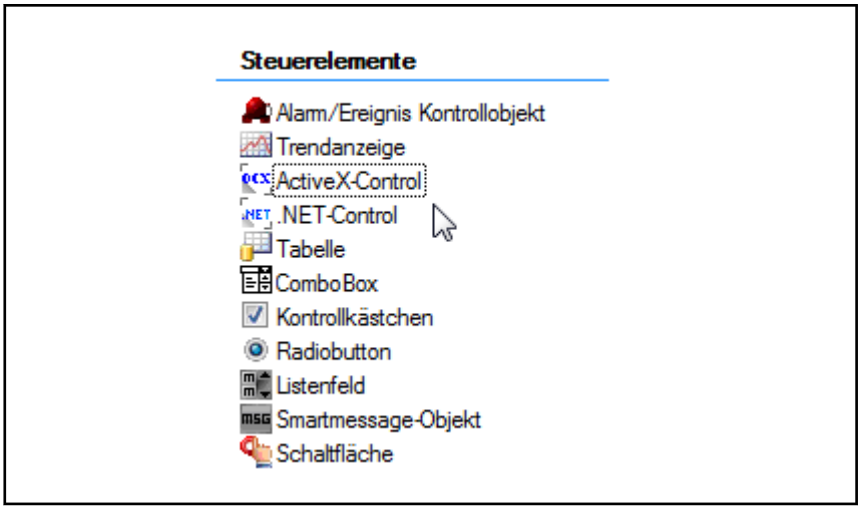


Abb.20-7: Symbol für ActiveX-Steuerelemente in der Menüleiste für aktive Objekte

Der Dialog "ActiveX-Objekt einfügen" wird geöffnet.

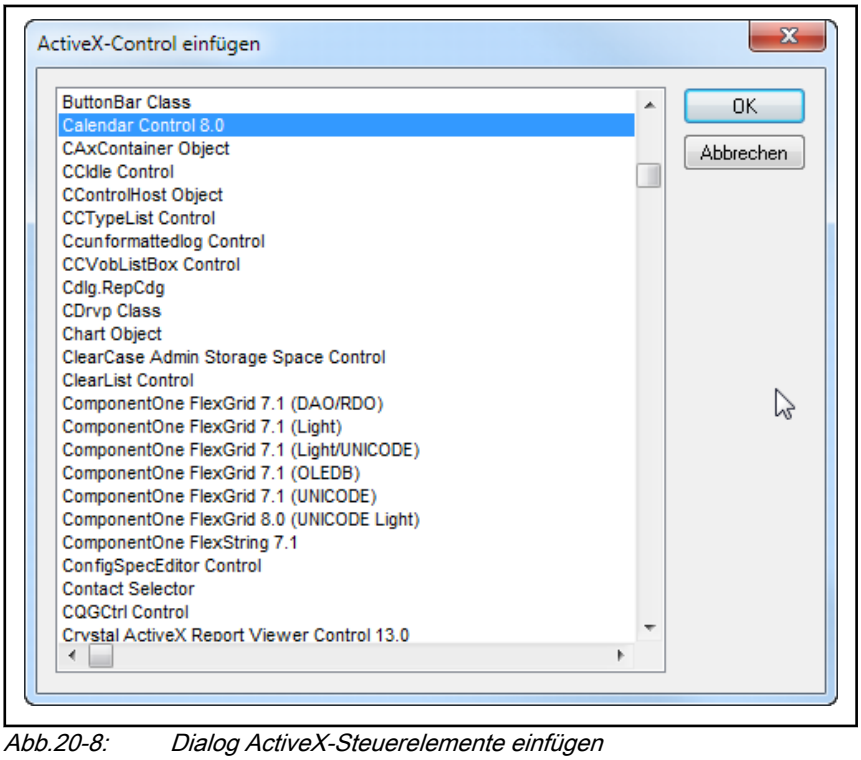


Abb.20-8: Dialog ActiveX-Steuerelemente einfügen

2. Wählen Sie das gewünschte ActiveX-Objekt aus und klicken Sie dann auf **OK**.
3. Das Objekt wird erzeugt und in der linken oberen Ecke in das Bild eingefügt. Mit einem Doppelklick auf das Objekt öffnet sich der Dialog "Objekteigenschaften".

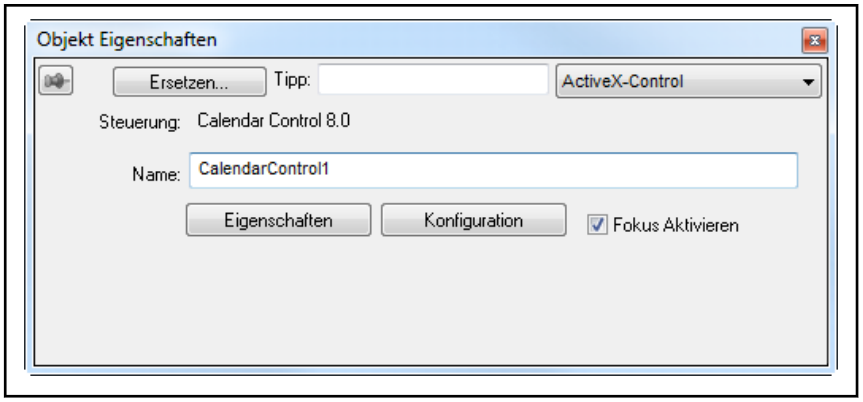


Abb.20-9: Dialog Objekteigenschaften

4. Klicken Sie auf die Schaltfläche **Konfiguration**. Der Dialog "Konfiguration" wird geöffnet. Klicken Sie auf die Registerkarte Ereignisse.

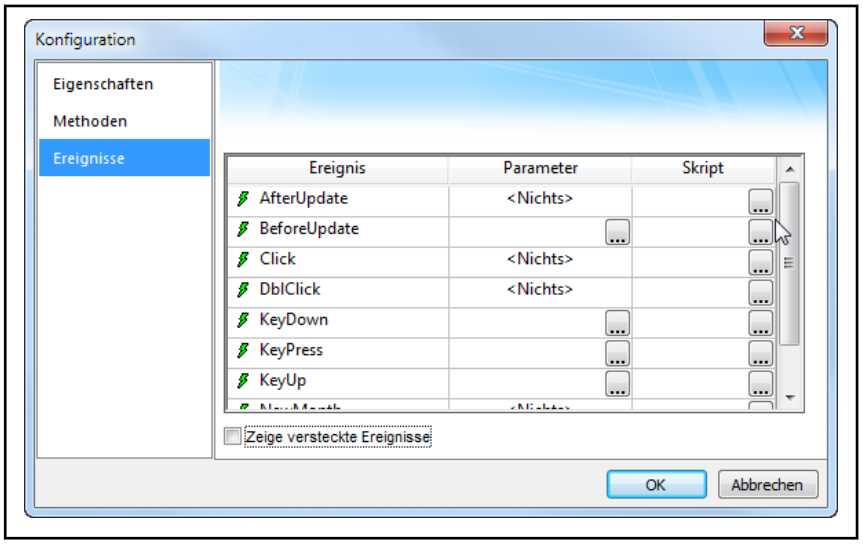


Abb.20-10: Registerkarte Ereignisse

5. Klicken Sie in der Skriptspalte auf die Schaltfläche **...**.

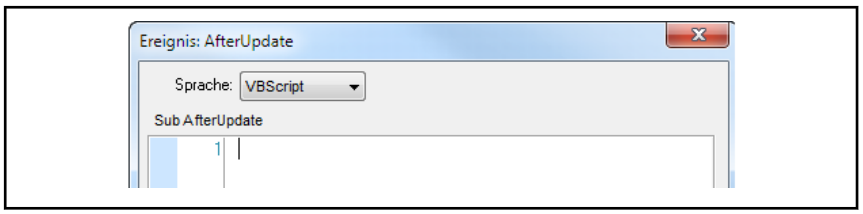


Abb.20-11: Skriptspalte

Dieser Schnittstelle wird zum Ausführen von Logiken verwendet, wenn ein ActiveX-Objekt ein Ereignis auslöst.

In dieser Schnittstelle deklarierte Variablen stehen nur für diese Schnittstelle zur Verfügung (lokaler Bereich). Sie stehen also nicht in anderen Objekten des Projekts verwendet werden.

In dieser Schnittstelle können keine Prozeduren implementiert werden. Es können aber Prozeduren aufgerufen werden, die in den Globalen Prozeduren oder im Bildskript desselben WinStudio-Bildes, in dem das ActiveX-Objekt enthalten ist, implementiert sind.

Weitere Informationen unter ActiveX-Steuerobjekt auf Seite 229.

Beispiel:

```
'The script below will be executed when the Calendar Control ActiveX
'triggers its "AfterUpdate" event
$MyYear = CalendarControl1.Year
$MyMonth = CalendarControl1.Month
$MyDay = CalendarControl1.Day
```

20.2.4 Hintergrundtask

Weitere Details siehe:

- Skriptarbeitsblatt Start auf Seite 369.
- Skriptarbeitsblatt auf Seite 368.

20.3 Sprachverweis

20.3.1 Sprachverweis – Überblick

- Konstanten, Seite 809
- Funktionen, Seite 817
- Fehler, Seite 818
- Schlüsselwörter, Seite 818
- Methoden, Seite 817
- Objekte und Kollektionen, Seite 813
- Operatoren, Seite 808
- Eigenschaften, Seite 813
- Anweisungen, Seite 814
- System-Tags, Seite 821

20.3.2 Operatoren

Arithmetische Operatoren

Symbol	Name	Beschreibung
^	Potenzierung	Potenziert eine Zahl mit einem Exponenten.
-	Einstellige Negation	Bildet die Differenz zwischen zwei Zahlen oder gibt den negativen Werts eines numerischen Ausdrucks an.
*	Multiplikation	Multipliziert zwei Zahlen.
/	Division	Dividiert zwei Zahlen und gibt ein Ergebnis im Gleitkommaformat aus.
\	Ganzzahldivision	Dividiert zwei Zahlen und gibt einen Ergebnis als Ganzzahl zurück.
Mod	Absolutwertarithmetik	Dividiert zwei Zahlen und gibt nur den Rest zurück.
+	Addition	Bildet die Summe zweier Zahlen.
-	Subtraktion	Bildet die Differenz zwischen zwei Zahlen oder gibt den negativen Werts eines numerischen Ausdrucks an.
&	Zeichenkettenverknüpfung	Erzwingt die Zeichenkettenverknüpfung zweier Ausdrücke.

Vergleichende Operatoren

Symbol	Name	Beschreibung
=	Gleichheit	Vergleich ist wahr, wenn der erste Ausdruck gleich dem zweiten Ausdruck ist.
<>	Ungleichheit	Vergleich ist wahr, wenn der erste Ausdruck nicht gleich dem zweiten Ausdruck ist.
<	Kleiner als	Vergleich ist wahr, wenn der erste Ausdruck kleiner als der zweite Ausdruck ist.
>	Größer als	Vergleich ist wahr, wenn der erste Ausdruck größer als der zweite Ausdruck ist.
<=	Kleiner oder gleich	Vergleich ist wahr, wenn der erste Ausdruck kleiner oder gleich dem zweiten Ausdruck ist.
>=	Größer oder gleich	Vergleich ist wahr, wenn der erste Ausdruck größer oder gleich dem zweiten Ausdruck ist.
Is	Objektgleichheit	Vergleicht zwei Objektreferenzvariablen. Vergleich ist wahr, wenn sich beide Objektnamen auf das gleiche Objekt beziehen.

Logische Operatoren

Symbol	Name	Beschreibung
Not	Logische Negation	Führt eine logische Negation an einem Ausdruck aus.
And	Logische Konjunktion	Führt eine logische Verknüpfung an zwei Ausdrücken aus.
Or	Logische Disjunktion	Führt eine logische Disjunktion an zwei Ausdrücken aus.
Xor	Logischer Ausschluss	Führt einen logischen Ausschluss an zwei Ausdrücken aus.
Eqv	Logische Äquivalenz	Führt eine logische Äquivalenz an zwei Ausdrücken aus.
Imp	Logische Implikation	Führt eine logische Implikation an zwei Ausdrücken aus.

Zuweisende Operatoren

Symbol	Name	Beschreibung
=	Zuweisung	Weist einer Variablen oder Eigenschaft einen Wert zu.

20.3.3 Konstanten

Farbkonstanten

Konstante	Wert	Beschreibung
vbBlack	&h00	Schwarz
vbRed	&hFF	Rot
vbGreen	&hFF00	Grün
vbYellow	&hFFFF	Gelb
vbBlue	&hFF0000	Blau
vbMagenta	&hFF00FF	Magenta

Anhang: VBScript-Sprache

Konstante	Wert	Beschreibung
vbCyan	&hFFFF00	Cyan
vbWhite	&hFFFFFF	Weiß

Vergleichende Konstanten

Konstante	Wert	Beschreibung
vbBinaryCompare	0	Führt einen binären Vergleich aus
vbTextCompare	1	Führt einen Textvergleich aus

Datums- und Uhrzeitkonstanten

Konstante	Wert	Beschreibung
vbSunday	1	Sonntag
vbMonday	2	Montag
vbTuesday	3	Dienstag
vbWednesday	4	Mittwoch
vbThursday	5	Donnerstag
vbFriday	6	Freitag
vbSaturday	7	Samstag
vbUseSystemDayOfWeek	0	Verwendet den in den Systemeinstellungen als ersten Wochentag festgelegten Tag.
vbFirstJan1	1	Verwendet die Woche, in die der 1. Januar fällt (Standard).
vbFirstFourDays	2	Verwendet die erste Woche im neuen Jahr, die mindestens vier Tage hat.
vbFirstFullWeek	3	Verwendet die erste volle Woche des Jahres.

Datumsformatkonstanten

Konstante	Wert	Beschreibung
vbGeneralDate	0	Zeigt Datum und/oder Uhrzeit an. Für reelle Zahlen werden Datum und Uhrzeit angezeigt. Ohne Bruchanteil wird nur das Datum angezeigt. Ohne Ganzzahlanteil wird nur die Uhrzeit angezeigt. Anzeige von Datum und Uhrzeit sind von Ihren Systemeinstellungen abhängig.
vbLongDate	1	Zeigt ein Datum in dem in den Regionseinstellungen Ihres Rechners festgelegten langen Datumsformat an.
vbShortDate	2	Zeigt ein Datum in dem in den Regionseinstellungen Ihres Rechners festgelegten kurzen Datumsformat an.
vbLongTime	3	Zeigt eine Uhrzeit in dem in den Regionseinstellungen Ihres Rechners festgelegten langen Uhrzeitformat an.
vbShortTime	4	Zeigt eine Uhrzeit in dem in den Regionseinstellungen Ihres Rechners festgelegten kurzen Uhrzeitformat an.

