



Crimson[®] 3.1 Benutzerhandbuch

LP1044-G | August 2019



COPYRIGHT

©2017-2019 Red Lion Controls, Inc. Alle Rechte vorbehalten. Red Lion, das Red Lion-Logo und Crimson sind eingetragene Marken von Red Lion Controls, Inc. Alle anderen Firmen- und Produktnamen sind Marken der jeweiligen Eigentümer.

Red Lion Controls, Inc.
20 Willow Springs Circle
York, PA 17406

KONTAKTINFORMATIONEN:

In den USA: +1 (877) 432-9908
Außerhalb den USA: +1 (717) 767-6511

Website: www.redlion.net

Unterstützung: support.redlion.net

Inhaltsverzeichnis

Vorwort	xxi
Haftungsausschluss	xxi
Zweck	xxi
Zielgruppe	xxi
Markenhinweise	xxi
Dokumentverlauf und zugehörige Veröffentlichungen	xxii
Zugehörige Dokumente	xxii
Zusätzliche Produktinformationen	xxii
Erste Schritte	1
Unterstützte Geräte	1
Systemanforderungen	1
Installation der Software	1
Registrierung	2
Überprüfen auf Updates	2
Installation der USB-Treiber	2
Fehlerbehebung	3
Hilfe und Unterstützung	4
Die nächsten Schritte	4
Umstieg von Crimson 3.0	5
Importieren von Datenbanken	5
Symbolauswahl	7
Modellkonvertierung	7
Konvertieren von Graphite aus	7
Konvertieren von G3 aus	8
Konvertieren von G3 Kadet aus	8
Konvertierung von Data Station Plus aus	8
Konvertieren von anderen Geräten aus	8
Benutzerdefinierte Webseiten	9
Neue Funktionen	9
Zero Config	9
OPC UA-Server	9
Neue Primitive	9
Symbolbibliothek	11
Webserver	11
IEC-61131	13

Crimson-Grundlagen	15
Fensterlayout	15
Das Navigationsfenster	15
Das Ressourcenfenster	15
Das Bearbeitungsfenster	15
Zuklappen von Fenstern	16
Die Kategorien	16
Kommunikationen	16
Daten-Tags	16
Anzeigeseiten	16
Programme	16
Webserver	17
Datenlogger	17
Sicherheit	17
SQL-Abfragen	17
Steuerung	17
E/A-Module	18
Navigation	18
Zurück und Vorwärts	18
Tastaturkürzel für Kategorien	18
Tastaturkürzel für Elemente	18
Navigationslisten	19
Arbeiten mit Ordnern	19
Sortieren von Listen und Ordnern	19
Drag-and-Drop-Vorgänge	20
Suchen in Listen	20
Private Elemente	20
Rückgängig machen und Wiederholen	20
Globale Suche	21
Arbeiten mit Datenbanken	21
Importieren aus Crimson 2.0	21
Importieren aus Crimson 3.0	21
Datenbankkennungen	22
Speichern eines Abbilds	23
Datenbankschutz	23
Konvertieren einer Datenbank	24
Suchen nach Datenbankfehlern	24
Download auf ein Gerät	25
Konfigurieren der Verbindung	25
Senden der Datenbank	26
Extrahieren von Datenbanken	26

Senden von Datum und Uhrzeit	27
Die Speicherkarte	28
Anmelden der Speicherkarte	28
Formatieren der Karte	28
Fernüberwachung	29
Systemmenü	29
Verwenden von Kommunikationen	31
Netzwerkconfiguration	31
Zero Config	31
Ethernet-Einstellungen	32
Anschlusseinstellungen	32
DNS-Einstellungen	33
Bitübertragungsschicht	33
Maximale Segmentgröße	33
Mehrere Ports	33
TLS-SSL-Einstellungen	33
Einstellungen	34
Vertrauenswürdige Roots	34
Routing-Einstellungen	34
Routing-Modus	35
Routingtafel	35
Download-Einstellungen	35
Remote Update	35
Einheitenadressierung	36
Hinzufügen von Ports	36
Protokollauswahl	36
Verwenden virtueller Ports	37
Auswahl eines seriellen Ports	37
Auswählen eines Protokolls	37
Protokolloptionen	38
Arbeiten mit Geräten	38
Erweiterte Einstellungen	40
Erstellen von Tags	40
Port- und Gerätenutzung	40
Verwenden von Erweiterungsmodulen	41
Slave-Protokolle	42
Auswählen des Protokolls	42
Hinzufügen von Gateway-Blöcken	43
Hinzufügen von Elementen zu einem Block	43
Zugriff auf einzelne Bits	44
Protokollkonvertierung	45
Master und Slave	45

Master und Master	45
Die richtige Vorgehensweise	45
Steuern von Master-Blöcken	46
Datentransformation	47
Deaktivieren der Kommunikation	47
Arbeiten mit Tags	49
Alles über Tags	49
Datenquellen	49
Typen von Tags	49
Tag-Tasten	50
Tag-Attribute	50
Vorteile von Tags	52
Bearbeiten von Eigenschaften	53
Ausdruckseigenschaften	53
Auswählen eines Tags	53
Erstellen von Tags	54
Kommunikationsreferenzen	54
Bearbeiten eines Ausdrucks	54
Komplexe Ausdrücke	55
Übersetzbare Zeichenfolgen	55
Zwei-Wege-Eigenschaften	56
Aktionseigenschaften	57
Farbeigenschaften	57
Protokolleigenschaften	59
Erstellen von Tags	59
Duplizieren von Tags	59
Bearbeiten mehrerer Tags	60
Verwenden von Kopieren von	60
Verwenden von Einfügen spezial	60
Eigenschaftsauswahl	60
Importieren und Exportieren	61
Suchen nach der Tag-Nutzung	61
Numerische Tags	62
Dateneigenschaften	62
Datenquelle	62
Datenskalierung	64
Datensimulation	65
Datenaktivitäten	65
Daten Sollwert	65
Formateigenschaften	65
Datenbezeichnungen	65
Formattyp	66

Datenlimits	66
Farbeigenschaften	66
Farbtyp	66
Alarmerigenschaften	67
Für jeden Alarm	67
Auslösereigenschaften	69
Für jeden Auslöser	69
Diagrammeigenschaften	69
Sicherheitseigenschaften	70
Markierungs-Tags	71
Dateneigenschaften	71
Datenquelle	71
Datensimulation	73
Datenaktivitäten	73
Daten Sollwert	74
Formateigenschaften	74
Datenbezeichnungen	74
Formattyp	74
Farbeigenschaften	75
Farbtyp	75
Alarmerigenschaften	75
Für jeden Alarm	75
Auslösereigenschaften	77
Für jeden Auslöser	77
Sicherheitseigenschaften	77
String-Tags	78
Dateneigenschaften	78
Datenquelle	78
Datensimulation	80
Datenaktivitäten	80
Formateigenschaften	81
Datenbezeichnungen	81
Formattyp	81
Farbeigenschaften	82
Farbtyp	82
Sicherheitseigenschaften	82
Basis-Tags	83
Datenwert	83
Datensimulation	83
Datenbezeichnungen	83
Weiterführende Themen	84
Array-Eigenschaften	84
Tag-Datenfluss	84

Lesevorgang bei numerischen Tags	84
Schreibvorgang bei numerischen Tags	85
Verwenden von Bei Schreiben	85
Verwenden von Formaten	87
Formattypen	87
Format Allgemein	88
Format Verbunden	88
Format Numerisch	89
Festgelegtes Datenformat	89
Dynamisches Datenformat	89
Formateinheiten	90
Format Wissenschaftlich	91
Festgelegtes Datenformat	91
Dynamisches Datenformat	91
Formateinheiten	91
Format Uhrzeit und Datum	92
Formatmodus	92
Zeitformat	92
Datumsformat	93
Format IP-Adresse	93
Format Zweifachstatus	93
Format Mehrfachstatus	94
Formatsteuerung	94
Formatstatus	94
Formatbefehle	94
Format Zeichenfolge	95
Verwenden von Färbungen	97
Arten von Färbungen	97
Färbung Allgemein	97
Färbung Verbunden	98
Färbung Festgelegt	98
Färbung Zweifachstatus	98
Färbung Mehrfachstatus	99
Formatsteuerung	99
Formatstatus	99
Farbe Befehle	99
Erstellen von Anzeigeseiten	101
Editor-Grundlagen	101
Arbeiten mit Seiten	101

Ändern der Vergrößerungseinstellung	101
Das Ressourcenfenster	102
Primitive	102
Symbolbibliothek	103
Datentags	103
Hinzufügen von Elementen zu einer Seite	104
Verwendung des Notizblocks	104
Arbeiten mit Primitiven	106
Auswählen von Primitiven	106
Verborgene Primitive	106
Verwenden der Schnellleiste	106
Verschieben von Primitiven zwischen Seiten	106
Verschieben von Primitiven zwischen Datenbanken	107
Ändern der Größe von Primitiven	107
Drehen und Spiegeln von Primitiven	107
Verwenden von Layoutgriffen	107
Intelligente Ausrichtung	107
Schnellausrichtung	108
Verwenden des Gitters	108
Ausrichten von Primitiven	109
Gleichmäßige Verteilung von Primitiven	109
Ändern der Reihenfolge von Primitiven	109
Duplizieren von Primitiven	110
Bearbeiten mehrerer Primitive	110
Verwenden von Kopieren von	110
Verwenden von Einfügen spezial	110
Eigenschaftsauswahl	111
Springen zu anderen Elementen	111
Eigenschaften von Primitiven	112
Ein- oder Ausblenden von Primitiven	112
Definieren von Farben für Primitive	112
Definieren von blinkenden Farben	114
Definieren von Zweifachstatus-Farben	114
Definieren von Vierfachstatus-Farben	115
Definieren von gemischten Farben	115
Definieren von Behälterfüllungen	115
Definieren von Füllformaten	116
Ältere Füllformate	117
Definieren von Randformaten	117
Definieren des Randbeschnitts	118
Ältere Randformate	119
Zuweisen neuer Farben zu Symbolen	120

Verwenden von Gruppen	121
Erstellen und Auflösen von Gruppen	121
Bearbeiten von Gruppen	121
Bearbeiten verschachtelter Gruppen	122
Erweitern von Gruppen	122
Hinzufügen von Bewegungen zu Primitiven	123
Hinzufügen von Text zu Primitiven	124
Texteigenschaften	124
Weitere Eigenschaften	125
Hinzufügen von Daten zu Primitiven	126
Dateneigenschaften	126
Weitere Eigenschaften	127
Eingabeeigenschaften	128
Formateigenschaften	129
Farbeeigenschaften	129
Hinzufügen von Aktionen zu Primitiven	131
Schützen von Aktionen	131
Aktivieren von Aktionen	131
Die Aktion Gehe zu Seite	132
Die benutzerdefinierte Aktion	132
Die Aktion Drucktaste	133
Die Aktion Wert ändern	134
Die Aktion Wert steigern	134
Die Aktion Melodie abspielen	135
Die Aktion Benutzer anmelden	135
Die Aktion Benutzer abmelden	135
Die Aktion Popup ausblenden	135
Die Aktion Alle Popups ausblenden	135
Hinzufügen von Aktionen zu Symbolen	136
Bearbeiten von Seiteneigenschaften	137
Allgemeine Eigenschaften	137
Weitere Eigenschaften	138
Aktionseigenschaften	139
Sicherheitseigenschaften	139
Einstellungen der Benutzeroberfläche	140
Globale Eigenschaften	140
Globale Aktivitäten	140
Globale Timeouts	140
Popup-Position	141
Diagnostik	141
Sprachen	141

Eingabeeigenschaften	141
Ziffernblockoptionen	142
Dateneingabemodus	142
Sonstige	142
Eigenschaften von Bildern	143
Bilder	143
Wartung	143
Schrifteigenschaften	144
Wartung	144
Symboleigenschaften	144
Verwalten von Bildern	146
Verwalten von Schriftarten	148
Weiterführende Themen	150
Definieren von Farbausdrücken	150
Erstellen von Farben	150
Aufteilen von Farben	150
Auswählen von Farben	150
Mischen von Farben	150
Reagieren auf Berührungen	150
Primitiv-Arten	151
Kernprimitive	151
Geometrische Primitive	151
Grundeigenschaften	151
Weitere Eigenschaften	151
Das Linien-Primitiv	152
Linieigenschaften	152
Kanteneigenschaften	152
Text- und Daten-Primitive	152
Das Bild-Primitiv	153
Definieren von Bildern	154
Anpassen von Bildern	154
Das abgeschrägte Primitiv	154
Figureigenschaften	155
Linieigenschaften	155
Die Schaltflächen-Primitive	155
Messgerät-Primitive	157
Messgerät-Konzepte	157
Typen	157
Fehler	157
Bänder	158
Stile	158
Nackte Messgeräte	159
Eigenschaften von Messgeräten	159
Formateigenschaften	159

Layouteigenschaften für radiale Messgeräte	160
Layouteigenschaften für lineare Messgeräte	161
Stileigenschaften	161
Bandeigenschaften	162
Eigenschaften von Einstellmarken	162
Balken- und Liniendiagramme	163
Die Balkendiagramm-Primitive	163
Optionseigenschaften	163
Figureigenschaften	163
Eigenschaften von Streudiagrammen	164
Linieigenschaften	164
Optionseigenschaften	164
Figureigenschaften	165
Aktionsschaltflächen	166
Beleuchtete Schaltflächen	167
Indikatoren	169
Umschalter mit Zweifachstatus	170
Schaltereigenschaften	170
Erweiterte Eigenschaften	171
Umschalter mit Dreifachstatus	172
Schaltereigenschaften	172
Erweiterte Eigenschaften	173
Selektoren mit Zweifachstatus	174
Selektoren mit Dreifachstatus	174
System-Primitive	175
Betrachterformat	175
Der Alarmbetrachter	175
Optionseigenschaften	176
Aktionseigenschaften	177
Zeiteigenschaften	177
Farbeeigenschaften	177
Der Alarm-Ticker	177
Optionseigenschaften	178
Der Ereignisbetrachter	178
Optionseigenschaften	179
Aktivierungseigenschaften	179
Zeiteigenschaften	179
Dateibetrachter	179
Optionseigenschaften	180
Benutzer-Manager	180
Trendbetrachter	181
Optionseigenschaften	181
Formateigenschaften	182

Schaltflächeneigenschaften	182
Zeiteigenschaften	182
Stifeigenschaften	183
Füllungseigenschaften	183
Berührungskalibrierung	184
Berührungstester	184
PDF-Viewer	184
Optionseigenschaften	184
Kamera	185
Optionseigenschaften	185
Figureigenschaften	185
Ältere Primitive	186
Skala-Primitiv	186
Dateneigenschaften	186
Figureigenschaften	187
Grenzwerteigenschaften	187
Formateigenschaften	188
Lokalisierung	189
Auswählen von Sprachen	189
Konfigurieren der automatischen Übersetzung	191
Übersetzen der Datenbank	192
Eingeben von Übersetzungen	192
Konfigurieren der automatischen Übersetzung	192
Exportieren und Importieren	192
Anwenden eines Lexikons	193
Vorschau von Übersetzungen	193
Wechseln zwischen Sprachen	193
Verwenden von Widgets	195
Erstellen eines Widgets	195
Zusammenfassung	199
Warum dies wichtig ist	199
Einzelheiten	199
Datendefinitionen für Widgets	200
Archivieren von Widgets	202
Ordnerbindung	203
Erweiterte Bindung	205
Klassenübereinstimmung	205
Bindungsvorzeichen	205
Verwenden von Binden an	205
Verwenden von Punkten	205
Verwenden von Caret-Zeichen	205

Spezielle Namen	206
Details-Widgets	207
Aktivieren der Detailerstellung	207
Definieren von Datenelementen	207
Ergebnisse der Bindung	208
Mehrere Detailseiten	208
Verwenden des Datenloggers	209
Erstellen von Datenprotokollen	209
Einrichtungseigenschaften	209
Inhaltseigenschaften	210
Monitoreigenschaften	210
Batch-Protokollierung	211
Steuern eines Batch	211
Digitale Signaturen	212
Speicherung von Protokolldateien	213
Der Protokollierungsvorgang	214
Zugriff auf Protokolldateien	215
Verwenden des Webservers	217
Wechseln zwischen Versionen	217
Webserver-Eigenschaften	218
Steuerungseigenschaften	218
Einstellung der Eigenschaften	220
Sicherheitseigenschaften	222
Erweiterte Eigenschaften	223
Hinzufügen von Webseiten	224
Arbeiten mit Zertifikaten	225
Einführung	225
Verwenden von Zertifikaten	226
Beziehen eines kommerziellen Zertifikats	226
Nutzen einer lokalen Zertifizierungsstelle	226
Verwenden eines selbst signierten Zertifikats	226
Verwenden des Standardzertifikats	227
Jenseits des Subnetzes	227
Untätig bleiben	228
Verwenden einer benutzerdefinierten Webseite	229
Erstellen benutzerdefinierter Webseiten	231
Namenskonventionen	231
Ressourcen	232
Serverbefehle	233
Servere Enthalten	233

Steuern der Zwischenspeicherung	233
Eingebettete Daten	233
Tagdaten	233
Globale Daten	234
Datenindex-Daten	234
Datenansichts-Daten	234
AJAX-Updates	235
Lesen von Tags	235
Schreiben von Tags	235
Bereitstellung von Sites	235
Verwenden des Sicherheitssystems	237
Grundsätzliches zum Thema Sicherheit	237
Objektbasierte Sicherheit	237
Benannte Benutzer	237
Benutzerrechte	238
Zugangssteuerung	238
Protokollierung von Schreibvorgängen	238
Standardzugriff	239
Anmeldung auf Anforderung	239
Wartungszugriff	239
Vor Verwendung prüfen	239
Sicherheitseinstellungen	240
Erstellen von Benutzern	241
Festlegen der Tag-Sicherheit	241
Festlegen der Seitensicherheit	242
Sicherheitsbezogene Funktionen	242
Verwenden von SQL-Abfragen	243
Konfigurieren des Servers	243
Erstellen von Abfragen	244
Filtern der Ergebnisse	245
Prüfen des SQL-Codes	245
Erstellen von Spalten	246
Zugeordnete Tags	247
Verwenden von IEC-61131	249
Was bietet die Kategorie Steuerung?	249
Erlernen von IEC-61131	249
Arbeiten mit Programmen	250
Programmtypen	250
Erstellen von Programmen	250
Bearbeiten von Programmen	251

Der Editor für Strukturierter Text	251
Der Editor für die Funktionsbausteinsprache	251
Der Kontaktplan-Editor	252
Der Editor für Anweisungsliste	253
Konvertieren von Programmen	253
Programmeigenschaften	253
Verwenden von Variablen	255
Erstellen von Variablen	255
Eigenschaften von Variablen	255
Parameter	256
Projekteigenschaften	257
Neuerstellen von Projekten	258
IEC-61131-Debugging	259
Verwenden von Anschlüssen	261
Konfigurieren von Anschlüssen	261
Allgemeine Einstellungen	261
Netzwerkoptionen	262
Gerätedaten-Optionen	263
Optionen von Tag-Daten	264
Modus Ausgelöst	266
Anschlussdiagnose	267
Datenpufferung	268
Datenlayout in JSON	269
Anschlussoptionen	269
Verbindungsstatus	269
Gerätedaten	270
Tagdaten	270
Site-Benennung	273
Der generische MQTT-Anschluss	274
Anschlusseinstellungen	274
Anschlussbetrieb	275
Der Amazon MQTT-Anschluss	276
Anschlusseinstellungen	276
Anschlussbetrieb	277
Der Azure MQTT-Anschluss	278
Anschlusseinstellungen	278
Anschlussbetrieb	279
Der Google MQTT-Konnektor	281
Anschlusseinstellungen	281
Bedienung des Steckverbinders	282
Der Sparkplug MQTT-Konnektor	283

Anschlusseinstellungen	283
Anschlussbetrieb	284
Verwenden von Diensten	285
Verwenden des OPC UA-Servers	285
Konfigurieren des Dienstes	285
Diensteigenschaften	285
Dateneigenschaften	286
Darstellung von Daten	287
Historische Daten	287
Verwenden von Zeitmanagement	288
Konfigurieren des Dienstes	288
Zeitserver	288
Time Client	288
Zeitstempel	289
Auswählen eines SNTP-Servers	290
Konfigurieren von Zeitzonen	290
Verwenden des FTP-Servers	291
Konfigurieren des Dienstes	291
FTP-Sicherheit	291
Verwenden von OPCWorx Proxy	292
Konfigurieren des Dienstes	292
Verwenden der Dateisynchronisierung	293
Konfigurieren des Dienstes	293
FTP-Client	293
Synchronisierung von Protokolldateien	294
Verwenden von E-Mail	295
Hinzufügen von Kontakten	295
Konfigurieren von SMTP	296
Konfigurieren von SMS	297
Verwenden der SQL-Synchronisierung	299
Konfigurieren des Dienstes	299
Freigeben von Ports	301
Aktivieren von TCP/IP	301
Freigeben des erforderlichen Ports	301
Herstellen einer Verbindung über einen anderen Port	302
Herstellen einer Verbindung über Ethernet	303
Rein virtuelle Ports	304
Einschränkungen	304
Verwenden von Modems	305
Hinzufügen einer Dial-In-Verbindung	305

Hinzufügen einer Dial-Out-Verbindung	308
Hinzufügen einer SMS-Verbindung	309
Verarbeitung von SMS-Nachrichten	309
Erweiterungskarten für Modems	310
Überprüfen des Modemstatus	311
Verwenden von mehreren Schnittstellen	313
Schnittstellenauswahl	313
Standardroute	313
Verwenden des USB-Hosts	315
Unterstützung für Speicher-Sticks	315
Allgemeine Eigenschaften	315
Übertragungseigenschaften	316
Tastaturunterstützung	316
Mausunterstützung	316
Verwenden von Programmen	317
Die Programmliste	317
Ermitteln der Programmnutzung	317
Bearbeiten von Programmen	318
Anzeigen von Hilfetext	318
Das Ressourcenfenster	319
Programmdatentypen	320
Globale Einstellungen	321
Programmeigenschaften	322
Hinzufügen von Kommentaren	324
Zurückgeben von Werten	324
Vermeiden von potenziellen Problemen	325
Übergeben von Argumenten	325
Tipps für das Programmieren	326
Mehrere Aktionen	326
If-Anweisungen	326
Switch-Anweisungen	327
Lokale Variablen	328
Schleifenkonstrukte	328
Die While-Schleife	328
Die For-Schleife	329
Die Do-Schleife	330
Schleifensteuerung	330
Schreiben von Ausdrücken	331
Datenwerte	331
Konstanten	331

Integerkonstanten	331
Zeichenkonstanten	332
Logische Konstanten	332
Gleitkommakonstanten	332
String-Konstanten	332
Tag-Werte	332
Tag-Eigenschaften	333
Seiteneigenschaften	333
Kommunikationsverweise	333
Einfache Berechnungen	334
Operator-Priorität	334
Typenkonvertierung	335
Vergleichen von Werten	336
Testen von Bits	336
Mehrere Bedingungen	337
Auswählen von Werten	337
Manipulation von Bits	338
AND, OR und XOR	338
Shift-Operatoren	338
Bitweises NOT	338
Indizierung von Arrays	338
Indizierung von Strings	339
Hinzufügen von Strings	339
Aufrufen von Programmen	339
Verwenden von Funktionen	339
Priorität Zusammenfassung	339
Schreiben von Aktionen	341
Ändern einer Seite	341
Ändern numerischer Werte	342
Einfache Zuweisung	342
Zusammengesetzte Zuweisung	342
Schrittweise erhöhen und verringern	342
Ändern von Bitwerten	343
Ausführen von Programmen	343
Verwenden von Funktionen	343
Erweitertes Debugging	345
Auf eine Konsole zugreifen	345
Anzeigen der Ausgabe	345
Debugging-Programme	346
Ausführen von Befehlen	346

Tag-Befehle	347
Programmbefehle	347
Graphite E/A-Module	349
Auswählen von Modulen	349
Die GMPID PID-Module	350
Allgemeine Eigenschaften	350
Vorgang	350
Einheiten	350
Initialisierung	351
Smart OnOff	351
Beispiel für Steigung und „Offset“	352
Steuerungseigenschaften	353
Sollwert	353
Einstellungen für Auto-Abstimmung	354
Benutzer-PID-Einstellungen	354
Leistungseigenschaften	355
Kraftübertragung	355
Transferkurve	356
Alarmerigenschaften	356
Diagramm zum Alarmverhalten	358
Heizstrom	358
Eingangsfehler	359
Ausgangseigenschaften	359
Digitalausgänge	359
Linearer Ausgang (nur GMPID1)	360
Auto-Abstimmung	360
Aufrufen der Auto-Abstimmung	361
Verfügbare Daten	361
Schleifenstatus	362
Schleife/Steuerung	363
Schleife/Leistung	364
Schleife/Alarmer	364
Schleife/PID	365
Schleife/Profil	365
Ausgänge/Zykluszeiten	365
Ausgänge/Externe Daten	366
Ausgänge/Informationen	366
Das GMSG-Eingangsmodul für Dehnungsmessstreifen	366
Allgemeine Eigenschaften	366
Eingänge	367
Vorgang	367
Initialisierung	367
Smart OnOff	367
Steuerungseigenschaften	368