Verschiedene Konstanten

Konstante	Wert	Beschreibung
vbObjectError	-2147221504	Benutzerdefinierte Fehlernummern sollten größer als dieser Wert sein.

Fensterkonstanten – Schaltflächen und Symbole

Konstante	Wert	Beschreibung
vbOKOnly	0	Nur Schaltfläche OK anzeigen.
vbOKCancel	1	Schaltflächen OK und Abbrechen anzeigen.
vbAbortRetryIgnore	2	Schaltflächen Abbrechen, Wiederholen und Ignorieren anzeigen.
vbYesNoCancel	3	Schaltflächen Ja, Nein und Abbrechen anzeigen.
vbYesNo	4	Schaltflächen Ja und Nein anzeigen.
vbRetryCancel	5	Schaltflächen Wiederholen und Abbrechen anzeigen.
vbCritical	16	Symbol für kritische Meldung anzeigen.
vbQuestion	32	Symbol für Warnabfrage anzeigen.
vbExclamation	48	Symbol für Warnmeldung anzeigen.
vbInformation	64	Symbol für Informationsmeldung anzeigen.
vbDefaultButton1	0	Erste Schaltfläche ist Standard.
vbDefaultButton2	256	Zweite Schaltfläche ist Standard.
vbDefaultButton3	512	Dritte Schaltfläche ist Standard.
vbDefaultButton4	768	Vierte Schaltfläche ist Standard.
vbApplicationModal	0	Anwendung modal. Der Benutzer muss das Meldungsfenster schließen, bevor mit der aktuellen Anwendung weitergearbeitet werden kann.
vbSystemModal	4096	System modal. Auf Win16-Systemen werden alle Programme angehalten, bis der Benutzer das Meldungsfenster schließt. Auf Win32-Systemen stellt dieses Programm ein programmmodales Meldungsfenster zur Verfügung, das vor allen laufenden Programmen angezeigt wird.

Fensterkonstanten – ausgewählte Schaltflächen

Konstante	Wert	Beschreibung
vbOK	1	Schaltfläche OK wurde angeklickt.
vbCancel	2	Schaltfläche Abbrechen wurde angeklickt.
vbAbort	3	Schaltfläche Abbrechen wurde angeklickt.
vbRetry	4	Schaltfläche Wiederholen wurde angeklickt.
vbIgnore	5	Schaltfläche Ignorieren wurde angeklickt.
vbYes	6	Schaltfläche Ja wurde angeklickt.
vbNo	7	Schaltfläche Nein wurde angeklickt.

Anhang: VBScript-Sprache

String-Konstanten

Konstante	Wert	Beschreibung
vbCr	Chr(13)	Eingabetaste
VbCrLf	Chr(13) & Chr(10)	Kombination Eingabetaste...Zeilenvorschub
vbFormFeed	Chr(12)	Zeilenvorschub; in Microsoft Windows nicht sinnvoll
vbLf	Chr(10)	Zeilenvorschub
vbNewLine	Chr(13) & Chr(10) oder Chr(10)	Plattform-spezifisches Zeichen für neuen Zeile; entsprechend der Plattform
vbNullChar	Chr(0)	Zeichen mit dem Wert 0
vbNullString	Zeichenkette mit dem Wert 0	Nicht gleichwertig mit Zeichenkette mit Länge null (""); verwendet zum Aufrufen externer Prozeduren
vbTab	Chr(9)	Horizontaler Tab
vbVerticalTab	Chr(11)	Vertikaler Tab; in Microsoft Windows nicht sinnvoll

Dreiwertige Konstanten

Konstante	Wert	Beschreibung
vbUseDefault	−2	Verwendet den Standardwert aus den Regionseinstellungen des Rechners.
vbTrue	−1	TRUE
vbFalse	0	FALSE

VarType-Konstanten

Konstante	Wert	Beschreibung
vbEmpty	0	Nicht initialisiert (Standard)
vbNull	1	Enthält keine gültigen Daten
vbInteger	2	Subtyp Integer
vbLong	3	Subtyp Lang
vbSingle	4	Subtyp Single
vbDouble	5	Subtyp Double
vbCurrency	6	Subtyp Währung
vbDate	7	Subtyp Datum
vbString	8	Subtyp String
vbObject	9	Objekt
vbError	10	Subtyp Fehler
vbBoolean	11	Subtyp Boolean
vbVariant	12	Variante (nur für Variantenbereich verwendet)
vbDataObject	13	Datenzugriffsobjekt
vbDecimal	14	Subtyp Dezimal

Konstante	Wert	Beschreibung
vbByte	17	Subtyp Byte
vbArray	8192	Array

20.3.4 Objekte und Kollektionen

Class Object

Das mit der Klassenanweisung erstellte Objekt. Bietet Zugriff auf die Ereignisse der Klasse.

Debug Object

Ein intrinsisches globales Objekt, das Ausgaben an einen Skript-Debugger senden kann, wie z.B. den Microsoft Script Debugger.

Err Object

Enthält Informationen über Laufzeitfehler. Akzeptiert die Methoden Raise und Clear zum Erzeugen und Löschen von Laufzeitfehlern.

Match Object

Bietet Zugriff auf die schreibgeschützten Eigenschaften eines Treffers eines regulären Ausdrucks.

Matches Collection

Kollektion regulärer Ausdrucks-Matchobjekte.

Objekt regulärer Ausdruck (RegExp)

Bietet Unterstützung für einfache reguläre Ausdrücke.

SubMatches Collection

Kollektion regulärer Ausdrucks-Submatch-Strings.

20.3.5 Eigenschaften

Beschreibung

Setzt oder gibt einen beschreibenden String in Verbindung mit einem Fehler zurück.

FirstIndex

Gibt die Position in einem Such-String zurück, in dem ein Match auftritt.

Global

Setzt oder gibt einen Booleschen Wert zurück, der anzeigt, ob ein Muster zu allen Vorkommen in einem gesamten Such-String passen soll oder nur zum ersten.

HelpContext

Setzt oder gibt eine Kontext-ID für ein Thema in einer Hilfedatei zurück.

HelpFile

Setzt oder gibt einen voll qualifizierten Pfad zu einer Hilfedatei zurück.

IgnoreCase

Setzt oder gibt einen Booleschen Wert zurück, der angibt, ob eine Mustersuche Groß- und Kleinschreibung beachtet oder nicht.

Length

Setzt oder gibt einen Booleschen Wert zurück, der angibt, ob eine Mustersuche Groß- und Kleinschreibung beachtet oder nicht.

Number

Setzt oder gibt einen numerischen Wert zur Angabe eines Fehlers zurück. Number ist die Standardeigenschaft des Objekts Err.

Pattern

Setzt oder gibt das gesuchte reguläre Ausdrucksmuster zurück.

Source

Setzt oder gibt den Namen des Objekts oder der Anwendung zurück, die ursprünglich den Fehler erzeugt hat.

Wert

Gibt den Wert oder Text eines in einem Such-String gefundenen Match zurück.

20.3.6 Anweisungen

Call

Übergibt die Kontrolle an eine Sub- oder Funktionsprozedur.

Class

Deklariert den Namen einer Klasse sowie eine Definition der Variablen, Eigenschaften und Methoden, aus denen die Klasse besteht.

Const

Deklariert Konstanten zur Verwendung anstelle von Literalwerten.

Dim

Deklariert Variablen und weist Speicherplatz zu.

Do…Loop

Wiederholt einen Anweisungsblock, während eine Bedingung Wahr ist oder bis eine Bedingung Wahr wird.

Erase

Reinitialisiert die Elemente eines Bereichs mit fester Größe und entfernt die Zuweisung von Speicherplatz zu dynamischen Bereichen.

Execute

Führt eine oder mehrere festgelegte Anweisungen aus.

ExecuteGlobal

Führt eine oder mehrere festgelegte Anweisungen im globalen Namensraum eines Skripts aus.

Exit

Verlässt einen Block von Do...Loop, For...Next, Function oder Sub Code.

For Each…Next

Wiederholt eine Gruppe von Anweisungen für jedes Element in einem Bereich oder einer Kollektion.

For … Next

Führt eine festgelegte Anzahl von Wiederholungen einer Gruppe von Anweisungen aus.

Funktion

Deklariert Namen, Argumente und Code für den Hauptteil einer Funktionsprozedur.

If…Then…Else

Bedingte Ausführung einer Gruppe von Anweisungen, abhängig vom Wert eines Ausdrucks.

Option Explicit

Erzwingt die ausdrückliche Deklarierung aller Variablen in einem Skript.

Private

Deklariert private Variablen und weist Speicherplatz zu. Deklariert eine private Variable in einem Klassenblock.

Property Get

Deklariert in einem Klassenblock Name, Argumente und Code für den Hauptteil einer Eigenschaftsprozedur, die den Wert einer Eigenschaft abruft (zurückgibt).

Anhang: VBScript-Sprache

Property Let

Deklariert in einem Klassenblock Name, Argumente und Code für den Hauptteil einer Eigenschaftsprozedur, die den Wert einer Eigenschaft zuweist (setzt).

Property Set

Deklariert in einem Klassenblock Name, Argumente und Code für den Hauptteil einer Eigenschaftsprozedur, die einen Bezug auf ein Objekt setzt.

Public

Deklariert öffentliche Variablen und weist Speicherplatz zu. Deklariert eine private Variable in einem Klassenblock.

Randomize

Initialisiert den Zufallsgenerator für Zahlen.

ReDim

Deklariert Variablen in einem dynamischen Bereich und weist auf Prozedurebene Speicherplatz zu bzw. weist diesen erneut zu.

Rem

Fügt erklärende Bemerkungen in ein Programm ein.

Select

Führt eine von mehreren Gruppen von Anweisungen aus, abhängig vom Wert eines Ausdrucks.

Set

Weist einer Variablen oder Eigenschaft eine Objektreferenz zu oder verbindet einen Prozedurbezug mit einem Ereignis.

Stop

Pausiert die Ausführung.

Sub

Deklariert Namen, Argumente und Code für den Hauptteil einer Subprozedur.

While

Führt eine Reihe von Anweisungen aus, solange eine gegebene Bedingung Wahr ist.

With

Führt eine Reihe von Anweisungen auf einem einzelnen Objekt aus.

20.3.7 Methoden

Clear

Löscht alle Eigenschaftseinstellungen auf dem Objekt Err.

Execute

Führt eine reguläre Ausdruckssuche mit einem angegebenen String aus.

Raise

Erzeugt einen Laufzeitfehler.

Replace

Ersetzt bei einer regulären Ausdruckssuche gefundenen Text.

Test

Führt eine reguläre Ausdruckssuche mit einem angegebenen String aus und gibt einen Booleschen Wert zurück, der anzeigt, ob ein Match für das Muster gefunden wurde.

Write

Sendet Strings an den Skript-Debugger.

WriteLine

Sendet Strings an den Skript-Debugger, gefolgt von einem Zeichen „Neue Zeile“.

20.3.8 Funktionen

Funktionsnamen			
Abs	Array	Asc	Atn
CBool	CByte	CCur	CDate
CDbl	Chr	CInt	CLng
Conversions	Cos	CreateObject	CSng
CStr	Date	DateAdd	DateDiff
DatePart	DateSerial	DateValue	Day
Derived Math	Escape	Eval	Exp
Filter	FormatCurrency	FormatDateTime	FormatNumber
FormatPercent	GetLocale	GetObject	GetRef
Hex	Hour	InputBox	InStr
InStrRev	Int, Fix	IsArray	IsDate
IsEmpty	IsNull	IsNumeric	IsObject
Join	LBound	LCase	Left
Len	LoadPicture	Log	LTrim; RTrim; and Trim

Anhang: VBScript-Sprache

Funktionsnamen			
Maths	Mid	Minute	Month
MonthName	MsgBox	Now	Oct
Replace	RGB	Right	Rnd
Round	ScriptEngine	ScriptEngineBuildVersion	ScriptEngineMajorVersion
ScriptEngineMinorVersion	Second	SetLocale	Sgn
Sin	Space	Split	Sqr
StrComp	Zeichenkette	StrReverse	Tan
Time	Timer	TimeSerial	TimeValue
TypeName	UBound	UCase	Unescape
VarType	Weekday	WeekdayName	Year

20.3.9 Schlüsselwörter

Empty

Das Schlüsselwort Empty wird zum Anzeigen eines nicht initialisierten Variablenwerts verwendet. Dies ist nicht gleichwertig mit null.

False

Das Schlüsselwort False hat einen Wert gleich 0.

Nothing

Das Schlüsselwort Nothing wird in VBScript verwendet, um eine Objektvariable von einem tatsächlichen Objekt zu trennen.

Null

Das Schlüsselwort Null dient zum Anzeigen, dass eine Variable keine gültigen Daten enthält. Dies ist nicht gleichwertig mit Empty (leer).

True

Das Schlüsselwort True hat einen Wert gleich -1.