Leistungseigenschaften	368
Alarmerigenschaften	369
Ausgangseigenschaften	369
Auto-Abstimmung	369
Verfügbare Daten	370
Schleifen/Status	370
Schleife/Steuerung	370
Schleife/Alarmer	370
Schleife/Leistung	370
Schleife/ScaleInput1	370
Schleife/ScaleInput2	370
Schleife/PeakValley	371
Schleife/Tara	371
Ausgänge/Zykluszeiten	372
Ausgänge/Externe Daten	372
Ausgänge/Informationen	372
Die Temperaturmodule GMTC und GMRTD	372
Konfigurationseigenschaften	372
Allgemein	373
Eingänge	373
Millivolt – Eigenschaften (nur bei GMTC)	373
Allgemein	373
Millivolt Scaling (Millivolt-Skalierung)	374
Verfügbare Daten	374
Eingang/Status	374
Eingang/Steuerung	374
Eingang/ScaleMvInputs	374
Die Analogeingangsmodule GMINI und GMINV	375
Konfigurationseigenschaften	375
Allgemein	375
Eingänge	375
Verfügbare Daten	376
Eingang/Status	376
Eingang/Steuerung	376
Das Analogausgangsmodul GMOUT	377
Konfigurationseigenschaften	377
Ausgang 1 bis Ausgang 4	377
Anfänglicher Ausgang – Eigenschaften	378
Initialisierung	378
Ausgangsaktion	378
Verfügbare Daten	379
Daten	379
Skalierung	379
Alarmer	379

Das digitale E/A-Modul GMDIO	380
Konfigurationseigenschaften	380
Verfügbare Daten	380
Variablen/Eingänge	380
Variablen/Ausgänge	380
Das universelle Eingangsmodul GMUIN	381
Eingangseigenschaften	381
Vorgang	381
Einheiten	381
Verfügbare Daten	382
Eingang/Status	382
Eingang/Steuerung	382

Vorwort

Haftungsausschluss

Dieses Benutzerhandbuch bietet Anleitung zur Verwendung von Crimson® 3.1, um leistungsstarke und attraktive Crimson-Gerätelösungen für unterstützte Red Lion-Geräte zu entwickeln. Es ist nicht als schrittweise Anleitung oder als vollständige Beschreibung aller Verfahren gedacht, die zum Abschluss aller Operationen notwendig und ausreichend sind.

Obwohl alle Anstrengungen unternommen wurden, um sicherzustellen, dass dieses Dokument zum Zeitpunkt der Veröffentlichung vollständig und exakt ist, sind Änderungen an den darin enthaltenen Informationen vorbehalten. Red Lion Controls, Inc. ist nicht verantwortlich für Ergänzungen oder Änderungen des Originaldokuments. Industrielle Netzwerke variieren stark in der Konfiguration, Topologie und den Verkehrsbedingungen. Dieses Dokument ist nur als allgemeine Anleitung gedacht. Es wurde nicht auf alle möglichen Anwendungen getestet und es kann in manchen Situationen nicht vollständig oder genau sein.

Benutzer dieses Dokuments sind aufgefordert, die Warnungen und Vorsichtsmaßnahmen im gesamten Dokument zu beachten.

Zweck

In diesem Handbuch finden Sie konkrete Informationen dazu, wie Sie die von Crimson 3.1 bereitgestellten Funktionen anwenden und einsetzen, um auf effizientere Weise robuste und attraktive Crimson-Gerätelösungen zu entwickeln.

Zielgruppe

Dieses Handbuch ist für Mitarbeiter gedacht, die für die Konfiguration und Inbetriebnahme von Crimson-Geräten für die Verwendung in Visualisierungs-, Überwachungs- und Steuerungsanwendungen verantwortlich sind.

Markenhinweise

Red Lion Controls, Inc. erkennt das Eigentum der folgenden in diesem Dokument verwendeten geschützten Begriffe an:

- Microsoft®, Windows®, Windows NT® und Windows Vista™ sind eingetragene Marken oder Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

Alle anderen Firmen- und Produktnamen sind Marken der jeweiligen Eigentümer.

Dokumentverlauf und zugehörige Veröffentlichungen

Die Versionen dieses Dokuments auf Papier und elektronischen Medien werden nur bei der Veröffentlichung von Hauptversionen revidiert und enthalten deshalb möglicherweise nicht die neuesten Produktinformationen. Gegebenenfalls werden Dokumentationshinweise oder Produktdatenblätter mit den neuen Informationen oder Dokumentänderungen zwischen Hauptversionen zur Verfügung gestellt.

Die neueste Online-Version dieses Dokuments kann über die Red Lion-Webseite unter <https://www.redlion.net/red-lion-software/crimson/crimson-31> aufgerufen werden.

Zugehörige Dokumente

Verfügbare Dokumente in Bezug auf dieses Produkt finden Sie unter: <https://www.redlion.net/red-lion-software/crimson/crimson-31>.

Zusätzliche Produktinformationen

Zusätzliche Produktinformationen können von Ihrem lokalen Vertriebs Händler oder von Red Lion über die auf der Innenseite der Vorderabdeckung aufgelisteten Telefonnummern und/oder E-Mail-Adressen angefordert werden.

Kapitel 1 Erste Schritte

Willkommen bei Crimson® 3.1, der jüngsten Version der renommierten Gerätekonfigurationssoftware Crimson von Red Lion. Wenn Sie bereits eine frühere Version von Crimson verwendet haben, werden Sie bald feststellen, dass Crimson 3.1 nicht nur sehr viele Verbesserungen aufweist, sondern auch weiterhin die gewohnte Leistungsfähigkeit bietet. Wenn Sie bisher noch nicht mit Crimson gearbeitet haben, sollten Sie wenigstens die ersten paar Kapitel dieses Handbuchs lesen, um einen Überblick über die Funktionsweise der Software zu erhalten. Unabhängig von Ihren Vorkenntnissen werden Sie in kurzer Zeit herausfinden, wie Sie mit Crimson 3.1 schneller und einfacher denn je leistungsstarke und attraktive Crimson-Gerätelösungen entwickeln können.

1.1 Unterstützte Geräte

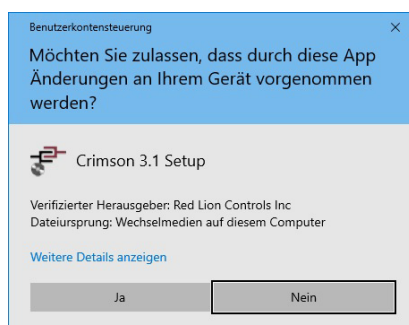
Crimson 3.1 unterstützt nur jene Red Lion-Produkte, die über die Speicherkapazität und Prozessorleistung verfügen, die für die Implementierung der von der Software bereitgestellten zusätzlichen Funktionen erforderlich sind. Dies bedeutet, dass die HMIs und Controller der Graphite-Familie mit Crimson 3.1 konfiguriert werden können, die G3 HMI- und G3 Kadet-Familien jedoch nicht unterstützt werden. Wir gehen davon aus, dass Sie Ihre G3 HMI- und Kadet-Anwendungen zur neuen CR3000- bzw. CR1000-Serie migrieren. Auch die derzeit verfügbaren Versionen von Data Station Plus, Modular Controller und ProductVity Station werden nicht von der Crimson 3.1-Software unterstützt. Daher müssen Sie diese Geräte mit Crimson 3.0 konfigurieren, bis aktualisierte Versionen verfügbar sind.

1.2 Systemanforderungen

Crimson 3.1 ist für die Ausführung unter einer beliebigen Version von Microsoft Windows ab Windows Vista konzipiert. Die Speicheranforderungen sind bescheiden und Crimson 3.1 kann auf allen Systemen ausgeführt werden, die die minimalen Systemanforderungen für das jeweilige Betriebssystem erfüllen. Für die Installation werden etwa 600 MB Festplattenspeicher benötigt, und idealerweise verfügen Sie über ein Display mit ausreichender Auflösung, das die Bearbeitung der Anzeigeseiten ohne übermäßiges Scrollen ermöglicht.

1.3 Installation der Software

Crimson 3.1 wird als ausführbare Datei bzw. .EXE-Datei bereitgestellt. Im Normalfall haben Sie diese Datei von der Red Lion-Webseite heruntergeladen. Falls der Download jedoch von einer anderen Quelle erfolgt ist, überprüfen Sie, ob Windows die digitale Signatur des Pakets akzeptiert. Auf diese Weise können Sie sicherstellen, dass Ihnen auch wirklich Originalsoftware von Red Lion vorliegt:



Wie in der Abbildung oben gezeigt, sollte als Herausgeber „Red Lion Controls, Inc.“ angezeigt werden. Bei Bedarf sollte es Ihnen möglich sein, über die Option „Show Details“ (Details anzeigen) die Integrität der digitalen Signatur weiter zu überprüfen. Wenn Sie mit dem Paket zufrieden sind, klicken Sie auf „Yes“ (Ja), um die Installation zu starten.

Der Installationsprozess ist ein Standardprozess und sollte abgesehen von der Angabe des Zielverzeichnis nur wenige Benutzereingriffe erfordern. Wenn der Vorgang abgeschlossen ist, öffnen Sie das Startmenü und suchen Sie nach dem Ordner „Red Lion“. Klicken Sie auf das Symbol „Crimson 3.1“, um die Software zu starten.

1.4 Registrierung

Wenn Sie Crimson 3.1 zum ersten Mal ausführen, haben Sie die Gelegenheit, die Software zu registrieren:

Registrieren Sie Ihre Kopie von Crimson 3.1

Details

Name:

E-Mail:

Unternehmen:

Straße:

Ort:

Staat:

PLZ:

Land:

Produkt-ID:

Senden Sie mir Informationen über Crimson 3.1 Updates.

Senden Sie mir Informationen über Produkte von Red Lion.

Status

Sammle Benutzerinformationen.

Die Registrierung ist zwar optional, wir empfehlen Ihnen jedoch dringend, uns Ihre Kontaktdaten mitzuteilen, damit wir Sie über Crimson 3.1-Updates und die zugehörigen Produkte auf dem neuesten Stand halten können. Da für die Registrierung eine Internetverbindung erforderlich ist, können Sie den Vorgang überspringen, wenn keine solche Verbindung verfügbar ist. Solange Ihre Software nicht registriert ist, wird Crimson 3.1 Sie regelmäßig daran erinnern, dass Sie mit einer nicht registrierten Softwareversion arbeiten.

1.5 Überprüfen auf Updates

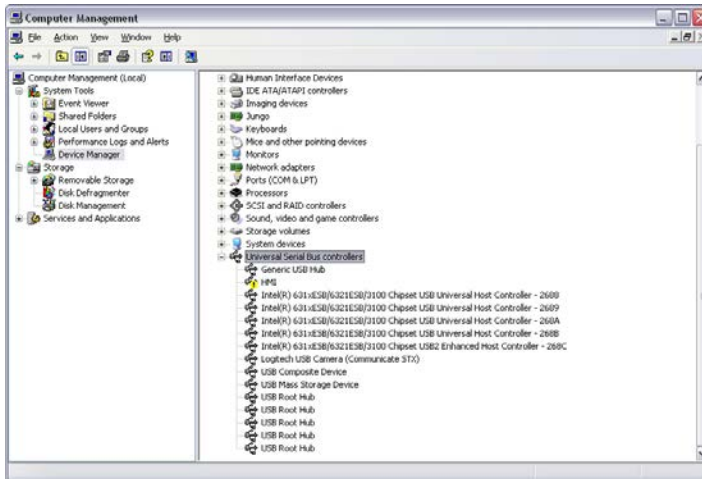
Wenn Sie über eine Internetverbindung verfügen, können Sie über den Befehl „Check for Update“ (Nach Update suchen) im Menü „Help“ (Hilfe) die Red Lion-Webseite nach einer neuen Crimson 3.1-Version durchsuchen. Wenn eine neuere Version als die von Ihnen verwendete gefunden wird, wird in Crimson die Frage eingeblendet, ob Sie das Upgrade herunterladen und die Software automatisch aktualisieren möchten. Sie können das Upgrade auch manuell von der Red Lion-Webseite herunterladen, indem Sie im Abschnitt „Support“ die Seite „Downloads“ besuchen.

1.6 Installation der USB-Treiber

Wenn Sie die mit der Zielhardware gelieferten Anweisungen befolgt haben, ist die Hardware noch nicht mit Ihrem PC verbunden. Nachdem die Installation von Crimson 3.1 abgeschlossen ist, können Sie das Gerät sicher über ein USB-Standardkabel anschließen. In Windows sollte nach einer kurzen Wartezeit die Meldung angezeigt werden, dass die Treiber für die neue Hardware gefunden wurden und dass die Hardware betriebsbereit ist. Es sollten keine weiteren Benutzereingriffe erforderlich sein.

1.7 Fehlerbehebung

Wenn Sie das Zielgerät mit dem PC verbunden haben, ohne zuerst Crimson zu installieren, ist es unter Umständen aufgrund einer abgebrochenen Installation nicht mehr möglich, die Treiber richtig zu installieren. Um dies zu prüfen, öffnen Sie den Windows Geräte-Manager. Klicken Sie mit der rechten Maustaste auf das Symbol „Arbeitsplatz“ und wählen Sie den Befehl „Verwalten“ aus. Daraufhin wird ein Fenster angezeigt, das der nachstehenden Abbildung ähnelt:

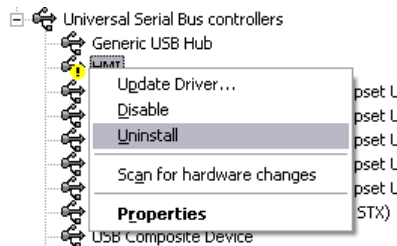


Wie Sie vorgehen müssen, um dieses Fenster anzuzeigen, ist vom verwendeten Betriebssystem abhängig. Der Grundgedanke ist aber in allen Systemen gleich: Suchen Sie nach dem Symbol „Arbeitsplatz“, das sich entweder auf dem Desktop oder im Startmenü befindet. Klicken Sie mit der rechten Maustaste darauf und wählen Sie „Verwalten“ aus. Falls das nicht funktioniert, wählen Sie in der Systemsteuerung die Option „System“ aus und aktivieren Sie dann auf der Registerkarte „Hardware“ den Geräte-Manager.

Wenn ein Problem mit den USB-Treibern vorliegt, wird in der Kategorie „Universal Serial Bus Controller“ ein gelbes Symbol mit einem Ausrufezeichen angezeigt. Der Name des Symbols lautet „HMI“ oder „Loader“ oder ähnlich. Der defekte Treiber ist in der nachstehenden Abbildung vergrößert abgebildet:



Um das Problem zu beheben, klicken Sie mit der rechten Maustaste auf das defekte Gerät und wählen im daraufhin eingblendeten Menü den Befehl „Deinstallieren“ aus:



Sie werden aufgefordert, den Vorgang zu bestätigen. Danach wird das Gerät von Windows vom System entfernt. Sie können das Crimson 3.1-Zielgerät jetzt ausschalten. Warten Sie ein paar Sekunden und schalten Sie das Gerät wieder ein. Windows startet die Treiberinstallation von Neuem.

Wie oben erwähnt, verwendet Crimson 3.1 unterschiedliche Gerätetreiber für den Bootloader und die Crimson-Laufzeit. Möglicherweise müssen Sie diesen Reparaturvorgang für jeden Treiber wiederholen, obwohl es unwahrscheinlich ist, dass bei einer fehlgeschlagenen Installation mehr als der Bootloader in Mitleidenschaft gezogen wurde.

1.8 Hilfe und Unterstützung

Wenn Sie auf ein Problem stoßen oder Hilfe benötigen, sind mehrere Ressourcen verfügbar.

Sprechblasenhilfe

Crimson 3.1 enthält eine sehr nützliche Funktion namens Sprechblasenhilfe:



Mit dieser Funktion können Sie Hilfeinformationen für jedes Element in Crimson 3.1 anzeigen. Sie wird über das Symbol am rechten Rand der Symbolleiste oder über Optionen im Menü „Help“ (Hilfe) gesteuert. Im Standardmodus können Hilfetexte durch Drücken der Taste F1 angezeigt werden. Dies ist insbesondere nützlich, wenn Sie sich bezüglich der Einstellungen für ein bestimmtes Feld unsicher sind und schnell Informationen benötigen. Wenn Sie daran denken, wird Ihnen die Bedienung des Programms wesentlich leichter fallen!

Technischer Support

Technische Unterstützung finden Sie im Web unter:

www.support.redlion.net

Alternativ können Sie auch die Telefonnummer +1 (717) 767-6511 anrufen und nach dem HMI-Supportteam fragen.

Online-Foren

Es gibt eine Reihe von Online-Foren zur Unterstützung der Benutzer von SPS und HMIs. Red Lion empfiehlt das Q&A-Forum unter <http://www.plctalk.net/qanda/>. In diesem Diskussionsforum sind viele Experten vertreten, die gern bereit sind, anderen zu helfen. Darüber hinaus verfolgen auch Mitarbeiter des technischen Supports von Red Lion die Diskussionen in diesem Forum, bei denen es um unsere Produkte geht.

1.9 Die nächsten Schritte

Wenn Sie von Crimson 2 migrieren, empfehlen wir, das nächste Kapitel zu überfliegen, um sich über die neue Benutzeroberfläche zu informieren. Weiterhin schlagen wir vor, dass Sie sich die Kapitel über Tags und die Konfiguration der Anzeigeseiten ansehen, weil einige der von Crimson 2 verwendeten Konzepte vereinfacht wurden und viele Dinge durch einfachere Methoden erreicht werden können. Wenn Sie bisher noch nicht mit Crimson 3.1 gearbeitet haben, sollten Sie mindestens bis zum Kapitel über Widgets lesen.

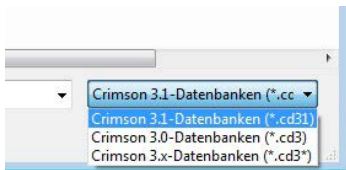
Viel Erfolg und viel Spaß!

Kapitel 2 Umstieg von Crimson 3.0

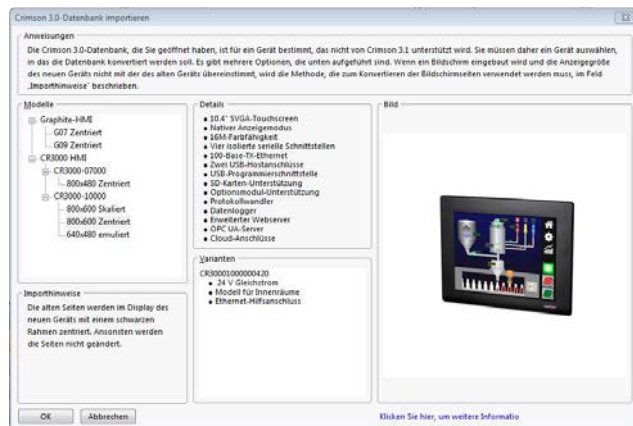
Dieses Kapitel enthält eine kurze Anleitung zur Migration Ihrer Datenbank von Crimson 3.0 zu Crimson® 3.1 sowie eine Liste einiger neuer Funktionen, die Sie nützlich finden dürften. Darüber hinaus wird kurz darauf eingegangen, welche Unterschiede zwischen den beiden Versionen hinsichtlich der unterstützten Hardware bestehen und wie diese Unterschiede sich auf den Datenbankimport auswirken. Wenn Sie ein neuer Benutzer sind, können Sie dieses Kapitel überspringen und direkt zum Kapitel „Crimson-Grundlagen“ gehen.

2.1 Importieren von Datenbanken

Datenbanken, die in Crimson 3.0 erstellt wurden, können in Crimson 3.1 direkt geöffnet werden. Dazu müssen Sie lediglich unten rechts im Dialogfeld „File Open“ (Datei öffnen) einen anderen Dateityp auswählen. Wenn Sie die Datenbank speichern möchten, stellt Crimson fest, dass die Datei im Crimson 3.1-Format gespeichert werden muss. Daraufhin wird ein Dialogfeld mit der Frage angezeigt, wo Sie die Datei speichern möchten und unter welchem Namen. Der neue Name ist in der Regel mit dem alten Namen identisch; allerdings wird die Erweiterung cd3 durch cd31 ersetzt. Crimson 3.0-Datenbanken können auch über den Befehl „File Import“ (Datei importieren) geöffnet werden. Dieser Befehl führt im Wesentlichen den gleichen Vorgang aus, aber es ist nicht erforderlich, den ausgewählten Dateityp zu ändern.



Wenn Sie eine Crimson 3.0-Datenbank öffnen oder importieren, die für ein Produkt konzipiert wurde, das nicht von Crimson 3.1 unterstützt wird, werden Sie gefragt, zu welchem Gerät Sie die Datei konvertieren möchten. Je nachdem, welche Produkte am besten dazu in der Lage sind, die Bildschirmauflösung des Originalgeräts zu unterstützen, stehen Ihnen verschiedene Optionen zur Auswahl.

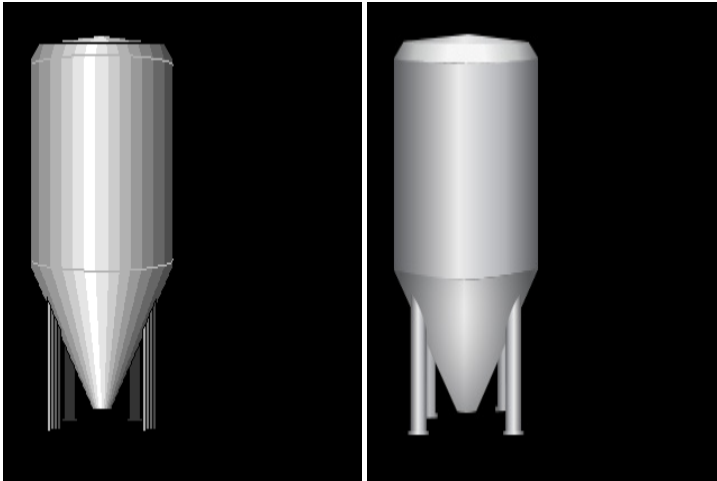


Wenn die Bildschirmauflösung des Zielgeräts nicht mit dem des ursprünglichen Geräts übereinstimmt, werden möglicherweise mehrere Optionen zur Konvertierung der Anzeigeseiten angeboten. Optionen, die als *Scaled* (Skaliert) gekennzeichnet sind, passen den Inhalt der einzelnen Anzeigeseiten an das neue Bildschirmformat an und passen dabei nach Möglichkeit die Schriftgrößen an. Optionen, die als *Centered* (Zentriert) gekennzeichnet sind, lassen den Inhalt der Anzeigeseiten unverändert, zentrieren ihn aber in der neuen Bildschirmgröße, sodass entweder vertikale Balken zu sehen sind oder der Inhalt von einem Rahmen umgeben ist. Optionen, die als *Emulated* (Emuliert) gekennzeichnet sind, schalten den Bildschirm des Zielgeräts in einen Modus um, in dem die Bildschirmausgabe skaliert wird, um die Auflösung des Originalgeräts nachzuahmen. Wenn die Skalierung anhand

eines Integralfaktors erfolgt, ergibt dies in der Regel eine klare Anzeige, jedoch mit größeren und damit auffälligeren Pixeln. Wenn ein nichtintegraler Faktor verwendet wird, sollten Sie die Ergebnisse in Augenschein nehmen, um festzustellen, ob die resultierende Anzeigequalität akzeptabel ist.

2.2 Symbolauswahl

Wenn Sie eine Crimson 3.0-Datenbank importieren, stellt die Symbolbibliothek der Datenbank weiterhin alte Symbole bereit, die aber nicht die für die neueren Versionen charakteristischen glatteren Kanten und subtileren Schattierungen aufweisen. Der Unterschied wird in den folgenden Abbildungen veranschaulicht. Bei dem links stehenden Behälter handelt es sich offensichtlich um die alte Version:



Sie können in der Datenbank zu den neuen Symbolen wechseln, indem Sie die Kategorie „Display Pages“ (Anzeigeseiten) aufrufen und im Stamm der Navigationsliste das Element „Pages“ (Seiten) auswählen. Bearbeiten Sie auf der Registerkarte „Images“ (Bilder) die Eigenschaft *Symbol Library* (Symbolbibliothek), um den bevorzugten Symbolsatz auszuwählen. Für die meisten alten Symbole gibt es zwar eine neue Entsprechung, doch es kann vorkommen, dass das Rendering nicht genau gleich ist. Beispielsweise kann ein Symbol einen Pixel schmaler oder breiter ausfallen. Wenn dies Auswirkungen auf Ihre Datenbank hat, müssen Sie möglicherweise entscheiden, ob Sie die einzelnen Seiten manuell anpassen oder weiterhin die alten Symbole verwenden möchten.

2.3 Modellkonvertierung

Crimson 3.1 unterstützt nicht alle von Crimson 3.0 unterstützten Modelle. Um die Funktionen zu bieten, die unsere Kunden verlangten, mussten wir die Unterstützung älterer, weniger leistungsfähiger Hardware einstellen und uns ausschließlich auf Geräte konzentrieren, die über ausreichend Arbeitsspeicher und Rechenleistung verfügen und somit die Anforderungen moderner Software erfüllen können.