20.3.10 Fehler

VBScript Laufzeitfehler

Fehlernummer	Beschreibung
5	Ungültiger Prozeduraufruf oder Argument
6	Überlauf
7	Zu wenig Speicherplatz
9	Indexliste überschritten
10	Dieser Bereich ist fixiert oder vorübergehend gesperrt
11	Division durch Null

Fehlernummer	Beschreibung
13	Typen unverträglich
14	Stringbereich überschritten
17	Angeforderte Operation kann nicht ausgeführt werden
28	Stack-Bereich überschritten
35	Sub oder Funktion nicht definiert
48	Fehler beim Laden von DLL
51	Interner Fehler
91	Objektvariable nicht gesetzt
92	Schleife „Für“ nicht initialisiert
94	Ungültige Verwendung von Null
424	Objekt erforderlich
429	ActiveX-Komponente kann Objekt nicht erstellen
430	Klasse unterstützt keine Automation
432	Dateiname oder Klassenname bei Automationsoperation nicht gefunden
438	Eigenschaft oder Methode vom Objekt nicht unterstützt
445	Aktion vom Objekt nicht unterstützt
447	Lokale Einstellung vom Objekt nicht unterstützt
448	Benanntes Argument nicht gefunden
449	Argument ist nicht optional
450	Falsche Anzahl an Argumenten oder ungültige Eigenschaftszuweisung
451	Objekt ist keine Kollektion
458	Variablen verwenden einen von VBScript nicht unterstützten Automationstyp
462	Das Remote-Server-Gerät existiert nicht oder ist nicht verfügbar
481	Ungültiges Bild
500	Variable ist nicht definiert
502	Objekt nicht sicher für Skripting
503	Objekt nicht sicher für Initialisierung
504	Objekt nicht sicher für Erstellung
505	Ungültiger oder unqualifizierter Bezug
506	Klasse nicht definiert
507	Ein Ausnahmefehler ist aufgetreten
5008	Unzulässige Zuweisung
5017	Syntaxfehler in regulärem Ausdruck
5018	Unerwartetes Klammerzeichen
5019	']' in regulärem Ausdruck erwartet

Anhang: VBScript-Sprache

Fehlernummer	Beschreibung
5020	')' in regulärem Ausdruck erwartet
5021	Ungültiger Bereich im Zeichensatz

VBScript Syntaxfehler

Fehlernummer	Beschreibung
1001	Zu wenig Speicherplatz
1002	Syntaxfehler
1005	'(' erwartet
1006	')' erwartet
1010	Bezeichner erwartet
1011	'=' erwartet
1012	'If' erwartet
1013	'To' erwartet
1014	'End' erwartet
1015	'Function' erwartet
1016	'Sub' erwartet
1017	'Then' erwartet
1018	'Wend' erwartet
1019	'Loop' erwartet
1020	'Next' erwartet
1021	'Case' erwartet
1022	'Select' erwartet
1023	Ausdruck erwartet
1024	Anweisung erwartet
1025	Anweisungsende erwartet
1026	Ganzzahlkonstante erwartet
1027	'While' oder 'Until' erwartet
1028	'While', 'Until' oder Anweisungsende erwartet
1029	'With' erwartet
1030	Bezeichner zu lang
1037	Ungültige Verwendung des Schlüsselworts 'Me'
1038	'loop' ohne 'do'
1039	Ungültige 'exit'-Anweisung
1040	Ungültige Schleifensteuervariable 'for'
1041	Name neu definiert
1042	Muss die erste Anweisung in der Zeile sein

Fehlernummer	Beschreibung
1044	Bei Aufruf von Sub können keine Klammern verwendet werden
1045	Literalkonstante erwartet
1046	'In' erwartet
1047	'Class' erwartet
1048	Muss innerhalb einer Klasse definiert werden
1049	Let, Set oder Get in Eigenschaftsdeklaration erwartet
1050	'Property' erwartet
1051	Anzahl der Argumente muss für Eigenschaftsspezifikation durchgängig sein
1052	Mehrfache Standardeigenschaften/-methoden in einer Klasse nicht zulässig
1053	Keine Argumente für Klasse initialisieren oder beenden
1054	Eigenschaft Set oder Let muss mindestens ein Argument haben
1055	'Next' unerwartet
1057	'Default'-Spezifikation muss auch 'Public' angeben
1058	Spezifikation 'Default' kann nur auf Eigenschaft Get sein

20.3.11 Liste der System-Tags

Liste der System-Tags

Tag-Name	Datentyp	Beschreibung	Bereich
AckAll	Boolean	Umschalten auf "1" zum Bestätigen aller Alarme. "1)"	Server
AckAlr	Boolean	Umschalten auf "1" zum Bestätigen des unbestätigten Alarms mit der höchsten Priorität; siehe "Alarm" unten. "1)"	Server
Alarm	Zeichenkette	Der Name des unbestätigten Alarms mit der höchsten Priorität. "1)"	Server
AnalogValue_	Real	Ein simulierter analoger Wert, der stetig von "0" auf "96" ansteigt mit einer Geschwindigkeit 1,6/sec. Wenn "96" erreicht ist, wird wieder bei "0" angefangen.	Lokal
Beep	Boolean	Der aktuelle Zustand des akustischen Alarms: "0" = AUS, "1" = EIN. "1)"	Server
BeepOff	Boolean	Umschalten auf "1" zum Deaktivieren des akustischen Alarms. "1)"	Server
BlinkFast	Boolean	Kontinuierliches Zwei-Sekunden-Blinken – eine Sekunde an, eine Sekunde aus.	Lokal
BlinkSlow	Boolean	Kontinuierliches Sechs-Sekunden-Blinken – drei Sekunden an, drei Sekunden aus.	Lokal
CrispDisplay	Integer	Altsystem-Tags für die benutzerdefinierte CrispView-Schnittstelle. Diese sind in "WinStudio" nur zur Abwärtskompatibilität enthalten und sollten in neuen Projekten nicht verwendet werden. Weitere Informationen erhalten Sie vom technischen Support.	Lokal
CrispInput	Zeichenkette		Lokal
CrispOutput	Zeichenkette		Lokal
Date	Zeichenkette	Das aktuelle Datum auf der lokalen Station, dargestellt als Zeichenkette im Format MM/TT/JJJJ. Zum Beispiel "05/15/2008". (**Hinweis**: Dies kann sich vom aktuellen Datum auf dem Projektserver unterscheiden; siehe "ServerDate_" unten.)	Lokal
Day	Integer	Der aktuelle Tag des Monats, zwischen "1" und "31".	Lokal

Anhang: VBScript-Sprache

Tag-Name	Datentyp	Beschreibung	Bereich
DayOfYear	Integer	Der aktuelle Tag des Jahres, zwischen "0" und "365".	Lokal
DigitalValue_	Boolean	Ein simulierter digitaler Wert, der jede Sekunde zwischen "0" und "1" wechselt.	Lokal
Goto	Zeichenkette	Goto, mathematische Funktion	Lokal
GroupCNFHiLevel	Integer	Ende des aktuellen Sicherheitsebenenbereichs zu CNF	Lokal
GroupCNFLoLevel	Integer	Anfang des aktuellen Sicherheitsebenenbereichs zu CNF	Lokal
GroupHiLevel	Integer	Die maximale Sicherheitsebene der aktuellen Benutzergruppe. "2)"	Lokal
GroupLoLevel	Integer	Die minimale Sicherheitsebene der aktuellen Benutzergruppe. "2)"	Lokal
GroupName	Zeichenkette	Der Name der Gruppe, der der aktuelle Benutzer zugeordnet ist. "2)"	Lokal
Hint	Zeichenkette	Der Hinweistext für das aktuelle Objekt. Weitere Informationen unter Objekteigenschaften, Seite 178	Lokal
Hour	Integer	Die aktuelle Stunde des Tages, zwischen "0" und "23".	Lokal
InputMaxRange	Real	Der Maximalwert (Max) für den aktuellen Tag bzw. das aktuelle Objekt; siehe "InputOutOfRange" unten.	Lokal
InputMinRange	Real	Der Minimalwert (Min) für den aktuellen Tag bzw. das aktuelle Objekt; siehe "InputOutOfRange" unten.	Lokal
InputOutOfRange	Boolean	Dieser Tag schaltet auf "1" um, wenn die Benutzereingabe sich außerhalb des Max-/Min-Bereichs des aktuellen Tags oder Objekts befindet.	Lokal
Label	Zeichenkette	Bezeichnung, mathematische Funktion	Lokal
LastCodeChar_	Integer	Letztes Codezeichen im Viewer (???)	Lokal
LptOff	Boolean	Umschalten auf "1" zum Deaktivieren des Druckens von Alarmen auf dem Standarddrucker. "1)"	Lokal
Minute	Integer	Der aktuelle Minute der Stunde, zwischen "0" und "59".	Lokal
Month	Integer	Der aktuelle Monat des Jahres, zwischen "1" und "12".	Lokal
Next	Integer	Nächste, mathematische Funktion	Lokal
Reserved___1	Boolean	reserviert für zukünftige Verwendung	Lokal
Reserved___2	Boolean	reserviert für zukünftige Verwendung	Lokal
Reserved___8	Boolean	reserviert für zukünftige Verwendung	Lokal
Reserved___9	Boolean	reserviert für zukünftige Verwendung	Lokal
Reserved___10	Boolean	reserviert für zukünftige Verwendung	Lokal
Second	Integer	Der aktuelle Sekunde der Minute, zwischen "0" und "59".	Lokal
ServerDate_	Zeichenkette	Das aktuelle Datum auf dem Projektserver, dargestellt als Zeichenkette im Format MM/TT/JJJJ. Zum Beispiel "05/15/2008".	Server
ServerTime_	Zeichenkette	Die aktuelle Zeit auf dem Projektserver, dargestellt als Zeichenkette im Format HH:MM:SS. Zum Beispiel "23:45:03".	Server
Time	Zeichenkette	Die aktuelle Zeit auf der lokalen Station, dargestellt als Zeichenkette im Format HH:MM:SS. Zum Beispiel "23:45:03". (**Hinweis**: Diese kann sich von der aktuellen Zeit auf dem Projektserver unterscheiden; siehe "ServerTime_" oben.)	Lokal
Tomorrow	Integer	Der nächste Tag des Monats, zwischen "1" und "31".	Lokal

Tag-Name	Datentyp	Beschreibung	Bereich
UserName	Zeichenkette	Der Name des aktuell angemeldeten Benutzers. "2)"	Lokal
Weekday	Integer	Der aktuelle Wochentag auf der lokalen Station, dargestellt als Ganzzahl: "0" = Sonntag, "1" = Montag, "2" = Dienstag, "3" = Mittwoch, "4" = Donnerstag, "5" = Freitag, "6" = Samstag.	Lokal
Year	Integer	Das aktuelle Jahr, vierstellig (z.B. "2008").	Lokal
Yesterday	Integer	Der vorhergehende Tag des Monats, zwischen "1" und "31".	Lokal

"1)" Weitere Information zum Konfigurieren von Alarmen finden Sie unter Ordner Alarme, Seite 273.

"2)" Weitere Informationen zum Konfigurieren von Benutzern, Gruppen und Sicherheitsebenen unter Ordner Sicherheit, Seite 407.

20.4 Tipps & Tricks

20.4.1 Tipps & Tricks – Überblick

- VBScript Editor IntelliSense, Seite 823
- VBScript im Vergleich mit VBA, Seite 825
- Bildereignisse, Seite 827
- Funktionen MsgBox() und InputBox(), Seite 827
- Unterstützung für ActiveX-Objekte, Seite 839
- Logischer Operator NOT, Seite 833
- Boolesche Variablen, Seite 831
- Windows CE Unterstützung, Seite 839
- Geltungsbereich und Lebensdauer von Variablen, Seite 830
- Deklarieren von Variablen, Seite 830
- Erstellen von Konstanten, Seite 829
- Priorität von VBScript-Operatoren, Seite 832
- Verwenden von bedingten Anweisungen, Seite 834
- Schleifen mit Code, Seite 836
- VBScript-Prozeduren, Seite 828

20.4.2 VBScript Editor IntelliSense

IntelliSense bietet eine Reihe von Möglichkeiten, die Sprachverweise leicht zugänglich machen. Beim Codieren müssen der Code Editor oder das Immediate-Mode-Befehlsfenster zum Durchführen von Suchen in Sprachelementen nicht verlassen werden. Sie können Ihren Kontext beibehalten, die gewünschte Information finden, Sprachelemente direkt in Ihren Code einfügen und sogar alle Eingaben von IntelliSense ausführen lassen.

IntelliSense umfasst die folgenden Optionen...

Listen-Member Sie können ein Liste mit gültigen Members aus Klassen-Tags, Feldern aus jedem beliebigen Tag, Eigenschaften/Methoden aus einem ActiveX-Objekt oder Funktionen aus der eingebauten Skriptsprache anzeigen lassen. Durch Auswählen aus der Liste wird das Member in Ihren Code eingefügt.

Anhang: VBScript-Sprache

Durch Eingabe des $-Zeichens in einer VBScript-Schnittstelle öffnet sich automatisch ein Listenfeld mit einer Liste aller für das aktuelle Projekt verfügbaren Tags sowie aller Funktionen aus der eingebauten Skriptsprache.

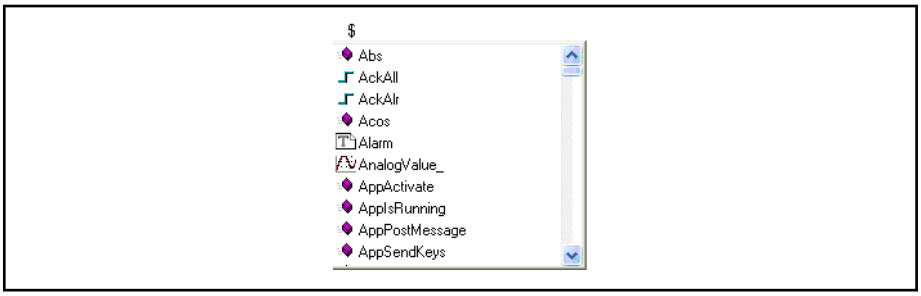


Abb.20-12: Listenfeld Tag/Funktion mit $-Zeichen öffnen

Wenn Sie in einer VBScript-Schnittstelle den Namen eines Klassen-Tags gefolgt von einem Punkt (. eingeben, öffnet sich automatisch ein Listenfeld mit der Liste aller Member aus dem Klassen-Tag:

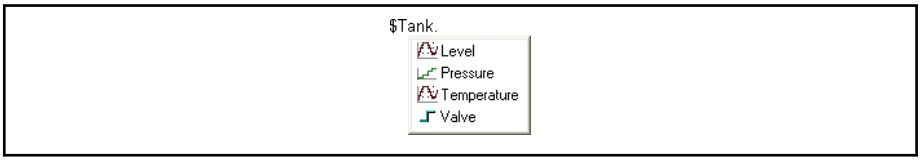


Abb.20-13: Member-Listenfeld mit "."-Zeichen öffnen

Wenn Sie den Namen eines Tags gefolgt von einem Bindestrich und dem Größer-als-Zeichen (->) in einer VBScript-Schnittstelle eingeben, öffnet sich automatisch ein Listenfeld mit einer Liste aller für diesen Tag verfügbaren Felder:

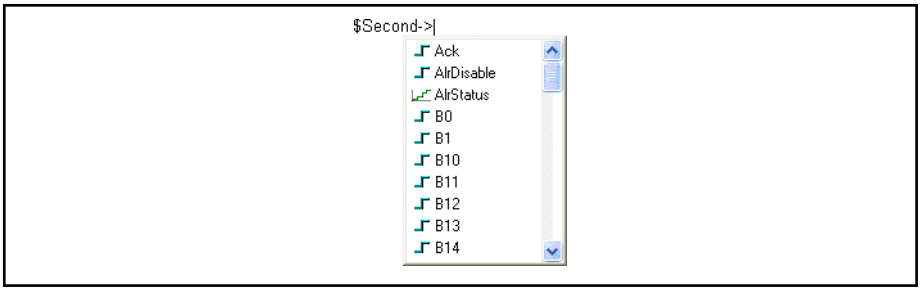


Abb.20-14: Listenfeld mit allen für einen Tag verfügbaren Felder mit "->" öffnen

Die Elemente werden in alphabetischer Reihenfolge angezeigt und jedes Element hat ein Symbol zur Anzeige seines Haupttyps, wie folgt:

Symbol	Typ
	Boolean-Tag
	Integer-Tag
	Real-Tag
	String-Tag
	Class-Tag
	Funktion aus der eingebauten Skriptsprache

Wenn Sie in einer VBScript-Schnittstelle aus dem Bild, in das das ActiveX-Objekt eingefügt ist, heraus den Name eines im Bild eingefügten ActiveX-Steuerelements gefolgt von einem Punkt (.) eingeben, öffnet sich automa-

tisch ein Listenfeld mit einer Liste aller Eigenschaften und Methoden dieses Objekts:

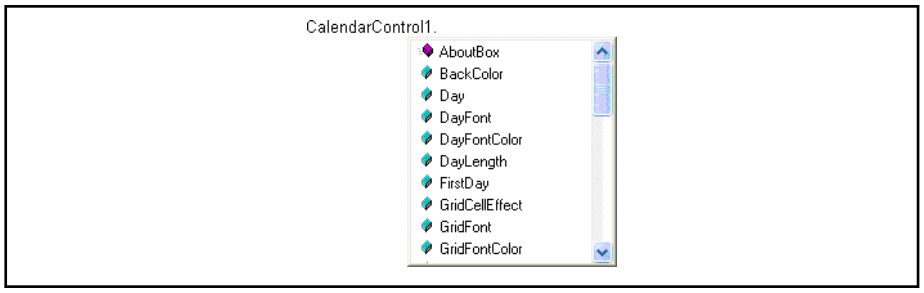


Abb.20-15: Listenfeld mit Eigenschaften und Methoden mit "."-Zeichen öffnen

Die Elemente werden in alphabetischer Reihenfolge angezeigt und jedes Element hat ein Symbol zur Anzeige seines Haupttyps, wie folgt:

Symbol	Typ
	Eigenschaften aus dem ActiveX-Objekt
	Methoden aus dem ActiveX-Objekt

Quick Info Parameter

Die Option Quick Info zeigt Popup-Boxen mit Informationen über die Funktionen aus der eingebauten Skriptsprache an. Die Information beinhaltet alle von dieser Funktion unterstützte Parameter mit den aktuell konfigurierten in Fettdruck.

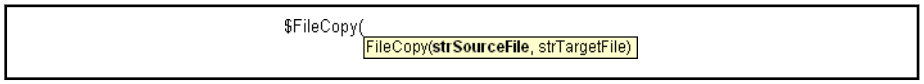


Abb.20-16: Quick Info Parameter

Complete Word

Complete Word vervollständigt Tags, Member, Felder, Funktionen oder ActiveX-Eigenschafts-/Methodennamen, sobald genügend eindeutige Zeichen für diesen Begriff gefunden wurden. Nach Eingabe der ersten Zeichen des Namens kann der Begriff durch Drücken von <Strg>+<Leertaste> vervollständigt werden.

20.4.3 VBScript im Vergleich mit VBA

Obwohl VBScript und Visual Basic for Applications (VBA) ähnlich sind und beide auf der Standardsprache Visual Basic basieren, gibt es für Benutzer von "WinStudio" Vorteile bei der Verwendung von VBScript:

- VBScript wird für das Betriebssystem Microsoft Windows Embedded Betriebssystem unterstützt und VBA nicht.
- VBScript bietet aktives Scripting für eine breite Vielzahl von Umgebungen, einschließlich Web-Client-Scripting im Microsoft Internet Explorer. Dies verhindert Vorgänge, die Risiken für den Thin-Client-Anwender darstellen können, z.B. direkten Zugriff auf lokale Dateien.
- VBScript wurde für einfaches und schnelles Erlernen entwickelt, wobei in VBScript einige Standards aus VBA vereinfacht wurden. So muss sich zum Beispiel der VBScript-Anwender beim Deklarieren von Variablen keine Gedanken um die verschiedenen Typen machen, weil VBScript automatisch den Typ für jede Variable annimmt.

Die folgende Tabelle zeigt VBScript-Merkmale, die in VBA nicht vorhanden sind.

Anhang: VBScript-Sprache

Kategorie	Merkmal/Schlüsselwort
Deklarierungen	Class
Verschiedenes	Eval
	Execute
Objekte	RegExp
Identifizierung Skript Engine	ScriptEngine
	ScriptEngineBuildVersion
	ScriptEngineMajorVersion

Die folgende Tabelle zeigt VBA-Merkmale, die in VBScript nicht vorhanden sind.

Kategorie	Wegfallendes Merkmal/Schlüsselwort
Handhaben von Bereichen	Option Base
	Deklarieren von Bereichen mit Untergrenze <> 0
Collection	Add, Count, Item, Remove
	Zugriff auf Kollektionen mit Zeichen !
Bedingte Kompilierung	#Const
	#If...Then...#Else
Steuerungsablauf	DoEvents
	GoSub...Return, GoTo
	On Error GoTo
	On...GoSub, On...GoTo
	Zeilennummern, Zeilenbezeichner
Konvertierung	CVar, CVDate
	Str, Val
Datentypen	Alle intrinsischen Datentypen außer Variant
	Type...End Type
Datum/Uhrzeit	Anweisung Datum, Anweisung Uhrzeit
DDE	LinkExecute, LinkPoke, LinkRequest, LinkSend
Fehlerbeseitigung	Debug.Print
	End, Stop
Deklarierung	Declare (zum Deklarieren von DLLs)
	Optional
	ParamArray
	Static
Fehlerbehandlung	Erl
	Fehler
	Resume, Resume Next

Kategorie	Wegfallendes Merkmal/Schlüsselwort
Dateieingabe/-ausgabe	Alle herkömmlichen Basic-Dateien E/A
Finanz	Alle Finanzfunktionen
Objekthandhabung	TypeOf
Objekte	Clipboard
	Collection
Operatoren	Like
Optionen	Deftype
	Option Base
	Option Compare
	Option Private Module
Fall auswählen	Ausdrücke mit dem Schlüsselwort "Is" oder beliebigen vergleichenden Operatoren
	Ausdrücke mit einer Reihe von Werten unter Verwendung des Schlüsselworts "To"
Strings	Strings mit fester Länge
	LSet, RSet
	Mid Statement
	StrConv
Verwenden von Objekten	Zugriff auf Kollektionen mit !

20.4.4 Bildereignisse

Zusätzlich zum Bildskript können Logiken mit der internen Skriptsprache für die Ereignisse "Beim Öffnen", "Während Geöffnet" und "Beim Schließen" für das Bild konfiguriert werden (siehe Schnittstelle Bildlogik aus dem Dialog Bildattribute). Beim Konfigurieren des Bildskripts (VBScript-Sprache) und der Bildlogik (eingebaute Skriptsprache) berücksichtigt "WinStudio" die folgende Ausführungsreihenfolge:

Ereignis	Ausführungsreihenfolge
Beim Öffnen des Bildes	• Subroutine Screen_OnOpen() aus der Bildskriptschnittstelle (VBScript-Sprache) • Beim Öffnen aus der Bildlogikschnittstelle (interne Skriptsprache)
Beim Schließen des Bildes	• Beim Schließen aus der Bildlogikschnittstelle (interne Skriptsprache) • Subroutine Screen_OnClose() aus der Bildskriptschnittstelle (VBScript-Sprache)

20.4.5 Funktionen MsgBox und InputBox

Mit den Funktionen **MsgBox()** und **InputBox()** aus der VBScript-Sprache können während der Laufzeit Popup-Meldungen angezeigt werden. Diese Funktionen sind synchron. Wenn eine dieser Funktionen ausgeführt wird, werden die verbleibenden Anweisungen aus dem Code erst dann ausgeführt, wenn die von der Funktion geöffneten Popup-Meldungen geschlossen wurden.

☞ Der in diesen Popup-Meldungen angezeigte Text ist vom Übersetzungswerkzeug von "WinStudio" nicht betroffen, es sei denn, der Text wird ausdrücklich mit der Funktion $Ext() aus der eingebauten Skriptsprache konfiguriert.

20.4.6 VBScript-Prozeduren

In VBScript gibt es zwei Arten von Prozeduren, die Prozedur Sub und die Prozedur Funktion.

Sub-Prozeduren

Eine Sub-Prozedur ist eine Serie von VBScript-Anweisungen (eingeschlossen in die Anweisungen **Sub** und **End Sub**), die Aktionen ausführen, aber keinen Wert zurückgeben. Eine Sub-Prozedur kann Argumente enthalten (Konstanten, Variablen oder Ausdrücke, die durch Aufruf einer Prozedur weitergegeben werden). Enthält eine Sub-Prozedur keine Argumente, muss die **Sub**-Anweisung einen leeren Klammersatz () enthalten.

Die folgende Sub-Prozedur verwendet zwei intrinsische (eingebaute) VBScript-Funktionen MsgBox und InputBox auf Seite 827, um einen Benutzer zu einer Eingabe aufzufordern. Dann werden die Ergebnisse einer Berechnung auf Grundlage dieser Information angezeigt. Die Berechnung wird in einer mit VBScript erstellten Funktionsprozedur ausgeführt. Die Funktionsprozedur wird nach der folgenden Betrachtung gezeigt.

Sub-Prozedur

```
Sub ConvertTemp()
  temp = InputBox("Please enter the temperature in degrees F.", 1)
  MsgBox "The temperature is " & Celsius(temp) & " degrees C."
End Sub
```

Funktionsprozeduren

Eine Funktionsprozedur ist eine Serie von VBScript-Anweisungen, eingeschlossen in die Anweisungen **Function** und **EndFunction**. Eine Funktionsprozedur gleicht einer Sub-Prozedur, kann aber zusätzlich einen Wert zurückgeben. Eine Funktionsprozedur kann Argumente enthalten (Konstanten, Variablen oder Ausdrücke, die durch eine aufgerufene Prozedur erhalten werden).

Enthält eine Function-Prozedur keine Argumente, muss die **Function**-Anweisung einen leeren Klammersatz () enthalten.

Eine Funktion gibt einen Wert zurück durch Zuweisung eines Wert zum Funktionsnamen in einer oder mehreren Anweisungen der Prozedur. Der Rückgabewert einer Funktion ist immer eine Variante.

Im folgenden Beispiel berechnet die Funktion **Celsius** Grad Celsius aus Grad Fahrenheit. Wenn die Funktion aus der Sub-Prozedur **ConvertTemp** aufgerufen wird, wird eine Variable mit dem Wert des Arguments an die Funktion weitergegeben. Das Ergebnis der Berechnung wird an die aufrufende Prozedur zurückgegeben und in einem Meldungsfenster angezeigt.

Funktionsprozedur

```
Sub ConvertTemp()
  temp = InputBox("Please enter the temperature in degrees F.", 1)
  MsgBox "The temperature is " & Celsius(temp) & " degrees C."
End Sub

Function Celsius(fDegrees)
  Celsius = (fDegrees - 32) * 5 / 9
End Function
```

Ein- und Ausgabe von Daten in bzw. aus Prozeduren

Jedes Datenelement wird durch ein Argument in die Prozeduren weitergegeben. Argumente dienen als Platzhalter für die Daten, die Sie in Ihre Prozedur weitergeben wollen. Argumente können jeden gültigen Variablennamen erhalten. Wird eine Prozedur mit der Anweisung **Sub** oder der Anweisung

Funktion erstellt, müssen nach dem Namen der Prozedur Klammern eingefügt werden. Alle Argumente werden in diese Klammern gesetzt, getrennt durch Kommas. Im folgenden Beispiel dient **fDegrees** als Platzhalter für den in die Funktion **Celsius** zur Umrechnung eingegebenen Wert.

Ein- und Ausgabe von Daten in bzw. aus Prozeduren

```
Function Celsius(fDegrees)
  Celsius = (fDegrees - 32) * 5 / 9
End Function
```

Um Daten aus einer Prozedur zu erhalten, muss eine Funktion verwendet werden. Eine Funktionsprozedur kann einen Wert zurückgeben; eine Sub-Prozedur kann dies nicht.

Verwendung von Sub- und Funktionsprozeduren im Code

Eine Funktion muss in Ihrem Code immer rechts von einer Variablenzuweisung oder in einem Ausdruck stehen. Zum Beispiel:

Verwenden von Funktionsprozeduren im Code

```
Temp = Celsius(fDegrees)
 // or
MsgBox "The Celsius temperature is " & Celsius(fDegrees) & " degrees."
```

Um eine Sub-Prozedur aus einer anderen Prozedur auszurufen, geben Sie den Namen der Prozedur zusammen mit den Werten für alle erforderlichen Argumente, getrennt durch Kommas, ein. Die Anweisung **Call** ist nicht erforderlich, wird sie aber verwendet, müssen alle Argumente in Klammern eingeschlossen werden.

Das folgende Beispiel zeigt zwei Aufrufe für die Prozedur **MyProc**. Eine verwendet die Anweisung **Call** im Code, die andere nicht. Beide haben genau die gleiche Wirkung.

Verwenden von Sub-Prozeduren in Code

```
Call MyProc(firstarg, secondarg)

MyProc firstarg, secondarg
```

Es ist zu beachten, dass die Klammern im Aufruf ausgelassen werden können, wenn die Anweisung **Call** nicht verwendet wird.

20.4.7 Erstellen von Konstanten

Eine Konstante ist ein sinnvoller Name, der die Stelle einer Zahl oder eines Strings annimmt und sich nie ändert. VBScript definiert eine Reihe von intrinsischen Konstanten.

Benutzerdefinierte Konstanten werden in VBScript mit der Anweisung Const erstellt.

Mit der Anweisung Const können String- oder numerische Konstanten mit sinnvollen Namen erstellt und ihnen literale Werte zugewiesen werden.

Zum Beispiel:

Const MeinString = "Dies ist mein Text."

Const MeinAlter = 49

Es ist zu beachten, dass die Zeichenkette in Anführungszeichen eingeschlossen wird (" "). Anführungszeichen sind der offensichtlichste Weg, String-Werte von numerischen Werten zu unterscheiden. Datenliterale und Zeitliterale können durch Einschließen in Rautenzeichen ("#") dargestellt werden.

Zum Beispiel:

Const EndTermin = #6-1-97#

Es ist sinnvoll, zum Unterscheiden von Konstanten und Variablen eine Namenskonvention zu erstellen. Dadurch wird vermieden, dass während der Ausführung des Skripts Konstanten neu zugewiesen werden. Sie können zum Beispiel die Vorsilben "vb" oder "con" vor den Konstantennamen setzen oder alle Konstanten in Großbuchstaben benennen. Eine klare Trennung von Konstanten und Variablen vermeidet Verwechslungen, wenn komplexere Skripte entwickelt werden.

20.4.8 Deklarieren von Variablen

Eine Variable ist ein sinnvoller Platzhalter, der sich auf einen Speicherort im Rechner bezieht, an dem die Programminformationen, die während der Ausführung des Scripts gegebenenfalls geändert werden sollen, gespeichert werden. In VBScript sind Variablen immer vom grundlegenden Datentyp Variant.

Variablen werden im Skript ausdrücklich mithilfe der Anweisung Dim, der Anweisung Public und der Anweisung Private deklariert.

Zum Beispiel:

```
Dim GradFahrenheit
```

Mehrere Variablen werden durch Trennung der einzelnen Variablenamen durch ein Komma deklariert.

Zum Beispiel:

```
Dim Oben, Unten, Links, Rechts
```

Eine Variable kann auch implizit deklariert werden, indem einfach deren Name im Skript verwendet wird. Dies ist generell keine gute Vorgehensweise, da der Name der Variablen an einer oder mehreren Stellen falsch geschrieben werden könnte, was zu unerwarteten Ergebnissen bei der Ausführung des Scripts führt. Daher wird die Anweisung Option Explicit standardmäßig in der Schnittstelle für Globale Prozeduren so konfiguriert, dass eine ausdrückliche Deklarierung aller Variablen benötigt wird. Wird diese Anweisung nicht gelöscht, müssen alle Variablen ausdrücklich deklariert werden; andernfalls erzeugt VBScript während der Laufzeit Fehler zur Angabe, dass die Variable nicht vorhanden ist.

In einem Ausdruck hat die Variable auf der linken Seite zu stehen und der Wert, der der Variablen zugewiesen werden soll, auf der rechten Seite.

Zum Beispiel:

```
MyVar = 100
```

20.4.9 Geltungsbereich und Lebensdauer von Variablen

Der Geltungsbereich einer Variablen wird dadurch bestimmt, wo sie deklariert wird. Wird eine Variable innerhalb einer Prozedur deklariert, kann nur Code aus dieser Prozedur auf diese Variable zugreifen oder deren Wert ändern. Sie hat einen lokalen Geltungsbereich und ist eine Variable auf Prozedurebene. Wird eine Variable außerhalb einer Prozedur deklariert, ist sie für alle Prozeduren in Ihrem Skript erkennbar. Es handelt sich um eine Variable auf Skriptebene und sie ist auf Skriptebene gültig.

Die Lebensdauer einer Variablen hängt davon ab, wie lange sie besteht. Die Lebensdauer einer Variablen auf Skriptebene erstreckt sich vom Zeitpunkt der Deklarierung bis zum Ende der Ausführung des Skripts. Auf Prozedurebene besteht eine Variable nur so lange, wie die Prozedur ausgeführt wird. Beim Verlassen der Prozedur wird die Variable zerstört. Lokale Variablen sind ideal als vorübergehender Speicherplatz bei der Ausführung einer Prozedur. Sie können lokale Variablen mit dem gleichen Namen in mehreren verschiedenen Prozeduren haben, da sie nur von der Prozedur erkannt wird, in der sie deklariert ist.

20.4.10 Boolesche Tags und Boolesche Variablen

Standardmäßig werden Boolesche Variablen in VBScript anders gehandhabt als Boolesche Tags auf Seite 124 in "WinStudio". Die Booleschen Zustände FALSE und TRUE haben die gleiche Bedeutung, aber die numerischen Werte von TRUE sind unterschiedlich, wie in der folgenden Tabelle gezeigt.

Boolescher Zustand	Numerischer Werte in...	
	Projekt-Tag	VBScript-Variable
FALSE	0	0
TRUE	1	−1



In VBScript sind "FALSE" und "TRUE" außerdem als Schlüsselworte auf Seite 818 reserviert.

Dieser Unterschied in der Handhabung von Booleschen Elementen kann das Laufzeitverhalten erheblich beeinflussen, wenn Sie VBScript in Ihrem Projekt verwenden. Logische und arithmetische Operationen – vor allem der Operator NOT auf Seite 833 – könnten Tag-Werte auf unerwartete Weise verändern.

Bei Bedarf können Sie dieses Verhalten durch Anpassung der Laufzeiteinstellungen des Projekts ändern.

Ändern der Handhabung von Booleschen Tags in VBScript

Die Handhabung von Booleschen Tags in VBScript kann durch Aktivierung des Modus „VB Boolean“ geändert werden. Dazu ändern Sie in Ihrer Projektdatei (**projekt_name.app**) die folgende Einstellung:

[Script]

VBBoolean=value

Wird **VBBoolean** auf **0** gesetzt, verhält sich das Projekt wie oben beschrieben: Alle VBScript-Funktionen und -vorgänge lesen/schreiben einen Wert von -1 für TRUE in Boolesche Tags. Dies ist die Standardeinstellung für mit "Rexroth WinStudio 7.0" oder früher erstellte und dann auf "Rexroth WinStudio 7.1" aktualisierte Projekte, um die Abwärtskompatibilität aufrechtzuerhalten.

Wird **VBBoolean** auf **1** gesetzt, liest/schreibt VBScript – weil es innerhalb von "WinStudio" implementiert ist – den Wert 1 für TRUE in Boolesche Tags. (Dies gilt nicht für Integer- oder Real-Tags.) Dies ist die Standardeinstellung für mit "Rexroth WinStudio 7.1" oder höher erstellte Projekte.



Beim Definieren einer benutzerdefinierten Eigenschaft auf einem Verknüpften Symbol auf Seite 220 unter Verwendung der Syntax **#Bezeichnung:@Zeiger** ist mit Vorsicht vorzugehen.

Zum Beispiel:

Syntax #Bezeichnung:@Zeiger

```
'The following statements are valid
If $MyBoolean = 1 Then
End If

If $MyBoolean = True Then
End If

If #Mne:@MyPointer = True Then
End If
```

```
'The following statement is invalid
If #Mne:@MyPointer = 1 Then
End If
```

20.4.11 Schreiben von reellen Werten in Integer-Tags

Standardmäßig wird ein reeller Wert (d.h. Gleitkommazahl) am Dezimaltrennzeichen abgeschnitten, wenn er in einen Integer-Tag auf Seite 124 geschrieben wird. Dieses Verhalten ist in der eingebauten Skriptsprache und in VBScript gleich.

In VBScript kann dieses Verhalten allerdings durch Deaktivieren der Laufzeiteinstellung „TruncRealToInt“ geändert werden. Dazu ändern Sie in Ihrer Projektdatei (**projekt_name.app**) die folgende Zeile:

[Script]

TruncRealToInt=value

Wird **TruncRealToInt** auf "1" gesetzt, verhält sich das Projekt wie oben beschrieben:

Reelle Werte werden ohne Runden am Dezimalzeichen abgeschnitten. (Zum Beispiel: Ein Wert von "5,56" wird als "5" in einen Integer-Tag geschrieben.) Dies ist die Standardeinstellung für mit "Rexroth WinStudio 7.1" oder früher erstellte und dann auf "Rexroth WinStudio 7.2" aktualisierte Projekte, um die Abwärtskompatibilität aufrechtzuerhalten.

Wird "TruncRealToInt" auf "0" gesetzt, runden die VBScript-Funktionen und -Operationen reelle Werte auf die nächste ganze Zahl.

(Zum Beispiel: Ein Wert von "5,56" wird als "6" in einen Integer-Tag geschrieben.) Dies ist die Standardeinstellung für mit "Rexroth WinStudio 7.2" oder höher erstellte Projekte.

Diese Einstellung beeinflusst nur das Verhalten von VBScript in "WinStudio". Das Verhalten der eingebauten Skriptsprache wird dadurch nicht beeinflusst.

20.4.12 Priorität von VBScript-Operatoren

VBScript verfügen über eine vollständige Bandbreite an Operatoren, darunter arithmetische Operatoren, vergleichende Operatoren, verknüpfende Operatoren und logische Operatoren.

Wenn mehrere Operationen in einem Ausdruck auftreten, wird jeder Teil gemäß einer als „Operatorenpriorität“ festgelegten Reihenfolge ausgewertet und aufgelöst. Um die Reihenfolge zu umgehen und die Auswertung bestimmter Teile eines Ausdrucks vor anderen Teilen zu erzwingen, können Klammern verwendet werden. Operationen in Klammern werden immer zuerst ausgeführt. Innerhalb der Klammern gilt aber weiterhin die Standardpriorität.

Enthalten Ausdrücke Operatoren aus mehr als einer Kategorie, werden arithmetische Operatoren zuerst ausgewertet, danach vergleichende Operatoren und zum Schluss logische Operatoren. Alle vergleichenden Operatoren haben die gleiche Priorität, d.h. sie werden von links nach rechts ausgewertet. Arithmetische und logische Operatoren werden in der folgenden Reihenfolge ausgewertet

arithmetisch	Vergleichend	Logisch
Negation (-)	Gleichheit (=)	Not
Potenzierung (^)	Ungleichheit (<>)	And
Multiplikation und Division (*, /)	Kleiner als (<)	Or

arithmetisch	Vergleichend	Logisch
Ganzzahldivision (\)	Größer als (>)	Xor
Absolutwertarithmetik (Mod)	Kleiner oder gleich (<=)	Eqv
Addition und Subtraktion (+, -)	Größer oder gleich (>=)	Imp
Zeichenkettenverknüpfung (&)	Is	&

Treten Multiplikation und Division zusammen in einem Ausdruck auf, wird jede Operation in der Reihenfolge von links nach rechts ausgewertet. Ebenso wird beim Auftreten von Addition und Subtraktion zusammen in einem Ausdruck jede Operation in der Reihenfolge von links nach rechts ausgewertet.

Der Operator zur Zeichenkettenverknüpfung (&) ist kein arithmetische Operator, liegt aber in der Reihenfolge nach allen arithmetischen Operatoren und vor allen vergleichenden Operatoren. Der Operator Is ist ein vergleichender Operator für Objektbezug. Er vergleicht Objekte nicht nach deren Wert, sondern prüft nur, ob zwei Objektbezüge sich auf das gleiche Objekt beziehen.

20.4.13 Logischer Operator NOT

Der logischen Operator NOT verhält sich in VBScript anders als in der eingebauten Skriptsprache.

Operator NOT in VBScript

In VBScript invertiert der Operator NOT die Bits eines gegebenen numerischen Werts und erstellt die Komplementärzahl gemäß des von Computern verwendeten „Zweierkomplement“-Systems von Zahlen mit Vorzeichen. Die folgende Tabelle zeigt das Verhalten des Operators NOT in VBScript für die Syntax...

ergebnis = NOT ausdruck

Wenn "Ausdruck" gleich...	Dann ist "Ergebnis"...
−3	2
−2	1
−1	0
0	−1
1	−2
2	−3
3	−4



Wenn versucht wird, einen numerischen Wert ungleich 0 in einen Booleschen Tag auf Seite 124 zu schreiben, nimmt der Tag standardmäßig automatisch den Wert 1 an.

Wenn daher der Operator NOT von VBScript auf einen Booleschen Tag mit dem Wert 1 angewendet wird, ändert sich der Tag-Wert nicht.

Der Operator gibt einen Wert von –2 zurück, der Tag kann diesen Wert aber nicht akzeptieren und nimmt wieder den Wert 1 an. "WinStudio" kann so konfiguriert werden, dass Boolesche Tags in VBScript wie Boolesche Variablen behandelt werden, so dass der Operator NOT in VBScript wie erwartet funktioniert. Weitere Informationen unter Boolesche Tags und Boolesche Variablen auf Seite 831.

Operator NOT in eingebauter Sprache

Dahingegen schaltet der Operator NOT auf Seite 559 in der eingebauten Skriptsprache den gegebenen numerischen Wert so um, als sei es ein natürlicher Boolescher Wert. Die folgende Tabelle zeigt das Verhalten des Operators NOT in der eingebauten Skriptsprache für die Syntax...

ergebnis = NOT ausdruck

Wenn "Ausdruck" gleich...	Dann ist "Ergebnis"...
0	1
≠0	0

20.4.14 Verwenden von bedingten Anweisungen

Der Ablauf Ihres Scripts kann mit bedingten Anweisungen und Schleifenanweisungen gesteuert werden. Mit bedingten Anweisungen kann VBScript-Code geschrieben werden, der Entscheidungen trifft und Aktionen wiederholt. Die folgenden bedingten Anweisungen stehen in VBScript zur Verfügung:

- Anweisung **If…Then…Else**
- Anweisung **Select Case**

Entscheidungen treffen mit If… Then…Else

Die Anweisung **If…Then…Else** wird verwendet zur Auswertung, ob eine Bedingung **Wahr** oder **Falsch** ist und je nach Ergebnis zur Angabe einer oder mehrerer auszuführenden Anweisungen. Üblicherweise ist die Bedingung ein Ausdruck, der einen vergleichenden Operator verwendet, um Werte bzw. Variablen miteinander zu vergleichen. Informationen über vergleichende Operatoren finden Sie unter Vergleichende Operatoren.

Anweisungen mit **If…Then…Else** können in beliebig viele Ebenen verschachtelt sein.