2.3.1 Konvertieren von Graphite aus

Alle Graphite HMIs und Graphite Controller werden von Crimson 3.1 unterstützt. Die Konvertierung Ihrer Datenbank besteht daher einfach aus dem oben beschriebenen Importprozess. Benutzerhandbuch

2.3.2 Konvertieren von G3 aus

Der Migrationspfad von der G3-Serie zur CR3000-Serie ist wegen der Unterschiede bei der Bildschirmauflösung zwischen den beiden Serien kompliziert. Die nachstehende Tabelle beschreibt die Optionen, die bei der Konvertierung zwischen verschiedenen Modellen verfügbar sind:

G3-Modell	CR3000-Modell	Beschreibung
G306A	7 Zoll	Der G306A-Bildschirm mit der Auflösung 320 x 240 kann durch Hardware- oder Softwareskalierung an den CR3000-Bildschirm mit der Auflösung 800 x 480 angepasst werden. Die Hardwareskalierung konvertiert jedes Pixel in einen 2 x 2 großen Pixelblock. Die Softwareskalierung passt die Größe der Primitiven an das neue Format an. Möglicherweise müssen Sie bei der Softwareskalierung die Seiten bearbeiten, um Schriftgrößen, Linienbreiten usw. anzupassen.
G308, G310	10 Zoll	Der G308- und der G310-Bildschirm haben beide die Auflösung 640 x 480, die durch Hardware- oder Softwareskalierung an den CR3000-Bildschirm mit der Auflösung 800 x 600 angepasst werden kann. Die Hardwareskalierung mischt benachbarte Pixel. Die Ergebnisse sind daher möglicherweise nicht perfekt. Möglicherweise müssen Sie bei der Softwareskalierung die Seiten bearbeiten, um Schriftgrößen, Linienbreiten usw. anzupassen.
G315	15 Zoll	Der G315 kann direkt und ohne Bildschirmanpassungen importiert werden.

Ein potenzielles Problem ist, dass der CR3000 weder über die MENU-Taste noch über die von G3 bereitgestellten Softkeys verfügt. Sie müssen daher jegliche Funktionalität von diesen Tasten entfernen und sie durch die auf dem Bildschirm angezeigten Schaltflächen und weiteren Elemente ersetzen, mit denen sich das gleiche Ergebnis erzielen lässt. Wir empfehlen Ihnen, dies in Crimson 3.0 vor der Konvertierung in Crimson 3.1 zu erledigen.

2.3.3 Konvertieren von G3 Kadet aus

In puncto Bildschirmauflösung kann der G304K2 sowohl direkt in ein CR1000 4-Zoll-Modell, und ein G307K2 kann direkt in ein CR1000 7-Zoll-Modell importiert werden. Die beiden Modelle unterscheiden sich jedoch hinsichtlich der seriellen Anschlüsse. Die CR1000-Modelle verfügen über einen einzelnen RS-232-Programmierschluss und einen RS-232- oder RS485-Kommunikationsanschluss im Dualmodus. Der G304K verfügt über einen Anschluss im Dualmodus und einen RS-485-Anschluss. Wenn Ihre Anwendung zwei RS-485-Anschlüsse benötigt, müssen Sie einen Adapter verwenden. Der G307K verfügt über einen RS-232-Programmierschluss und zwei Anschlüsse im Dualmodus. Wenn Ihre Anwendung drei serielle Verbindungen benötigt, ist der CR1000 nicht geeignet, und Sie müssen stattdessen ein CR3000-Modell verwenden.

2.3.4 Konvertierung von Data Station Plus aus

Die DSPLE kann direkt in einen DA10 importiert werden. Dieses Modell verfügt jedoch über einen RS-232-Programmierschluss und einen RS-232- oder RS-485-Kommunikationsanschluss mit Dualmodus. Wenn Ihre Anwendung drei serielle Verbindungen benötigt, ist der DA10 nicht geeignet, und Sie müssen stattdessen ein DA30-Modell verwenden.

Die Modelle DSPSX, DSPGT und DSPZR können direkt in das DA30-Modell konvertiert werden. Dabei werden die nativen virtuellen Bildschirmauflösungen und das Layout der seriellen Kommunikationsanschlüsse unverändert unterstützt. Es kann jedoch vorkommen, dass das gleiche Problem wie bei G3 auftritt, wo MENÜ und Softkeys im DA30 nicht unterstützt werden. Daher müssen Sie alle Funktionalitäten von diesen Tasten entfernen und sie durch die auf dem Bildschirm angezeigten Schaltflächen und weiteren Elemente ersetzen, mit denen sich das gleiche Ergebnis erzielen lässt. Wir empfehlen, dies in Crimson 3.0 vor der Konvertierung in Crimson 3.1 zu erledigen.

2.3.5 Konvertieren von anderen Geräten aus

Der Modular Controller und die ProductVity Station bieten derzeit keine Unterstützung für Crimson 3.1. Diese Produkte müssen daher weiterhin mit Crimson 3.0 programmiert werden. Red Lion Controls, Inc. wird aktualisierte

Hardware veröffentlichen, mit der Sie den Wechsel zu Crimson 3.1 vollziehen können. Weitere Informationen finden Sie auf der Red Lion-Webseite.

2.4 Benutzerdefinierte Webseiten

Wenn Sie in Crimson 3.0 eine benutzerdefinierte Webseite entwickelt haben, funktioniert diese nach dem Wechsel zu Crimson 3.1 u. U. nicht mehr genau wie vorher. Es wird empfohlen, das Kapitel zum Erstellen von benutzerdefinierten Webseiten zu lesen und zu überlegen, wie die neu verfügbaren Funktionen Ihnen helfen können, Ihre Site mit den Crimson 3.1-Standardseiten zu harmonisieren.

2.5 Neue Funktionen

Crimson 3.1 bietet viele neue Funktionen und Verbesserungen. Es wurden viele Änderungen vorgenommen, um die Benutzerfreundlichkeit zu erhöhen und den Zeitaufwand für das Erstellen einer Datenbank zu verringern. Außerdem wurde eine Reihe von wichtigen neuen Funktionen hinzugefügt.

2.5.1 Zero Config

- Crimson 3.1 unterstützt eine Zero-Configuration-Netzwerk-Option, mit der ein Gerät mit Ihrem Netzwerk verbunden und namentlich von Ihrem PC referenziert werden kann, ohne dass DNS- oder HOST-Dateieinträge hinzugefügt werden müssen. Crimson 3.1 erstellt außerdem ein Webserver-SSL-Zertifikat mit diesem Namen, wodurch eine fehlerfreie Entwicklung sicherer Webseiten ermöglicht wird.

2.5.2 OPC UA-Server

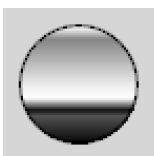
- Crimson 3.1 bietet einen OPC UA-Server in mittleren und High-End-Einheiten. Dieser Server macht ausgewählte Tags über das OPC-UA-Protokoll verfügbar, sodass geeignete Clients die Tag-Ordnerstruktur durchsuchen und Tag-Werte lesen und schreiben können. Dank Abonnementunterstützung können Clients nur Updates empfangen, wenn sich Tags um den angegebenen Totzonenwert ändern.

2.5.3 Neue Primitive

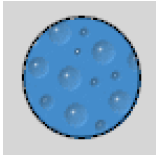
- Crimson 3.1 bietet attraktivere Primitive mit glatteren Kanten:



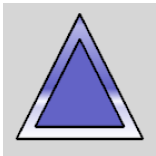
- Neuere Primitive können mit einer größeren Vielfalt von Verlaufsmustern gefüllt werden, einschließlich Mustern, die zylindrische Container oder metallische Oberflächen simulieren.



- Neuere Primitive können mit einer Vielzahl von Texturen aus der Symbolbibliothek gefüllt werden, wodurch natürliche oder künstliche Materialien dargestellt werden können.



- Neuere Primitive unterstützen Ränder beliebiger Dicke und bieten eine bessere Anzeigqualität auf hochauflösenden Geräten, bei denen ein einzelnes Pixel sehr klein ist. Diese Ränder können zwar eine Volltonfarbe haben, sie können aber auch mit einem der Verlaufsmuster oder Texturen gefüllt werden, mit denen die Form selbst gefüllt wird. Innen oder außen kann eine durchgehende Linie auf den Rand angewendet werden, um ihn besser von anderen Anzeigeelementen abzugrenzen.



- Crimson 3.1 bietet ein vielfältigeres Linien-Primitiv, das eine beliebige Breite haben kann. Die Linie kann genau wie andere Primitive eine Füllung und einen Rand haben. Linien können daher mit abgestuften Füllungen und Texturen gefüllt werden, und auf ihre Umrise können Ränder angewendet werden. Es können auch verschiedene Linienendstile angewendet werden, sodass Pfeile einfach erstellt werden können.



- Crimson 3.1 bietet flexiblere abgerundete, abgeschrägte und verrundete Formen sowie die Möglichkeit, den Eckeneffekt bei beliebig vielen Ecken auszulassen. Dies ermöglicht größere Flexibilität beim Erstellen benutzerdefinierter Formen.

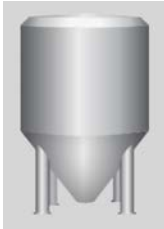


- Crimson 3.1 unterstützt neue Primitive für Dreiecke, Parallelelogramme und Trapeze durch proportionale Griffe, mit denen die Formen so angepasst werden können, dass die Proportionen beim Skalieren unverändert bleiben. Crimson 3.1 bietet außerdem eine neue, flexiblere Pfeil-Primitive mit mehreren Griffen, mit denen das Erscheinungsbild angepasst werden kann.

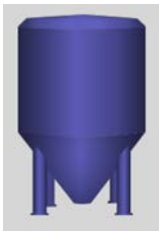


2.5.4 Symbolbibliothek

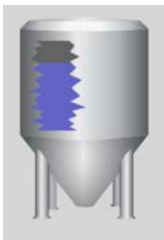
- Symbole haben glattere Kanten und subtilere Schattierungen.



- Symbole können zur Laufzeit dynamisch neu gefärbt werden, um Statusinformationen darzustellen.



- Symbole können zur Laufzeit dynamisch mit einer Behälterfüllung neu gefärbt werden, sodass beispielsweise die Behälterschnitt-Symbole mit den Behältersymbolen kombiniert werden können, um den Behälterfüllstand auf attraktive Weise darzustellen.



2.5.5 Webserver

- Der Webserver wurde vollständig überarbeitet und hat nun ein nützlicheres Erscheinungsbild, das auf den Bootstrap- und JQuery-Bibliotheken beruht. Die Seitenlayouts werden bei Verwendung von Mobilgeräten automatisch angepasst.



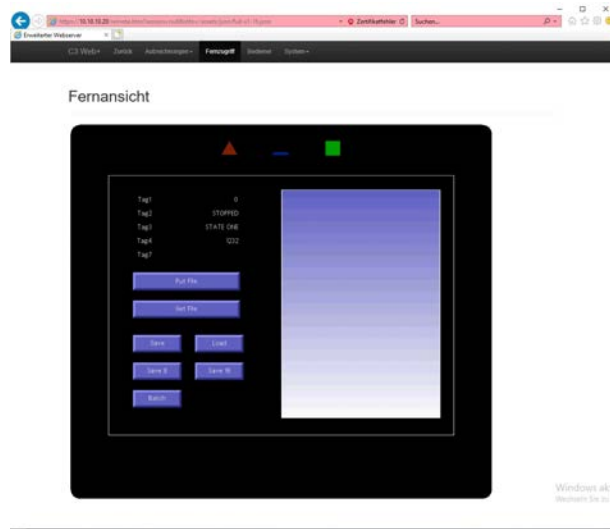
Willkommen zum Erweiterten Webserver!

Option	Beschreibung
Zurücksetzen	Daten wie alle Controller laden
Profil	Interaktiver Ansicht des Controller-Status
Daten-Modell	Zugriff auf die Daten-Modell des Geräts
Systemlog	Zugriff auf Log-Informationen über den Server

- Der Webserver unterstützt jetzt einen sicheren HTTPS-Betrieb und ermöglicht die Bereitstellung von Zertifikaten, um die Identität des Servers anzugeben. HTTP-Umleitung wird ebenfalls unterstützt, sodass Ihre Clients das Präfix https nicht eingeben müssen.
- Der Webserver bietet jetzt eine formularbasierte Authentifizierung für sichere Verbindungen und dadurch eine bequemere Anmeldung, einschließlich der Umleitung zu zuvor ausgewählten Seiten. Der Webserver

unterstützt jetzt einzelne Sicherheitsbeschreibungen für jede Funktion und für jede Tag-gesteuerte Seite. Dies ermöglicht eine detailliertere Zugangssteuerung.