Ausführen von Anweisungen wenn eine Bedingung Wahr ist

Zum Ausführen einer einzelnen Anweisung, wenn eine Bedingung Wahr ist, wird die einzeilige Syntax für die Anweisung **If…Then…Else** verwendet. Das folgende Beispiel zeigt die einzeilige Syntax. Bitte beachten Sie, dass in diesem Beispiel das Schlüsselwort **Else** ausgelassen wurde:

*Einzeilige Syntax für **If…Then…Else***

```
Sub FixDate()
  Dim myDate
  myDate = #2/13/95#
  If myDate < Now Then myDate = Now
End Sub
```

Zum Ausführen von mehr als einer Codezeile muss die mehrzeilige oder Block-Syntax verwendet werden. Diese Syntax enthält wie im folgenden Beispiel gezeigt die Anweisung **End If**:

Mehrzeilige oder Block-Syntax

```
Sub AlertUser(value)
  If value = 0 Then
    AlertLabel.ForeColor = vbRed
    AlertLabel.Font.Bold = True
    AlertLabel.Font.Italic = True
  End If
End Sub
```

Ausführen von bestimmten Anweisungen, wenn eine Bedingung Wahr ist und Ausführen anderer Anweisungen, wenn eine Bedingung Falsch ist

Die Anweisung **If…Then…Else** kann zum Definieren von zwei Blöcken mit ausführbaren Anweisungen verwendet werden: Ein Block zum Ausführen, wenn die Bedingung **Wahr** ist und der andere Block zum Ausführen, wenn die Bedingung **Falsch** ist:

Anweisung If…Then…Else zum Definieren von zwei Blöcken...

```
Sub AlertUser(value)
  If value = 0 Then
    AlertLabel.ForeColor = vbRed
    AlertLabel.Font.Bold = True
    AlertLabel.Font.Italic = True
  Else
    AlertLabel.Forecolor = vbBlack
    AlertLabel.Font.Bold = False
    AlertLabel.Font.Italic = False
  End If
End Sub
```

Entscheiden zwischen mehreren Alternativen

Mit einer Variante der Anweisung **If…Then…Else** kann aus mehreren Alternativen gewählt werden. Durch Hinzufügen des Satzes **ElseIf** wird die Funktionalität der Anweisung **If…Then…Else** erweitert, so dass der Programmablauf aufgrund verschiedener Möglichkeiten gesteuert werden kann. Zum Beispiel:

Anweisung If…Then…Else mit ElseIf

```
Sub ReportValue(value)
  If value = 0 Then
    MsgBox value
  ElseIf value = 1 Then
    MsgBox value
  ElseIf value = 2 then
    Msgbox value
  Else
    Msgbox "Value out of range!"
  End If
End Sub
```

Es können so viele **ElseIf**-Sätze wie für die Alternativen benötigt hinzugefügt werden, allerdings kann die übermäßige Verwendung von **ElseIf**-Sätzen behindernd wirken. Die Anweisung **Select Case** ist besser zum Auswählen zwischen verschiedenen Alternativen geeignet.

Entscheidungen treffen mit Select Case

Die Struktur "Select Case" stellt eine Alternative zu **If…Then…Else** da, die für das selektive Ausführen eines Anweisungsblocks aus mehreren Anweisungsblocks verwendet wird. Eine Anweisung **Select Case** bietet ähnliche Möglichkeiten wie die Anweisung **If…Then…Else**, macht den Code aber effizienter und leichter lesbar.

Eine Struktur mit **Select Case** arbeitet mit einem einzelnen Testausdruck, der am Anfang der Struktur einmal ausgewertet wird. Das Ergebnis des Aus-

drucks wird dann mit den Werten für jeden **Case** (Fall) in der Struktur verglichen. Wird eine Übereinstimmung gefunden, wird der mit diesem **Case** verbundene Anweisungsblock ausgeführt, wie im folgenden Beispiel dargestellt:

Struktur Select Case

```
Select Case Document.Form1.CardType.Options(SelectedIndex).Text
  Case "MasterCard"
    DisplayMCLogo
    ValidateMCAccount
  Case "Visa"
    DisplayVisaLogo
    ValidateVisaAccount
  Case "American Express"
    DisplayAMEXCOLogo
    ValidateAMEXCOAccount
  Case Else
    DisplayUnknownImage
    PromptAgain
End Select
```

Beachten Sie, dass die Struktur **Select Case** einen Ausdruck einmal am Anfang der Struktur auswertet. Die Struktur **If…Then…ElseIf** hingegen kann für jede "ElseIf"-Anweisung einen anderen Ausdruck auswerten. Die Struktur **If… Then…ElseIf** kann nur dann durch eine Struktur **Select Case** ersetzt werden, wenn jede **ElseIf**-Anweisung den gleichen Ausdruck auswertet.

20.4.15 Schleifen mit Code

Mit Schleifen kann eine Gruppe von Anweisungen wiederholt ausgeführt werden. Einige Schleifen wiederholen Anweisungen, bis eine Bedingung Falsch ist; andere Schleifen wiederholen Anweisungen, bis eine Bedingung Wahr ist. Außerdem gibt es Schleifen, die eine bestimmte Anzahl von Wiederholungen für die Anweisungen durchführen.

Die folgenden Schleifenanweisungen stehen in VBScript zur Verfügung:

- **Do…Loop**: Schleife während oder bis eine Bedingung Wahr ist
- **While…Wend**: Schleife während eine Bedingung Wahr ist
- **For…Next**: Verwendung eines Zählers zum Ausführen einer bestimmten Anzahl von Wiederholungen der Anweisungen

Verwenden von Schleifen mit Do Loop

"Do...Loop"-Anweisungen können zum unendlichen Ausführen eines Anweisungsblocks verwendet werden. Die Anweisungen werden wiederholt, während eine Bedingung Wahr ist oder bis eine Bedingung Wahr wird.

Wiederholen von Anweisungen während eine Bedingung Wahr ist

Verwenden Sie das Schlüsselwort While zum Überprüfen einer Bedingung in einer Anweisung mit **Do...Loop**. Die Bedingungen kann vor Betreten der Schleife (wie im folgenden Beispiel **ChkFirstWhile** gezeigt) oder nach mindestens einem Ausführungszyklus der Schleife (wie im Beispiel **ChkLastWhile** gezeigt) geprüft werden. In der Prozedur **ChkFirstWhile** ist **myNum** auf 9 anstelle von 20 gesetzt, die Anweisungen innerhalb der Schleife werden niemals ausgeführt. In der Prozedur **ChkLastWhile** werden die Anweisungen innerhalb der Schleife nur einmal ausgeführt, weil die Bedingung bereits Falsch ist.

Wiederholen von Anweisungen während eine Bedingung Wahr ist

```
Sub ChkFirstWhile()
  Dim counter, myNum
  counter = 0
  myNum = 20
  Do While myNum > 10
    myNum = myNum - 1
    counter = counter + 1
  Loop
```

```
  MsgBox "The loop made " & counter & " repetitions."
End Sub

Sub ChkLastWhile()
  Dim counter, myNum
  counter = 0
  myNum = 9
  Do
    myNum = myNum - 1
    counter = counter + 1
  Loop While myNum > 10
  MsgBox "The loop made " & counter & " repetitions."
End Sub
```

Wiederholen einer Anweisung bis eine Bedingung Wahr wird

Das Schlüsselwort **Until** kann auf zwei Weisen verwendet werden, um die Bedingung in einer Anweisung mit **Do...Loop** zu prüfen. Die Bedingungen kann vor Betreten der Schleife (wie im folgenden Beispiel **ChkFirstUntil** gezeigt) oder nach mindestens einem Ausführungszyklus der Schleife (wie im Beispiel **ChkLastUntil** gezeigt) geprüft werden. Die Schleife wird ausgeführt, solange die Bedingung **Falsch** ist.

Wiederholen einer Anweisung bis eine Bedingung Wahr wird

```
Sub ChkFirstUntil()
  Dim counter, myNum
  counter = 0
  myNum = 20
  Do Until myNum = 10
    myNum = myNum - 1
    counter = counter + 1
  Loop
  MsgBox "The loop made " & counter & " repetitions."
End Sub

Sub ChkLastUntil()
  Dim counter, myNum
  counter = 0
  myNum = 1
  Do
    myNum = myNum + 1
    counter = counter + 1
  Loop Until myNum = 10
  MsgBox "The loop made " & counter & " repetitions."
End Sub
```

Verlassen einer Anweisung mit Do...Loop aus der Schleife heraus

Ein **Do...Loop** kann mit der Anweisung **Exit Do** verlassen werden. Da man eine Schleife normalerweise nur in bestimmten Situationen, wie zum Beispiel zum Vermeiden einer Endlosschleife, verlassen möchte, sollte die Anweisung **Exit Do** im Anweisungsblock **True** einer Anweisung mit **If...Then...Else** verwendet werden. Wenn die Bedingung **Falsch** ist, läuft die Schleife normal.

Im folgenden Beispiel ist "myNum" ein Wert zugewiesen, der eine Endlosschleife erzeugt. Die Anweisung **If...Then...Else** sucht nach dieser Bedingung und verhindert die Endloswiederholung.

Verlassen einer Anweisung mit Do...Loop aus der Schleife heraus

```
Sub ExitExample()
  Dim counter, myNum
  counter = 0
  myNum = 9
  Do Until myNum = 10
    myNum = myNum - 1
    counter = counter + 1
    If myNum < 10 Then Exit Do
  Loop
```

```
  MsgBox "The loop made " & counter & " repetitions."
End Sub
```

Verwenden von While…Wend

Die Anweisung **While...Wend** wird von VBScript für entsprechende erfahrene Anwender zur Verfügung gestellt. Da **While...Wend** aber nicht flexibel ist, wird die Verwendung von **Do...Loop** empfohlen.

Verwenden von For…Next

For...Next-Anweisungen können zum Ausführen einer bestimmten Anzahl von Wiederholungen eines Anweisungsblocks verwendet werden. Für Schleifen wird eine Zählervariable verwendet, deren Wert mit jeder Wiederholung der Schleife abnimmt oder ansteigt.

Mit dem folgenden Beispiel wird eine Prozedur mit dem Namen **MyProc** 50 mal ausgeführt. Die Anweisung For legt die Zählervariable **x** und deren Start- und Endwerte fest. Die Anweisung "Next" erhöht die Zählervariable um 1.

Verwenden von For…Next (1)

```
Sub DoMyProc50Times()
  Dim x
  For x = 1 To 50
    MyProc
  Next
End Sub
```

Unter Verwendung des Schlüsselworts **Step** kann die Zählervariable um den von Ihnen angegebenen Wert erhöht oder verringert werden. Im folgenden Beispiel wird die Zählervariable **j** bei jeder Wiederholung der Schleife um 2 erhöht. Wenn die Schleife beendet ist, ist die Gesamtsumme die Summe von 2, 4, 6, 8 und 10.

Verwenden von For…Next (2)

```
Sub TwosTotal()
  Dim j, total
  For j = 2 To 10 Step 2
    total = total + j
  Next
  MsgBox "The total is " & total
End Sub
```

Zum Verringern der Zählervariable wird ein negativer Wert für **Step** verwendet. Es muss ein Endwert angegeben werden, der kleiner als der Startwert ist. Im folgenden Beispiel wird die Zählervariable **myNum** bei jeder Wiederholung der Schleife um 2 verringert. Wenn die Schleife beendet ist, ist die Gesamtsumme die Summe von 16, 14, 12, 10, 8, 6, 4 und 2.

Verwenden von For…Next (3)

```
Sub NewTotal()
  Dim myNum, total
  For myNum = 16 To 2 Step -2
    total = total + myNum
  Next
  MsgBox "The total is " & total
End Sub
```

Die Anweisung **For...Next** kann jederzeit vor Erreichen des Endwerts des Zählers durch Verwendung der Anweisung **Exit For** verlassen werden. Da man eine Schleife normalerweise nur in bestimmten Situationen, wie zum Beispiel bei Auftreten eines Fehlers, verlassen möchte, sollte die Anweisung **Exit For** im Anweisungsblock **True** einer Anweisung mit **If...Then...Else** verwendet werden. Wenn die Bedingung **Falsch** ist, läuft die Schleife normal.

20.4.16 Unterstützung für ActiveX-Steuerelemente

Mit den VB-Script-Schnittstellen für die Grafikmodule (Grafikskript, Bildskript, Befehlsanimation und ActiveX-Ereignisse) kann diese Syntax für den Zugriff auf Eigenschaften und Methoden direkt von jedem ActiveX-Steuerobjekt im Bild, in dem das Objekt konfiguriert ist, verwendet werden.

"WinStudio" weist dem Objekt im Bild einen eindeutigen Namen zu. Sie können die Eigenschaft Name (im Dialog **Objekteigenschaften**) zum Verändern dieses Namens verwenden.

Nach dem Einfügen des ActiveX-Steuerobjekts in das Bild kann von jeder mit diesem Bild verbundenen VBScript-Schnittstelle auf die Eigenschaften und Methoden aus diesem Objekt zugegriffen werden. Verwenden Sie die Syntax **Objekt_Name.Eigenschaften_oder_Methoden_Name**.

Beispiele:

```
//Access the value of the property Day from the CalendarControl1 ActiveX object
CalendarControl1.Day

//Triggers the method AboutBox from the CalendarControl1 ActiveX object
CalendarControl1.AboutBox
```

20.4.17 Unterstützung durch Windows Embedded

CEView unterstützt ebenfalls VBScript. Der Hardwarehersteller des Windows-Embedded-Geräts muss die Unterstützung für VBScript darauf freigeben, damit CEView die in der VBScript-Sprache auf dem Gerät konfigurierten Skripte ausführen kann.

Die Funktionen **MsgBox()** und **InputBox()** können vom Hardwarehersteller bei der Erstellung des Image für das Windows-Embedded-Gerät gezielt aktiviert/deaktiviert werden.

Wenn Sie nicht sicher sind, ob das auf Ihrem Gerät geladene Image VBScript unterstützt, wenden Sie sich bitte an den Hardwarehersteller.

21 Anhang: Treiber-Fehlercodes

Die Felder **Status Lesen** und **Status Schreiben** geben Fehlercodes an die für diese Felder festgelegten Tags aus. Diese Fehlercodes können Standardcodes sein oder für jeden Treiber angepasst werden. Beschreibungen für angepasste Fehlercodes finden Sie in der Dokumentation für den Treiber.

Die folgenden Werte sind Standardfehlercodes, die immer negative Werte haben:

Wert	Beschreibung
0	OK
-1	ungültiger serieller Port
-2	ungültige Baudrate
-3	ungültige Bit-Anzahl
-4	ungültige Stop-Bit-Anzahl
-5	ungültige Parität
-6	ungültige Unterbrechungsanforderung
-7	serieller Port bereits in Verwendung
-8	ungültige Puffergröße
-9	nicht genügend Speicherplatz
-10	TX-Puffer leer
-11	TX-Puffer voll
-12	RX-Puffer leer
-13	RX-Puffer voll
-14	Zeitüberschreitung bei Warten CTS
-15	Zeitüberschreitung beim Warten auf Beginn einer Meldung
-16	Zeitüberschreitung beim Warten auf Ende einer Meldung
-17	Zeitüberschreitung zwischen RX char
-18	Zeitüberschreitung zwischen TX char
-19	Kein Träger erkannt
-20	Kein DSR erkannt
-21	Konnte keine 8250 in Adresse finden
-22	TX-Leitung besetzt
-23	Abbruch durch Benutzer
-24	Funktion wird nicht unterstützt
-25	Überlauf
-26	Parität
-27	Überlauf und Parität
-28	Framing
-29	Framing und Überlauf

Anhang: Treiber-Fehlercodes

Wert	Beschreibung
-30	Framing und Parität
-31	Framing, Überlauf und Parität
-32	Zeitüberschreitung beim Warten auf Ende einer TX-Meldung

Die Datei **UNICOMM.MSG** im Verzeichnis **[...]\WinStudio\Bin** enthält auch Beschreibungen der Standardfehlercodes.

Die Datei **DriverName.MSG** im Verzeichnis **[...]\WinStudio\Drv** enthält Beschreibungen der angepassten Fehlercodes für jeden Treiber.

22 Service und Support

Für Ihre schnelle und optimale Unterstützung verfügen wir über ein dichtes weltweites Servicenetz. Unsere Experten stehen Ihnen mit Rat und Tat zur Seite. Sie erreichen uns täglich **rund um die Uhr - auch am Wochenende und an Feiertagen**.

Service Deutschland

Unser technologieorientiertes Competence Center in Lohr deckt alle Belange rund um den Service für elektrische Antriebe und Steuerungen ab.

Die **Service Helpdesk & Hotline** erreichen Sie unter:

Telefon:	**+49 9352 40 5060**
Fax:	**+49 9352 18 4941**
E-Mail:	service.svc@boschrexroth.de
Internet:	http://www.boschrexroth.com

Auf unseren Internetseiten finden Sie ergänzende Hinweise zu Service, Reparatur (z.B. Anlieferadressen) und Training.

Service weltweit

Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit Ihrem Ansprechpartner auf. Die Hotline-Rufnummern entnehmen Sie bitte den Vertriebsadressen im Internet.

Vorbereitung der Informationen

Wir können Ihnen schnell und effizient helfen, wenn Sie folgende Informationen bereithalten:

- detaillierte Beschreibung der Störung und der Umstände
- Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern
- Ihre Kontaktdaten (Telefon-, Faxnummer und E-Mail-Adresse)

Index

Symbole
.NET-Controls ... 236

A
Abs ... 559
ACos ... 600
ActiveX- und .NET-Control-Funktionen ... 763
ActiveX-Control ... 229
ActiveX-Ereignisse ... 805
Aktive Objekte
  Combobox ... 206
  Drucktaste ... 193
  Intelligente Meldung ... 214
  Kontrollkästchen ... 198
  Listenfeld ... 210
  Radiobutton ... 201
  Schaltfläche ... 191
  Text ... 190
Alarm-/Ereignissteuerungsobjekt ... 290
Alarmarbeitsblatt ... 273
  E-Mail-Einstellungen ... 276
  Weitere Einstellungen ... 279
Alarme, Ereignisse und Trends ... 273
Alarmhistorie
  Format ... 282
An- / Abmelden ... 432
Ändern des Produkttyps eines bestehenden Projekts ... 101
Anhang
  Built-in Skriptsprache ... 545
  Treiber-Fehlercodes ... 841
  VBScript ... 797
Anhang_A ... 335
Anhang_B ... 339
Animation Drehen ... 256
Animation Größenänderung ... 254
Animation Text-Daten-Verknüpfung ... 249
Animationen ... 178
  Balkendiagramm ... 248
  Befehl ... 243
  Drehen ... 256
  Farbe ... 251
  Größenänderung ... 254
  Hyperlink ... 247
  Position ... 253
  Text-Daten-Verknüpfung ... 249
Anlegen eines neuen Projekts ... 99
Anpassen der Größe eines Bildobjekts ... 263
Anpassen der Position eines Bildobjekts ... 263
Anweisungen ... 814
Anwenden von Animationen auf Bildobjekte ... 243
Apache für Windows ... 470
AppActivate ... 622
AppIsRunning ... 624
AppPostMessage ... 624
AppSendKeys ... 625
Arithmetische Funktionen ... 558
Arithmetische Operatoren ... 559
Array-Tags ... 125
Arten des Lizenzschutzes ... 47
Asc2Str ... 579
ASin ... 600
Ausführen ... 65
Ausführen eines Projekts ... 490
Ausführungsmodi ... 48
Ausgabe (LogWin) ... 443
Ausgabefenster LogWin ... 93
Ausrichten, Zentrieren und Verteilen ... 260
Autodeklaration
  Tags im WinStudio ... 134
AutoFormat ... 674
Automatische Übersetzung ... 437
Automatisches Ausführen eines Projekts ... 490
Avg ... 570

B
Balkendiagramm-Animation ... 248
Batch-Datei
  Aufruf einer externen Datei ... 652
Bearbeiten Gruppe
  Ersetzen von Projekt-Tags ... 149, 179
Bedingte Anweisungen ... 834
Befehls-Animation ... 243
Befehlsanimation ... 804
Berechnungs-Arbeitsblatt ... 367
Bibliotheken
  .NET-Controls ... 236
  ActiveX-Control ... 229
  Symbole ... 219, 220
Bild-/Arbeitsblatt-Editor ... 92
Bilder und Grafiken ... 169
Bilder, Symbole und Arbeitsblätter
  Passwortgeschützt ... 434
Bildereignisse ... 827
Bildfunktionen ... 603
Bildobjekte ... 178
  Aktive Objekte ... 190
  Animationen ... 243
  Formen ... 182
  Übersetzung deaktivieren ... 440
Bildobjekten bearbeiten ... 178
Bildskript ... 803
BlockUser ... 610
Boolesche Tags ... 831
Boolesche Variablen ... 831

C
CEWebServer ... 468
CharToValue ... 579
CharToValueW ... 580

Index

CheckESign ... 611
ClassMembersToStrVector ... 581
CleanReadQueue ... 626
Client/Server-Architektur ... 458
ClockGetDate ... 595
ClockGetDayOfWeek ... 596
ClockGetTime ... 597
Close ... 603
CloseSplashWindow ... 626
CnfEmail ... 737
CnfFTP ... 767
Control Layer
  ISSymbol ... 455
Cos ... 601
Cot ... 601
CreateUser ... 611

D
Dateifunktionen ... 653
Datenbank-/ERP-Arbeitsblatt ... 371
Datenbank-/ERP-Funktionen ... 772
Datenbankschnittstelle ... 493
Datenbankspion ... 93, 445
  Remote ... 447
Datenblatt Projekt-Tags
  Erweitern ... 118
Datenserver
  Konfigurieren ... 463
Datentypen
  WinStudio ... 124
DateTime2Clock ... 597
Datums- und Zeitfunktionen ... 595
DBCursorClose ... 773
DBCursorColumnCount ... 774
DBCursorColumnInfo ... 775
DBCursorCurrentRow ... 776
DBCursorGetValue ... 776
DBCursorMoveTo ... 777
DBCursorNext ... 778
DBCursorOpen ... 779
DBCursorOpenSQL ... 781
DBCursorPrevious ... 783
DBCursorRowCount ... 783
DBDelete ... 784
DBExecute ... 785
DBInsert ... 787
DBSelect ... 788
DBUpdate ... 790
DbVersion ... 688
DeleteOlderFiles ... 654
DialError ... 744
DialGetClientIP ... 750
DialGetServerIP ... 751
Dialog Bildeigenschaften ... 170
Dialog Objekteigenschaften ... 178
Dialog Sicherheitssystem ... 430
DialStatus ... 753
DialUp ... 755
DialUpToCE ... 756
DirCreate ... 655
DirDelete ... 655
DirLength ... 656
DirRename ... 657
DisableMath ... 627
Div ... 560
Drehen ... 262

E
E-Mail-Einstellungen für Alarmarbeitsblatt ... 276
E-Mail-Funktionen ... 737
Eigenschaften ... 813
Einwahlfunktionen ... 743
email
  Konfigurieren ... 737
EnableMath ... 627
EndTask ... 627
Entfernen nicht verwendeter Tags ... 150
Entwicklungsstation
  Konfigurieren ... 489
Entwicklungsumgebung
  Winstudio ... 53
Ereignisprotokoll
  Format ... 288
Ereignisprotokollierung ... 285
Ereignisse und Ereignisprotokolle ... 285
Ersetzen von Projekt-Tags in einem Dokument oder Bildobjekt ... 149
Erstellen und Konfigurieren von Benutzern ... 428
Erstellen und Konfigurieren von Gruppen ... 423
ExecuteAlarmAck ... 715
ExitWindows ... 628
Exp ... 573
ExportSecuritySystem ... 612
Ext ... 684

F
False ... 576
Farb-Animation ... 251
Farbschnittstelle ... 265
Fehler ... 818
Fehlerbeseitigung ... 533
Fehlerbeseitigung Datenbank ... 509
Fehlercodes
  Treiber ... 841
FileCopy ... 657
FileDelete ... 658
FileLength ... 659
FileRename ... 659
FileWrite ... 659
FindAllDevices ... 758
FindFile ... 661
FindModem ... 758
FindPath ... 662
For … Next ... 718

ForceTagChange... 716
Format... 560
FTP-Funktionen... 767
FTPGet... 768
FTPPut... 770
FTPStatus... 771
Füllfarbe... 264
Funktionen... 817
Funktionen MsgBox und InputBox... 827
Funktionen zur Ereignisprotokollierung... 765
Funktionen zur Modulaktivität... 621
Funktionsbeschreibungen
Lesen... 556

G
Geltungsbereich von Variablen... 830
Geschlossenes Polygonobjekt... 184
GetAppHorizontalResolution... 689
GetAppPath... 689
GetBit... 564
GetClock... 598
GetComputerIP... 690
GetComputerName... 690
GetCursorX... 691
GetCursorY... 691
GetDisplayHorizontalResolution... 691
GetDisplayVerticalResolution... 692
GetFileAttributes... 662
GetFileTime... 663
GetFreeMemoryCE... 692
GetHardKeyModel... 693
GetHardKeySN... 693
GetHstInfo... 663
GetIPAll... 694
GetLine... 664
GetMemoryCE... 694
GetNetMACID... 695
GetOS... 696
GetPrivateProfileString... 696
GetProductPath... 697
GetRegValue... 697
GetRegValueType... 698
GetScrInfo... 675
GetSecuritySystemStatus... 613
GetServerHostName... 699
GetStatusSendEmailExt... 740
GetTagValue... 716
GetTickCount... 699
GetUserFullName... 615
GetUserNames... 615
GetUserPwdAging... 616
GetUserState... 617
Gliederung der Dokumentation... 19
Globale Prozeduren... 799
Grafik
Aktive Objekte... 190
Animationen... 243
Grafiken
Formen... 182
Grafische Funktionen... 674
Gruppe Bearbeiten
Anordnen von Objekten... 180
Deaktivieren von Ziehen in ein Bild... 181
Objekteigenschaften... 178
Rastereinstellungen... 181
Rückgängig... 181
Gruppe Formen... 182
Geschlossenes Polygonobjekt... 184
Hinzufügen von Bildern zu einem Bild... 189
Objekt abgerundetes Rechteck... 187
Objekt Ellipse... 188
Objekt Linie... 182
Objekt Rechteck... 185
Offenes Polygonobjekt... 183
Gruppieren und Gruppierung aufheben... 259

H
HangUp... 759
Häufig gestellte Fragen... 534
Hauptteil Alarmarbeitsblatt... 280
Hilfsmittel Globales Ersetzen... 148
Hilfsmittel Objektsuche... 152
Hilfsmittel Tag-Eigenschaften... 153
Hintergrund-Task
Ausführen... 35
Wechseln... 35
Hintergrundbild... 174
Hintergrundfarbe eines Bildes... 174
Hintergrundtasks... 357
Hinzufügen einer Verknüpfung zu einer externen Bilddatei... 242
Horizontal spiegeln... 263
Hour2Clock... 598
HST2TXT... 667
HST2TXTIsRunning... 668
Hyperlink-Animation... 247

I
If... 576
IIS... 468
Import-Assistent... 154
Importieren aus anderen Projektdatenbanken... 157
Importieren aus CSV-Datenbanken... 161
Importieren aus ODBC-Datenbanken... 162
Importieren aus OPC-Server-Datenbanken, Seite ... 159
ImportSecuritySystem... 617
In den Vordergrund und In den Hintergrund... 258
In dieser Dokumentation verwendete Konventionen... 22
Indirekte Tags (Zeiger)... 131
InfoAppAlrDir... 700
InfoAppHSTDir... 700
InfoDiskFree... 700

Index

InfoResources ... 701
Installation ... 39
  Systemanforderungen ... 39
Installation der Software ... 40
Integrieren von Tags ... 163
IntelliSense
  VBScript Editor ... 823
IsActiveXReg ... 701
IsAppChangedOnServer ... 702
IsScreenOpen ... 629
ISSymbol Control Layer ... 455
IsTaskRunning ... 629
IsViewerInFocus ... 630

K
KeyPad ... 630
Klassen
  Erstellen ... 135
  Tag ... 128
Kommunikation mit anderen Geräten ... 379
Kommunikation mit einem Remote-Gerät ... 379
  Standardtreiberblätter ... 387
Konfigurieren einer DDE-Client-Verbindung mit einem DDE-Server ... 403
Konfigurieren einer OPC-Client-Verbindung mit einem OPC-Server ... 390
Konfigurieren einer OPC-UA die-Client-Verbindung mit einem OPC-UA-Server ... 395
Konfigurieren einer TCP/IP-Client-Verbindung zu einem anderen Projekt ... 401
Konstanten ... 809
  Erstellen ... 829
Kontrollkästchen
  Betriebsmodi ... 204
Konvertieren von Trendhistoriendateien
  von binär in Text ... 303
  von Text in binär ... 304

L
Laufzeitschnittstelle Trendobjekt ... 310
Lebensdauer von Variablen ... 830
Linienfarbe ... 265
Liste der System-Tags ... 821
Liste der verfügbaren Funktionen ... 545
Lizenz
  ungültig ... 51
Lizenzeinstellungen ... 47
Lizenzierung ... 47
Log ... 573
Log10 ... 574
Logarithmische Funktionen ... 573
Logische Funktionen ... 574
Logische Operatoren ... 574
Logischer Operator NOT ... 833
LogOff ... 632
LogOn ... 632
LogWin
  Remote ... 448
LookupContains ... 669
LookupGet ... 670
LookupLoad ... 670
Löschen eines Tags
  aus der Projektdatenbank ... 148

M
Math ... 633
Max ... 571
Methoden ... 817
Microsoft Access Datenbanken
  Verwendung ... 520
Microsoft Excel
  Verwendung ... 526
Microsoft SQL Server
  Verwendung ... 515
Min ... 571
Mobile Access ... 467
Mod ... 564
Multimediafunktionen ... 686
MySQL
  Verwendung ... 530