- Der Webserver unterstützt jetzt die Eingabe direkt von der Tastatur, wenn die Fernsteuerung verwendet wird. Der Benutzer muss die Schaltflächen im Popup-Tastenfeld nicht mehr mit der Maus betätigen. Daten werden bei der Eingabe in das Feld eingefügt.
- Der Webserver enthält jetzt eine aktualisierte Fernansicht und Fernsteuerungsfunktion, die wesentlich reibungsloser arbeitet und nur einen Bruchteil der Daten sendet, die von der vorherigen Implementierung benötigt wurden. Gleichzeitig wird die Farbtiefe von 8 auf 16 Bit erhöht.



- Der Webserver verwendet nun AJAX-Technologie, um Tag-gesteuerte Seiten zu aktualisieren. Updates sind jetzt deutlich schneller, ohne Flimmern und mit minimaler Datennutzung. Die gleichen Techniken sind für benutzerdefinierte Webseiten verfügbar.
- Der Webserver unterstützt jetzt die Antwortkomprimierung. Zusammen mit den letzten beiden Funktionen trägt dies zu einer erheblichen Verringerung der Datenanforderungen bei. Dies ist besonders wichtig für den Mobilfunkbetrieb, und zwar sowohl für die Leistung als auch für die Abrechnungskontrolle.
- Mit dem Webserver können benutzerdefinierte Stylesheets und JavaScript in jede Seite eingebunden werden, sodass Erscheinungsbild und Verhalten der Site geändert werden können.

2.5.6 IEC-61131

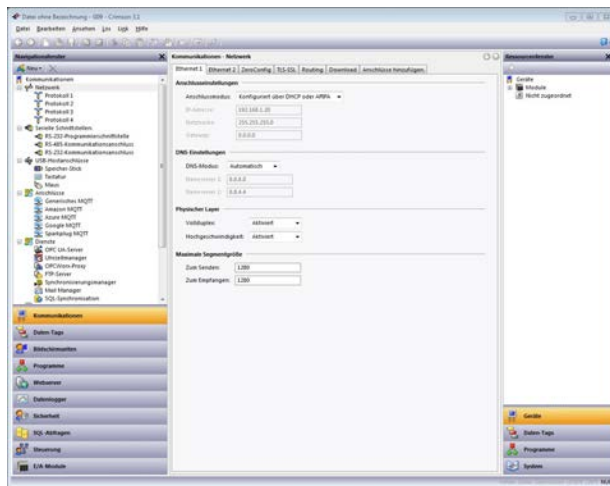
- Die Steuer-Engine unterstützt jetzt eine deterministischere Ausführung. Dadurch wird sichergestellt, dass Abtastzeiten und E/A-Updates nicht durch Seitenaktualisierungen und andere Aktivitäten unterbrochen werden.
- Der Steuerungseditor unterstützt jetzt Online-Debugging, sodass die Ausführung eines Programms auf einem externen Gerät in Echtzeit überwacht werden kann.
- Der Steuerungseditor bietet jetzt eine bessere Unterstützung für die Offline-Simulation, sodass die Navigation zwischen Seiten möglich ist, ohne den Simulationsmodus zu verlassen.
- Der Steuerungseditor unterstützt nun Vorgänge mit der Zwischenablage und Drag-and-Drop, sodass Programme und Variablen innerhalb von Crimson 3.1 und sogar zwischen Crimson-Datenbanken verschoben werden können.
- Der Steuerungseditor wurde optimiert, um eine flüssigere Navigation zu ermöglichen und einige der früher feststellbaren Unterbrechungen zu vermeiden.
- Mit dem Steuerungseditor können Anfangswerte für Variablen definiert werden, um sicherzustellen, dass sie beim Systemstart richtig initialisiert werden.

Kapitel 3 Crimson-Grundlagen

Wählen Sie im Abschnitt Red Lion Controls, Inc des Startmenüs das Symbol Crimson 3.1 aus, um Crimson® 3.1 auszuführen. Nach ein paar Sekunden wird Crimson angezeigt. Wenn Sie bisher mit Crimson 2 gearbeitet haben, dürfte Ihnen als Erstes auffallen, dass die Benutzeroberfläche überarbeitet wurde. Die neue Benutzeroberfläche ermöglicht eine flinkere Navigation und schnellere Datenbankerstellung.

3.1 Fensterlayout

Das Hauptfenster von Crimson 3.1 besteht aus drei Bereichen, wie aus der nachstehenden Abbildung hervorgeht.



3.1.1 Das Navigationsfenster

Der linke Bereich des Fensters wird als Navigationsfenster bezeichnet. Hier können Sie zwischen den verschiedenen Elementkategorien in einer Crimson 3.1-Konfigurationsdatei hin und her wechseln. Jede Kategorie wird durch eine Leiste im unteren Fensterbereich dargestellt. Wenn Sie auf die Leiste klicken, navigieren Sie zum entsprechenden Abschnitt. Der obere Bereich des Navigationsfensters zeigt die in der aktuellen Kategorie verfügbaren Elemente an, und es wird eine Symbolleiste zur Bearbeitung dieser Elemente bereitgestellt. Wenn Sie den oberen Bereich vergrößern möchten, ziehen Sie einfach die Trennlinie zwischen dem Bereich und den Kategorieleisten, bis die gewünschte Größe erreicht ist.

3.1.2 Das Ressourcenfenster

Der rechte Bereich des Fensters wird als Ressourcenfenster bezeichnet. Hier haben Sie Zugriff auf verschiedene Elemente, die bei der Bearbeitung der aktuellen Kategorie nützlich sind. Genau wie im Navigationsfenster gibt es auch hier eine Reihe von Kategorien, auf die Sie über die Kategorieleisten zugreifen können. Elemente in einer bestimmten Ressourcenkategorie können per Drag-and-Drop an die Stellen gezogen werden, an denen Sie sie verwenden möchten. Beispielsweise können Sie ein Daten-Tag aus dem Ressourcenfenster auf ein Konfigurationsfeld ziehen, um das Feld vom Wert des ausgewählten Tags abhängig zu machen. Viele Elemente lassen auch einen Doppelklick zu, wodurch das aktuelle Feld auf dieses Element eingestellt wird.

3.1.3 Das Bearbeitungsfenster

Der mittlere Bereich des Fensters dient zur Bearbeitung des gegenwärtig ausgewählten Elements. Was in diesem Fenster angezeigt wird, richtet sich nach dem ausgewählten Element. Beispielsweise kann es eine Reihe von Registerkarten enthalten, auf denen die Eigenschaften dieses Elements angezeigt werden, oder aber einen Editor speziell für das Element, an dem Sie arbeiten.

3.1.4 Zuklappen von Fenstern

Das Navigationsfenster und/oder das Ressourcenfenster können am Rand des Hauptfensters zugeklappt werden, um mehr Platz für das Bearbeitungsfenster freizugeben. Zum Schließen eines Fensters klicken Sie in dessen Titelleiste auf das „X“ oben links. Das Fenster wird dann durch eine kleinere, mit Pfeilen gekennzeichnete Leiste ersetzt. Durch einen Klick auf diese Leiste wird das zugehörige Fenster wieder aufgeklappt. Durch einen Klick auf das Reißzwecken-Symbol eines vorübergehend aufgeklappten Fensters wird das Fenster an der aktuellen Stelle verankert.

3.2 Die Kategorien

In diesem Abschnitt werden die Hauptkategorien einer Crimson 3.1-Datenbank beschrieben.

3.2.1 Kommunikationen



Diese Kategorie gibt an, welche Protokolle an den seriellen und Ethernet-Anschlüssen des Zielgeräts verwendet werden sollen. Wenn Master-Protokolle verwendet werden (d. h. Protokolle, in denen die Red Lion-Hardware die Datenübertragung an ein und von einem externen Gerät initiiert), können Sie mit diesem Symbol auch ein oder mehrere Geräte angeben, auf die zugegriffen werden soll. Wenn Slave-Protokolle verwendet werden (d. h. Protokolle, bei denen die Red Lion-Hardware Anforderungen von anderen Systemen empfängt und darauf reagiert), können Sie festlegen, welche Datenelemente für Lese- oder Schreibzugriff zugänglich sein sollen. Sie können diese Kategorie auch verwenden, um über den Protokoll-Konverter Daten zwischen externen Geräten zu verschieben sowie um Erweiterungskarten und Dienste zu konfigurieren.

3.2.2 Daten-Tags



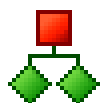
Diese Kategorie definiert die Datenelemente, die in den externen Geräten als Zugriffsdaten oder im Zielgerät zum Speichern von Informationen verwendet werden sollen. Jedes Tag hat eine Reihe von Eigenschaften, einschließlich Formatierungsdaten, die angeben, wie die in einem Tag enthaltenen Daten auf dem Bildschirm des Geräts oder in anderen Kontexten wie etwa Webseiten angezeigt werden sollen. Wenn Sie diese Informationen innerhalb des Tags angeben, ist es in Crimson nicht mehr nötig, jedes Mal erneut die Formatierungsdaten einzugeben, wenn ein Tag angezeigt wird. Zu den fortschrittlicheren Tag-Eigenschaften gehören Alarme, die aktiviert werden können, wenn verschiedene Bedingungen im Zusammenhang mit dem Tag eintreten, oder Auslöser, die programmierbare Aktionen durchführen, wenn diese Bedingungen erfüllt sind.

3.2.3 Anzeigeseiten



Mit dieser Kategorie werden Anzeigeseiten erstellt und bearbeitet. Mit dem Seiteneditor können Sie verschiedene grafische Elemente anzeigen, die als „Primitive“ bekannt sind. Unter diesen Begriff fallen einfache Elemente wie Rechtecke und Linien sowie komplexere Elemente, die mit dem Wert eines bestimmten Tags oder mit einem Ausdruck verknüpft werden können. Primitive übernehmen standardmäßig die Formatierungsinformationen, die bei der Erstellung des Tags definiert wurden. Diese Informationen können jedoch bei Bedarf überschrieben werden. Mit dem Editor können Sie außerdem angeben, welche Aktionen ausgeführt werden sollen, wenn Tasten oder Primitive gedrückt, losgelassen oder gedrückt gehalten werden.

3.2.4 Programme



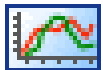
Diese Kategorie dient zum Erstellen und Bearbeiten von Programmen mit der Crimson-eigenen Programmiersprache, die der Programmiersprache C ähnelt. Diese Programme können komplexe Entscheidungsfindungs- oder Datenbearbeitungsvorgänge durchführen, die auf Datenelementen innerhalb des Systems basieren. Sie dienen dazu, die Funktionalität von Crimson über die in der Software enthaltenen Standardfunktionen hinaus zu erweitern, damit selbst die komplexesten Anwendungsbereiche bewältigt werden können. Programme können sich an verschiedene Systemfunktionen wenden, um geläufige Vorgänge durchzuführen.

3.2.5 Webserver



Diese Kategorie dient zum Konfigurieren des Webserver von Crimson 3.1 sowie zum Erstellen und Bearbeiten von Webseiten. Der Webserver kann den Fernzugriff auf das Zielgerät über verschiedene Mechanismen bereitstellen. Erstens können Sie mit Crimson automatische Webseiten erstellen, die Tag-Listen enthalten und jeweils den Tag-Eigenschaften entsprechend formatiert sind. Zweitens können Sie mit dem HTML-Editor eines Drittanbieters eine benutzerdefinierte Site erstellen und dann Direktiven einbinden, die Crimson anweisen, Live-Tag-Werte einzufügen. Drittens können Sie die einzigartige Fernzugriffs- und -steuerungsfunktion von Crimson aktivieren, mit der ein Webbrowser den Bildschirm des Zielgeräts anzeigen und dessen Tastatur steuern kann. Außerdem können Sie mit dem Webserver auf Dateien vom Datenlogger zugreifen.

3.2.6 Datenlogger



Diese Kategorie dient zum Erstellen und Verwalten von Datenprotokollen, die jeweils eine beliebige Anzahl von Variablen auf der Speicherkarte des Zielgeräts aufzeichnen können. Die Daten können bis zu einmal pro Sekunde aufgezeichnet werden. Die aufgezeichneten Werte werden in CSV-Dateien (Comma Separated Variable, durch Kommata getrennte Variablen) gespeichert, die sich leicht in Anwendungen wie Microsoft Excel importieren lassen. Für den Zugriff auf die Dateien stehen verschiedene Möglichkeiten zur Auswahl. Sie können die Speicherkarte herausnehmen; die Karte als Laufwerk auf einem PC anmelden, der mit dem USB-Anschluss des Zielgeräts verbunden ist; oder die Dateien mithilfe eines Ethernet-Anschlusses oder Modems von den Webservern oder FTP-Servern von Crimson abrufen. Protokolldateien können durch kryptographische Signaturen geschützt werden, um sicherzustellen, dass sie seit ihrer Speicherung auf der Speicherkarte nicht manipuliert wurden.