N
Nach Hinten, eine Ebene ... 259
Nach Vorn, eine Ebene ... 259
Namenssyntax für Tags ... 121
NCopy ... 582
Next ... 718
NoInputTime ... 703
NTWebServer ... 468
Num ... 583

O
Objekt abgerundetes Rechteck ... 187
Objekt Combobox ... 206
Objekt Drucktaste ... 193
Objekt Ellipse ... 188
Objekt intelligente Meldung ... 214
Objekt Kontrollkästchen ... 198
Objekt Linie ... 182
Objekt Listenfeld ... 210
Objekt Radiobutton ... 201
Objekt Rechteck ... 185
Objekt Schaltfläche ... 191
Objekt Text ... 190
Objekte und Kollektionen ... 813
Objekteigenschaften
  .NET-Controls ... 236
  Abgerundetes Rechteck ... 187
  ActiveX-Control ... 229
  Animation ... 256
  Balkendiagramm ... 248
  Befehl ... 243
  Combobox ... 206
  Drucktaste ... 193

Ellipse ... 188
Farben ... 251
Geschlossenes Polygon ... 184
Größe ... 254
Hyperlink ... 247
Intelligente Meldung ... 214
Kontrollkästchen ... 198
Linie ... 182
Listenfeld ... 210
Offenes Polygon ... 183
Position ... 253
Radiobutton ... 201
Rechteck ... 185
Schaltfläche ... 191
Text ... 190
Text-Daten-Verknüpfung ... 249
ODBC-Arbeitsblatt ... 364
ODBC-Datenbanken
Verwendung ... 513
ODBC-Funktionen ... 719
ODBCBeginTrans ... 720
ODBCBindCol ... 721
ODBCCanAppend ... 722
ODBCCanUpdate ... 723
ODBCClose ... 723
ODBCCommitTrans ... 724
ODBCDelete ... 724
ODBCExecuteSQL ... 725
ODBCInsert ... 725
ODBCIsBOF ... 726
ODBCIsDeleted ... 727
ODBCIsEOF ... 727
ODBCIsFieldNULL ... 728
ODBCIsFieldNullable ... 728
ODBCMove ... 729
ODBCMoveFirst ... 729
ODBCMoveLast ... 730
ODBCMoveNext ... 730
ODBCMovePrev ... 731
ODBCOpen ... 732
ODBCQuery ... 733
ODBCRollBack ... 733
ODBCSetFieldNULL ... 734
ODBCSetFilter ... 734
ODBCSetSort ... 735
ODBCUnbindCol ... 736
ODBCUpdate ... 736
Offenes Polygonobjekt ... 183
Open ... 604
OpenPrevious ... 606
Operatoren ... 808
arithmetische ... 559
Logisch ... 574
Optionsfeld
Betriebsmodi ... 204
ORACLE-Datenbanken
Verwendung ... 517
Ordner
Gemeinsame Datenbank ... 120
Klassen ... 118
Projekt-Tags ... 116
System-Tags ... 121
Ordner Bildgruppen ... 175
Ordner Gemeinsame Datenbank ... 120
Ordner Klassen ... 118
Ordner Projekt-Tags ... 116
Ordner System-Tags ... 121
Ordner Webseiten ... 176
Ordner WinStudio ... 170

P

Passwortschutz
Bilder, Symbole und Arbeitsblätter ... 434
PDFCreate ... 671
PhoneDialUp ... 759
PhoneDisableListen ... 760
PhoneEnableListen ... 761
PhoneHangUp ... 761
PhoneStatus ... 762
Pi ... 602
Play ... 686
Port-Belegung ... 482
Positions-Animation ... 253
PostKey ... 633
Pow ... 565
Print ... 672
PrintSetup ... 676
PrintWindow ... 677
ProductVersion ... 703
Produktmerkmale ... 26
Produkttyp ... 101
Produktübersicht ... 25
Produktversionen ... 49
Projekt überprüfen ... 64
Projekt-Tags in einem Dokument oder Bildobjekt ersetzen ... 179
Projektausführung
Windows-Systemdienste ... 105
Projekteinstellungen
E-Mail ... 102
FTP-Einstellungen ... 104
Registerkarte Informationen ... 67
Registerkarte Kommunikation ... 81
Registerkarte Optionen ... 68
Registerkarte Persönliche Einstellungen ... 83
Registerkarte Viewer ... 75
Registerkarte Web ... 86
Zusätzlich Konfigurieren ... 67
Projektexplorer ... 59
Registerkarte Global ... 60
Registerkarte Grafik ... 60, 170
Registerkarte Komm ... 63
Registerkarte Tasks ... 61
Projektsicherheit ... 407
Projektsprache

Index

Einstellen bei Programmstart..... 439
Einstellen während der Laufzeit..... 439
Protokollmeldungsfunktionen..... 557

Q
Querverweisliste
Winstudio..... 96

R
Rand..... 572
Rasterobjekt..... 345
Dialog Daten..... 349
Dialog Erweiterte Einstellungen..... 351
Dialog Spalten..... 347
RDFileN..... 673
Recipe..... 634
Reelle Werte in Integer-Tags..... 832
Registerkarte Format..... 257
Registerkarte Global..... 60
Registerkarte Grafik..... 60, 170
Registerkarte Informationen..... 67
Registerkarte Komm..... 63
Registerkarte Kommunikation..... 81
Registerkarte Optionen..... 68
Alarm-Historie..... 69, 281
Datenbankkonfiguration..... 70, 502
Ereignisse..... 287
Konfigurieren einer Standarddatenbank für alle Task-Historien..... 70, 508
Registerkarte Persönliche Einstellungen..... 83
Registerkarte Tasks..... 61
Registerkarte Viewer..... 75
Registerkarte Web..... 86
RegSaveCE..... 704
Relationale SQL-Datenbanken..... 494
ReloadAppFromServer..... 704
Remote LogWin..... 448
Remote-Datenbankspion..... 447
Remote-Verwaltung..... 66
RemoveUser..... 619
Report..... 635
Report-Arbeitsblatt auf Seite 363
ResetBit..... 565
ResetDecimalPointsTable..... 678
Rezeptur-Arbeitsblatt..... 361
RGBColor..... 679
RGBComponent..... 679
Round..... 566
RunGlobalProcedureAsync..... 636
RunGlobalProcedureAsyncGetStatus..... 637
RunGlobalProcedureOnFalse..... 639
RunGlobalProcedureOnTrigger..... 641
RunGlobalProcedureOnTrue..... 642
RunVBScript..... 643

S
SaveAlarmFile..... 705
SaveScreenShot..... 680
Scheduler-Arbeitsblatt..... 370
Schleifen mit Code..... 836
Schleifenfunktionen..... 718
Schlüsselwörter..... 818
SecureViewerReload..... 644
SendEmail..... 740
SendEmailExt..... 741
SendEvent..... 766
SendKeyObject..... 644
SetAppAlarmPath..... 706
SetAppHSTPath..... 706
SetAppPath..... 646
SetBit..... 567
SetDateFormat..... 707
SetDecimalPoints..... 681
SetDisplayUnit..... 682
SetKeyboardLanguage..... 707
SetLanguage..... 685
SetPassword..... 619
SetRegValue..... 708
SetSystemDate..... 599
SetSystemTime..... 599
SetTagDisplayUnit..... 683
SetTagValue..... 717
SetTranslationFile..... 685
SetViewerInFocus..... 647
SetViewerPos..... 647
SetWebConfig..... 709
ShowInplaceInput..... 607
ShowMessageBox..... 608
ShutDown..... 648
Sicherheit..... 407
Sicherheit für webbasierte Anwendungen
Implementieren..... 477
Sicherheitsberechtigungen
Speichern..... 417
Sicherheitsfunktionen..... 610
Sicherheitshinweise..... 20, 21
Sicherheitsmodi..... 407
Servereinstellungen..... 412
Sicherheitssystem
Konfigurationsassistent..... 410
Sicherheitssystemkonfiguration
Datensicherung..... 431
Sicherheitszugriffsebenen..... 408
Signalgrafik..... 21
Signalwörter..... 21
Sin..... 602
Skript-Arbeitsblatt..... 368
Startup..... 369
SMA..... 467
SNMPGet..... 711
SNMPSet..... 712
Speichern von eigenen Projektsymbolen..... 220
Sperren / Entsperren eines Benutzers..... 434
Sprachverweis..... 808
Anweisungen..... 814

Eigenschaften ... 813
Fehler ... 818
Funktionen ... 817
Konstanten ... 809
Methoden ... 817
Objekte und Kollektionen ... 813
Operatoren ... 808
Schlüsselwörter ... 818
System-Tags ... 821
SQL Server auf einem Windows-Embedded-Gerät ... 524
SQL Server CE
Verwendung ... 523
Sqrt ... 568
Stapelerstellung Historie ... 305
Starten der Software ... 44
StartTask ... 648
Statistische Funktionen ... 569
Statuszeile
Winstudio ... 96
Stopp ... 66
Str ... 583
Str2Asc ... 583
StrCompare ... 584
StrCompareNoCase ... 585
StrFromInt ... 585
StrFromReal ... 586
StrFromTime ... 587
StrGetElement ... 588
String-Funktionen ... 578
StrLeft ... 588
StrLen ... 589
StrLower ... 589
StrRChr ... 589
StrRight ... 590
StrSetElement ... 590
StrStr ... 591
StrStrPos ... 591
StrTrim ... 592
StrTrimAll ... 593
StrUpper ... 593
Studio-Datenbank-Gateway ... 496
Support
siehe Service-Hotline ... 843
Swap16 ... 568
Swap32 ... 568
Sybase
Verwendung ... 525
Symbole ... 21, 219
Symbolleiste
WinStudio-Ausführungsfunktionen ... 54
WinStudio-Ausrichtungsfunktionen ... 55
WinStudio-Debug-Funktionen ... 57
WinStudio-Dynamikfunktionen ... 58
WinStudio-Objekteigenschaften ... 56
WinStudio-Tabellenfunktionen ... 57
WinStudio-Tag-Eigenschaften ... 56
WinStudio, Anpassen ... 53
SyncAlarm ... 791
SyncAlarmStatus ... 792
SyncEvent ... 793
SyncEventStatus ... 793
synchron
gegenüber asynchron ... 651
gegenüber synchron ... 651
SyncTrendStatus ... 795
System-Tags ... 821
Systemanforderungen ... 39
Systeminformationsfunktionen ... 687

T

Tag-Datenbank zurücksetzen ... 152
Tag-Datenbank-Funktionen ... 714
Tag-Datentyp
Auswahl ... 124
Tag-Eigenschaften
Liste der ... 142
Tag-Eigenschaften und -Parameter ... 136
Tag-Eigenschaften: Alarme
Verwenden ... 138
Tag-Eigenschaften: Historie
Verwenden ... 141
Tag-Erstellung
"On the Fly" ... 134
Datenblatt ... 132
Tag-Gültigkeitsbereich
Auswahl ... 132
Tag-Typ
Auswahl ... 122
Tags gemeinsam verwenden
IndraLogic-Geräte ... 163
Tags in Ihrem Projekt
Verwenden von ... 147
Tags und die Projektdatenbank ... 115
Tags-Erstellung
Modifikation ... 134
Tan ... 602
Tastatur ... 80
virtuell ... 80
Thin Client
Konfigurieren ... 473
Thin Clients ... 449
Thin-Client-Programm ... 451
Tipps & Tricks ... 823
Titelbereich Alarmarbeitsblatt ... 275
Toggle ... 577
Trace ... 557
Treiber-Fehlercodes ... 841
Trendanzeige ... 309
Trendarbeitsblatt ... 301
Trenddatenbank ... 308
Trendobjekt
Datenquellen ... 326
Objekteigenschaften ... 313
Optionen ... 317

Index

Punkte ... 314
Stiftstil ... 315
Verwenden der Datenquellen-Datenbank ... 339
Verwenden der Textdatei Datenquelle ... 335
Trendsteuerung
Achsen ... 320
Legende ... 330
SPC ... 319
Symbolleiste ... 325
Weitere Einstellungen ... 332
Trigonometrische Funktionen ... 599
True ... 577
Trunc ... 569

U
Übersetzungsfunktionen ... 684
Übersetzungstabelle
Hinzufügen einer Sprache ... 437
Übung
Betrachten Ihres Projekts im Web ... 483
UnblockUser ... 620
Unterstützung durch Windows Embedded ... 839

V
ValueToChar ... 594
ValueWToChar ... 594
Variablen
Deklarieren ... 830
VBScript ... 797
Aufruf einer externen Datei ... 652
VBScript Editor IntelliSense ... 823
VBScript-Operatoren
Priorität von ... 832
VBScript-Prozeduren ... 828
VBScript-Schnittstellen in der Software ... 797
Verknüpfen der Datenbank über einen Remote-Datenbankanbieter ... 495
Vertikal spiegeln ... 262
Verwenden des LogWin-Moduls ... 446
ViewerPostMessage ... 649
Virtuelle Tastatur ... 80

W
Wait ... 650
Web Tunneling Gateway ... 467
Installieren ... 470
Webanwendung
Konfigurieren des Seitenservers ... 467
WebGetFile ... 673
Webserver
Apache für Windows ... 470
CEWebServer ... 468
Installieren und Konfigurieren ... 467
Microsoft IIS ... 468
NTWebServer ... 468
Webserver als Host Ihrer Projektseiten
Konfigurieren ... 467
Weitere Einstellungen für Alarmarbeitsblatt ... 279
Werkzeug Schriftarten ... 269
Werkzeuge
Winstudio ... 90
Windows
Prozesse ... 739
Windows-Befehl
Ausführung ... 651
WinExec ... 651
WinExecIsRunning ... 653
WinStudio-Bibliotheken ... 219
WinStudio-Symbolleiste
Ausrichtungsfunktionen ... 257
WritePrivateProfileString ... 714

X
XGet ... 763
XRun ... 764
XSet ... 765

Z
Zeiger ... 131
Zielplattformen und Produkttypen ... 100
Zielsystem
Konfigurieren ... 487
Zu dieser Dokumentation ... 19
Gültigkeit der Dokumentation ... 19
Zustandscodes für externe Datenbankoperationen ... 375

Notizen

Rexroth
Bosch Group

Bosch Rexroth AG
Electric Drives and Controls
Postfach 13 57
97803 Lohr, Deutschland
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr, Deutschland
Tel. +49 9352 18 0
Fax +49 9352 18 8400
www.boschrexroth.com/electrics

R911336881

DOK-IWORKS-WINSTUD*V13-AP01-DE-P