3.2.7 Sicherheit



Diese Kategorie dient zum Erstellen und Verwalten der verschiedenen Benutzer des Zielgeräts sowie der Zugriffsrechte, die den einzelnen Benutzern gewährt wurden. Es können auch echte Namen angegeben werden. In diesem Fall kann die Sicherheitsprotokollierung nicht nur erfassen, welche Daten wann verändert wurden, sondern auch von wem. Die zum Ändern eines bestimmten Tags oder zum Zugriff auf eine Seite erforderlichen Rechte werden über die Sicherheitseigenschaften des einzelnen Elements festgelegt. Es können auch Rechte zugewiesen werden, mit denen der Zugriff auf den FTP-Server oder den Webserver gestattet oder verweigert wird.

3.2.8 SQL-Abfragen



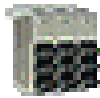
Diese Kategorie dient zum Erstellen und Verwalten von Abfragen, die in SQL geschrieben sind. Mit diesen Abfragen werden Daten aus einer Microsoft SQL Server-Datenbank extrahiert und in einen Satz Crimson-Tags übertragen. Es können mehrere Abfragen erstellt werden, wobei jede Spalte der zurückgegebenen Daten einem Satz Tags zugeordnet wird, die die Zeilen dieser Daten darstellen. Abfragen können regelmäßig oder nach Bedarf ausgeführt werden. Dies bedeutet beispielsweise, dass Informationen zur Produktionsplanung an einen Controller übergeben werden können, der für den Betrieb der Maschinen verantwortlich ist.

3.2.9 Steuerung



Diese Kategorie ermöglicht die Erstellung und Bearbeitung von IEC-61131-Programmen in den verschiedenen Sprachen, die in diesem Standard definiert sind. Die Programme können in regelmäßigen Abständen oder nach Bedarf aufgerufen werden. Sie arbeiten möglicherweise mit beliebigen Crimson-Tags oder E/A-Geräten zusammen, um eine Steuerfunktionalität bereitzustellen, die über die von der Crimson-eigenen Programmiersprache unterstützte Funktionalität hinausgeht. Eine vollständige Online-Überwachung wird bereitgestellt. Eine Offline-Simulation ermöglicht Ihnen, das Verhalten ihrer Programme zu debuggen, ohne ein externes Gerät anschließen zu müssen.

3.2.10 E/A-Module



Diese Kategorie ist nur beim Konfigurieren eines Graphite HMI oder eines anderes Geräts verfügbar, das Graphite-Module unterstützt. Sie ermöglicht Ihnen, die E/A-Module konfigurieren, die Sie direkt mit dem Gerät bzw. über ein Erweiterungs- oder angebundenes Rack verbunden haben. Beachten Sie, dass die Kategorie nur für die Konfiguration von E/A-Modulen vorgesehen ist. Module, die für die Kommunikation verwendet werden, werden über die Kategorie „Kommunikationen“ mithilfe eines Mechanismus konfiguriert, der im Kapitel „Verwenden von Kommunikationen“ eingehender erläutert wird.

3.3 Navigation

Die einfachste Methode, durch eine Crimson 3.1-Datenbank zu navigieren, besteht darin, auf die Kategorieleisten im Navigationsfenster und dann auf das Element zu klicken, das Sie bearbeiten möchten. Es gibt aber eine Reihe von Tastaturkürzeln, die eine noch schnellere Navigation ermöglichen und die Produktivität steigern. Die meisten dieser Tastaturkürzel können über das Menü „Go“ oder über die entsprechenden Tastenkombinationen aufgerufen werden.



3.3.1 Zurück und Vorwärts

Mit dem ersten Symbol auf der Symbolleiste oder der Tastenkombination **ALT+LINKS** gehen Sie zurück zu Elementen, die Sie zuvor ausgewählt hatten. Mit dem nächsten Symbol oder der Tastenkombination **ALT+RECHTS** bewegen Sie sich wieder vorwärts, d. h. Sie kehren zum ursprünglichen Element zurück. Diese Funktion ist sehr nützlich, wenn Sie zwischen Datenbankkategorien hin und her wechseln.

3.3.2 Tastaturkürzel für Kategorien

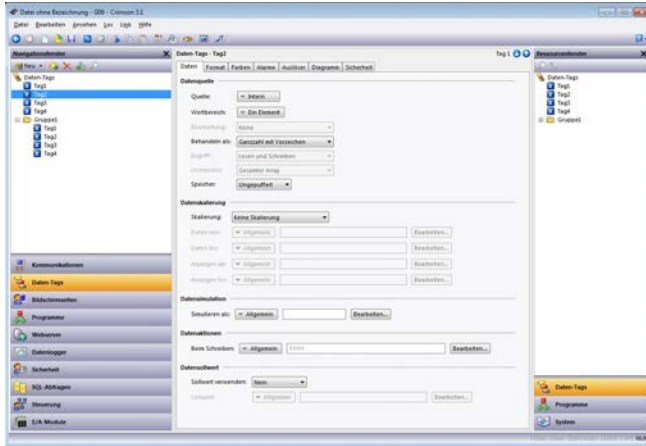
Jeder Kategorie ist eine Tastenkombination zugeordnet, die aus der Taste **STRG** und einer Zahl besteht. Die Zahl gibt dabei die Position der Kategorie im Navigationsfenster an. Beispielsweise kann der Abschnitt „Kommunikationen“ direkt über die Tastenkombination **STRG+1** aufgerufen werden. Mit den Tastenkombinationen **ALT+BILD AUF** und **ALT+BILD AB** können Sie sich in der Kategorieliste nach oben und unten bewegen.

3.3.3 Tastaturkürzel für Elemente

Im Bearbeitungsfenster können Sie mit den Tastenkombinationen **ALT+NACH OBEN** und **ALT+NACH UNTEN** zwischen Elementen wechseln. Crimson wechselt zum vorherigen bzw. nächsten Element in der Elementliste und versucht, das aktuell ausgewählte Datenfeld beizubehalten. Dies ist sehr nützlich, wenn Sie dasselbe Feld in mehreren Elementen ändern möchten, weil Sie nicht wiederholt zu diesem Feld navigieren oder zum Navigationsfenster wechseln müssen, um Elemente zu ändern.

3.4 Navigationslisten

Mehrere der Kategorien in Crimson enthalten Elementlisten. Wenn Sie beispielsweise die Kategorie „Daten-Tags“ auswählen, wird im Navigationsfenster eine Liste aller Daten-Tags in Ihrer Datenbank angezeigt. Sie können dann die gewünschten Daten-Tags auswählen und bearbeiten:



Elemente in diesen Navigationslisten können auf verschiedene Weise bearbeitet werden:

- Wenn Sie schnell nach einem bestimmten Element suchen möchten, geben Sie die ersten paar Buchstaben seines Namens ein. Crimson wählt das erste Element aus, das mit den eingegebenen Zeichen übereinstimmt. Je mehr Zeichen Sie eingeben, desto spezifischer wird die Auswahl. Wenn Sie die **ESC**-Taste drücken, können Sie eine neue Suchzeichenfolge eingeben.
- Zum Erstellen eines Elements klicken Sie in der Symbolleiste des Navigationsfensters auf die Schaltfläche „New“ (Neu). Bei Listen, die nur einen einzelnen Elementtyp unterstützen, können Sie auch die Tastenkombination **ALT+EING** verwenden. Die Schaltfläche „New“ (Neu) in der Symbolleiste enthält möglicherweise eine Liste der verfügbaren Elemente. In diesem Fall können Sie den Typ des zu erstellenden Elements auswählen.
- Zum Löschen eines Elements verwenden Sie entweder das Symbol „Löschen“ in der Symbolleiste des Navigationsfensters oder drücken die Tastenkombination **ALT+ENTF**. Wenn Sie einen Ordner löschen, werden auch alle Elemente gelöscht, die sich in diesem Ordner befinden. Bei Mehrfachlösungen werden Warnungen angezeigt, doch Löschvorgänge dieser Art können immer mit dem Befehl „Undo“ (Rückgängig) rückgängig gemacht werden.
- Zum Umbenennen eines Elements wählen Sie das gewünschte Element aus und drücken **F2**. Geben Sie dann den neuen Namen ein, und drücken Sie die **EINGABETASTE**. Alternativ können Sie auch das Element auswählen und dann erneut ein Mal auf den Namen klicken, um die Bearbeitung zu aktivieren. Drücken Sie die **EINGABETASTE**, wenn Sie fertig sind.

3.4.1 Arbeiten mit Ordnern

Einige Listen unterstützen die Gruppierung von Elementen in Ordnern. Ordner können mit dem Symbol „Neuer Ordner“ in der Symbolleiste des Navigationsfensters erstellt und genau wie konventionellere Elemente umbenannt und gelöscht werden. Wenn beim Erstellen eines Elements ein Ordner ausgewählt ist, wird das Element im ausgewählten Ordner platziert. Ordner können bis zu einer beliebigen sinnvollen Tiefe verschachtelt werden.

3.4.2 Sortieren von Listen und Ordnern

Eine ganze Navigationsliste oder der Inhalt eines Ordners kann sortiert werden, indem Sie mit der rechten Maustaste auf das Stammelement oder den Ordner klicken und einen der Sortierbefehle auswählen. Elemente

können in aufsteigender oder absteigender alphabetischer Reihenfolge sortiert werden. Ordner werden unabhängig von der angewendeten Sortierreihenfolge immer vor anderen Elementen platziert.

3.4.3 Drag-and-Drop-Vorgänge

Elemente in Navigationslisten können per Drag-and-Drop an eine andere Stelle in der gleichen Liste oder zwischen Ordnern verschoben werden. Wenn Sie beim Ziehen die STRG-Taste gedrückt halten, wird eine Kopie des ursprünglichen Elements erstellt. Die Links-nach-rechts-Position eines Elements kann manchmal bei der Entscheidung helfen, wo das Element in der Ordnerhierarchie platziert werden soll. Wenn das Element im falschen Ordner abgelegt wird, bewegen Sie es nach links oder rechts, um es an die richtige Position zu bringen.

Datenbankelemente wie Tags und Anzeigeseiten können auch zwischen Datenbankdateien gezogen werden. Öffnen Sie dazu zwei Kopien von Crimson, und ziehen Sie die gewünschten Elemente aus dem Navigationsfenster der Quelldatenbank in das Navigationsfenster der Zieldatenbank. Wenn am Ziel noch nicht die richtige Kategorie ausgewählt ist, halten Sie das gezogene Element vorübergehend auf der erforderlichen Kategorieleiste. Nach einem Augenblick wird diese Kategorie ausgewählt und verhindert, dass der Vorgang abgebrochen und wiederholt werden muss.

3.4.4 Suchen in Listen

Das oben beschriebene Tastaturkürzel ist nützlich, um direkt zu einem einzelnen Element zu gelangen. Gelegentlich ist es aber erforderlich, alle Elemente herauszusuchen, deren Namen eine bestimmte Zeichenfolge enthalten. Dies erreichen Sie mit dem Befehl „Find Item“ (Element suchen), der in der Symbolleiste des Navigationsfensters angezeigt wird. Mit diesem Befehl wird die aktuelle Liste durchsucht, und alle übereinstimmenden Elemente werden in der globalen Suchergebnisliste angezeigt. Mit der Taste F4 und der Tastenkombination UMSCHALT+F4 können Sie diese Liste durchsehen. Mit der Taste F8 wird die ganze Liste angezeigt. Die globalen Suchfunktionen werden in diesem Kapitel noch ausführlicher erörtert.

3.4.5 Private Elemente

Tags und bestimmte andere Elemente in Navigationslisten können mit dem Attribut „Private Access“ (Privater Zugriff) gekennzeichnet werden. Klicken Sie dazu mit der rechten Maustaste und wählen Sie im Kontextmenü das entsprechende Element aus. Derartige Elemente können normal bearbeitet und gehandhabt werden, bis der Datenbankschutz auf „Private Access“ (Privater Zugriff) gesetzt wird. Danach ist der Inhalt nur sichtbar, wenn das Datenbankkennwort angegeben wurde. Die Elemente sind zwar weiterhin vorhanden und können auf die übliche Weise referenziert werden, sie können aber nicht bearbeitet oder angezeigt werden. Dieser Mechanismus soll OEMs ermöglichen, ihren Kunden Datenbanken bereitzustellen, ohne dabei wichtiges geistiges Eigentum preiszugeben oder die Integrität von Schlüsselementen zu gefährden. Weitere Informationen finden Sie im Abschnitt „Datenbankschutz“ weiter unten.

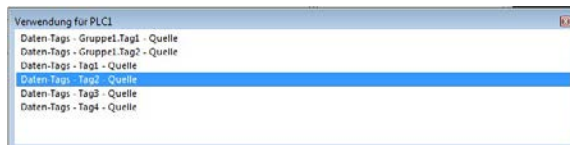
3.5 Rückgängig machen und Wiederholen

Crimson 3.1 implementiert eine universelle Rückgängig- und Wiederherstellen-Struktur. Dies bedeutet, dass Sie eine Datenbank laden, stundenlang bearbeiten und dann ganz einfach in den ursprünglichen Zustand zurückversetzen können, indem Sie die Tastenkombination STRG+Z gedrückt halten. Anschließend können Sie Ihre Änderungen erneut anwenden, indem Sie STRG+Y drücken. Alle Ihre Aktionen werden gespeichert, und Crimson navigiert automatisch zwischen Elementen und Kategorien, wenn Änderungen rückgängig gemacht oder neu angewendet werden.

3.6 Globale Suche

Crimson 3.1 bietet verschiedene Optionen für die Suche in einer Datenbank. Die einfachste Option ist die Suche nach einer Textzeichenfolge in der Datenbank mithilfe der Tastenkombination **STRG+UMSCHALT+F**. Wie später erörtert, können Sie alternativ auch nach Ausdrücken suchen, die Fehler enthalten, oder nach Elementen, die auf ein Tag oder ein Kommunikationsgerät verweisen. Bei allen Suchvorgängen wird die Ausgabe in der globalen Suchergebnisliste angezeigt. Sie können die Ergebnisse prüfen oder zwischen den gefundenen Elementen hin und her navigieren.

Die Ergebnisliste kann jederzeit durch Drücken der Taste **F8** angezeigt werden.



Die Titelleiste des Fensters enthält eine Beschreibung des Suchvorgangs, durch den die Liste entstanden ist. Die einzelnen Zeilen wiederum enthalten die Beschreibung eines Elements, das mit den Suchkriterien übereinstimmt. Im obigen Beispiel wurde mit der rechten Maustaste auf ein Kommunikationsgerät geklickt und der Befehl „Find Usage“ (Verwendung suchen) gewählt. Daraufhin wurden alle Stellen aufgeführt, an denen das Gerät referenziert wurde. Durch einen Doppelklick auf einen bestimmten Eintrag gelangen Sie direkt zu diesem Element. Mit der Taste **F4** und der Tastenkombination **UMSCHALT+F4** können Sie die Liste vorwärts und rückwärts durchsehen. Die mit dieser Funktion verknüpften Befehle können auch im Menü „Edit“ (Bearbeiten) über die Befehle „Find Global“ (Globale Suche) aufgerufen werden.

3.7 Arbeiten mit Datenbanken

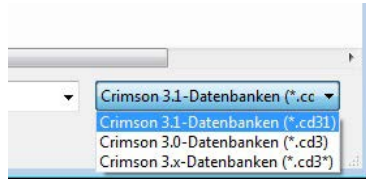
Crimson 3.1 speichert alle Informationen über die Konfiguration eines Geräts in einer Datenbankdatei. Diese Dateien tragen die Erweiterung `cd31`. In Windows Explorer wird diese Erweiterung allerdings ausgeblendet, wenn Sie ihn in der Standardkonfiguration belassen. Crimson 3.1-Datenbanken sind zwar im Prinzip Textdateien, doch sie sind komprimiert und in einigen Fällen verschlüsselt und können daher nicht direkt mit einem Texteditor wie Notepad bearbeitet werden. Datenbanken werden über die Standardbefehle im Menü „File“ (Datei) bearbeitet.

3.7.1 Importieren aus Crimson 2.0

Der Import aus Crimson 2.0 wird von Crimson 3.1 nicht direkt unterstützt. Wenn Sie eine solche Datenbank konvertieren müssen, wenden Sie sich an den technischen Support von Red Lion. Die Support-Mitarbeiter können die Datenbank für Sie verarbeiten. Alternativ können Sie die Datenbank in Crimson 3.0 importieren und die resultierende Datei wie unten beschrieben in Crimson 3.1 konvertieren.

3.7.2 Importieren aus Crimson 3.0

Datenbanken, die in Crimson 3.0 erstellt wurden, können in Crimson 3.1 direkt geöffnet werden. Dazu müssen Sie lediglich unten rechts im Dialogfeld „File Open“ (Datei öffnen) einen anderen Dateityp auswählen. Wenn Sie die Datenbank speichern möchten, stellt Crimson fest, dass die Datei im Crimson 3.1-Format gespeichert werden muss. Daraufhin wird ein Dialogfeld mit der Frage angezeigt, wo Sie die Datei speichern möchten und unter welchem Namen. Der neue Name ist in der Regel mit dem alten Namen identisch; allerdings wird die Erweiterung `cd3` durch `cd31` ersetzt. Crimson 3.0-Datenbanken können auch über den Befehl „File Import“ (Datei importieren) geöffnet werden. Dieser Befehl führt im Wesentlichen den gleichen Vorgang aus, aber es ist nicht erforderlich, den ausgewählten Dateityp zu ändern.



Wenn Sie eine Crimson 3.0-Datenbank öffnen oder importieren, die für ein Produkt konzipiert wurde, das nicht von Crimson 3.1 unterstützt wird, werden Sie gefragt, zu welchem Gerät Sie die Datei konvertieren möchten. Je nachdem, welche Produkte am besten dazu in der Lage sind, die Bildschirmauflösung des Originalgeräts zu unterstützen, stehen Ihnen verschiedene Optionen zur Auswahl.



Wenn die Bildschirmauflösung des Zielgeräts nicht mit dem des ursprünglichen Geräts übereinstimmt, werden möglicherweise mehrere Optionen zur Konvertierung der Anzeigeseiten angeboten. Optionen, die als *Scaled* (Skaliert) gekennzeichnet sind, passen den Inhalt der einzelnen Anzeigeseiten an das neue Bildschirmformat an und passen dabei nach Möglichkeit die Schriftgrößen an. Optionen, die als *Centered* (Zentriert) gekennzeichnet sind, lassen den Inhalt der Anzeigeseiten unverändert, zentrieren ihn aber in der neuen Bildschirmgröße, sodass entweder vertikale Balken zu sehen sind oder der Inhalt von einem Rahmen umgeben ist. Optionen, die als *Emulated* (Emuliert) gekennzeichnet sind, schalten den Bildschirm des Zielgeräts in einen Modus um, in dem die Bildschirmausgabe skaliert wird, um die Auflösung des Originalgeräts nachzuahmen. Wenn die Skalierung anhand eines Integralfaktors erfolgt, ergibt dies in der Regel eine klare Anzeige, jedoch mit größeren und auffälligeren Pixeln. Wenn ein nichtintegraler Faktor verwendet wird, sollten Sie die Ergebnisse in Augenschein nehmen, um festzustellen, ob die resultierende Anzeigequalität akzeptabel ist.

3.7.3 Datenbankennungen

Jeder von Crimson 3.1 erstellten Datenbank wird eine eindeutige Kennung zugewiesen. Diese Kennung wird beim Herunterladen einer neuen Datenbank verwendet, um festzustellen, ob das Zielgerät seinen internen Speicher leeren und alle auf der Speicherkarte des Geräts aufgezeichneten Protokolldateien löschen muss. Wenn die Kennung mit der bereits im Gerät vorhandenen Datenbank übereinstimmt, wird davon ausgegangen, dass die Datenbank lediglich eine andere Version derselben Datei ist. Die Daten werden daher beibehalten. Wenn die Kennungen jedoch unterschiedlich sind, werden die Daten gelöscht. Wenn Sie im Menü „File“ (Datei) den Befehl „Save as“ (Speichern unter) verwenden, um eine Kopie einer Datenbankdatei zu speichern, werden Sie von Crimson gefragt, ob Sie eine neue Kennung zuweisen möchten. Wählen Sie „Yes“ (Ja), wenn es sich um ein neues Projekt handelt. Wählen Sie „No“ (Nein), wenn Sie lediglich eine Sicherungskopie einer im Wesentlichen identischen Datenbank speichern. Dadurch wird sichergestellt, dass die remanenten Daten des Zielgeräts richtig verarbeitet werden.

3.7.4 Speichern eines Abbilds

Im Menü „File“ (Datei) befindet sich der Crimson-spezifische Befehl „Save Image“ (Abbild speichern). Mit diesem Befehl wird eine Datei erstellt, die zur Aktualisierung der Datenbank in einem Terminal über eine Speicherkarte oder einen USB-Speicherstick verwendet werden kann. Die Datei enthält eine nicht bearbeitbare Form der Datenbank sowie alle für die Ausführung erforderlichen Firmware- und Bootloader-Updates. Wenn Sie eine Abbilddatei namens `image.ci3` im Stammverzeichnis der Speicherkarte des Zielgeräts platzieren und das Gerät dann zurücksetzen, werden der Bootloader, die Firmware und die Datenbank anhand des Inhalts der Abbilddatei aktualisiert. Beachten Sie, dass Abbilddateien optional Upload-Informationen enthalten können und es dadurch ermöglichen, eine bearbeitbare Version der Datenbankdatei von einem Terminal zu extrahieren.

3.7.5 Datenbankschutz

Datenbanken können mithilfe des Befehls „Protection“ (Schutz) im Menü „File“ (Datei) durch ein Kennwort geschützt werden:



Mit dem Parameter „Standardzugriff“ wird festgelegt, inwieweit ein Zugriff ohne vorherige Eingabe des Datenbankkennworts zulässig ist. Verschiedene Einstellungen sind verfügbar:

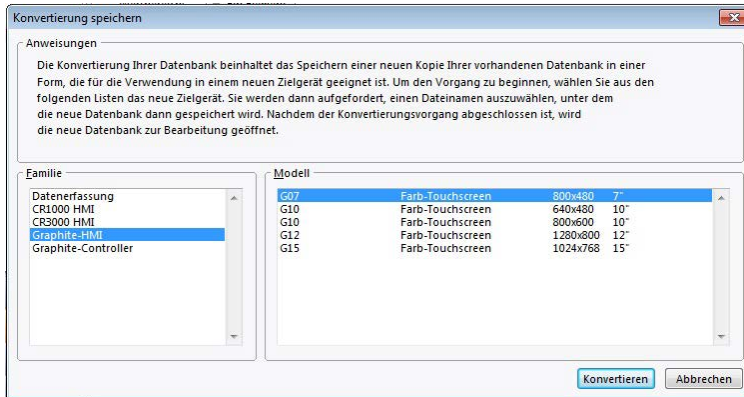
- Bei der Einstellung *Full Access* (Voller Zugriff) kann die Datenbank ganz ohne Kennworteingabe geöffnet und bearbeitet werden. Dies ist die Standardeinstellung.
- Bei der Einstellung *Read-Only Access* (Schreibgeschützter Zugriff) kann die Datenbank zwar geöffnet werden, aber es ist nicht möglich, Änderungen vorzunehmen oder zu speichern.
- Bei der Einstellung *Private Access* (Privater Zugriff) kann die Datenbank zwar geöffnet werden, aber es ist nicht möglich, Elemente mit der Kennzeichnung „Private Access“ (Privater Zugriff) anzuzeigen oder zu bearbeiten.
- Bei der Einstellung *Download Only Access* (Nur Download-Zugriff) kann die Datenbank zwar geöffnet werden, aber es ist nicht möglich, Elemente anzuzeigen oder zu bearbeiten. Downloads werden weiterhin unterstützt.
- Bei der Einstellung *No Access* (Kein Zugriff) ist der Zugriff ausschließlich mit dem Kennwort möglich.

Verlorene Kennwörter können von Red Lion kostenlos wiederhergestellt werden – oder gegen eine geringe Gebühr, wenn Sie diese Möglichkeit wiederholt in Anspruch nehmen. Beachten Sie, dass das wiederhergestellte Kennwort aus Sicherheitsgründen nur an die im Dialogfeld „Protection“ (Schutz) konfigurierte Wiederherstellungs-E-Mail gesendet wird. Achten Sie darauf, hier eine gültige E-Mail-Adresse anzugeben, weil es Ihnen andernfalls nicht möglich ist, Ihr Kennwort wiederherzustellen.

Es sei darauf hingewiesen, dass dieser Datenbankschutzmechanismus nur dazu gedacht ist, die beschriebenen Zugriffsebenen für vertrauenswürdige Benutzer anzubieten. Es ist nicht beabsichtigt, 100% kryptographisch sicher zu sein und bietet keinen Schutz vor böswilligen Angriffen. Es wird empfohlen, beim Speichern oder Übertragen Ihrer Datenbank eine sichere Schutzmethode eines Drittanbieters zu verwenden.

3.7.6 Konvertieren einer Datenbank

Eine Datenbank, die für ein bestimmtes Zielgerät entworfen wurde, kann mit dem Befehl „Save Conversion“ (Konvertierung speichern) im Menü „File“ (Datei) für die Verwendung auf einem anderen Gerät konvertiert werden. Welche Konvertierungen durchgeführt werden können, hängt vom ursprünglichen Zielgerät ab. Grundsätzlich werden aber die meisten Kombinationen unterstützt.

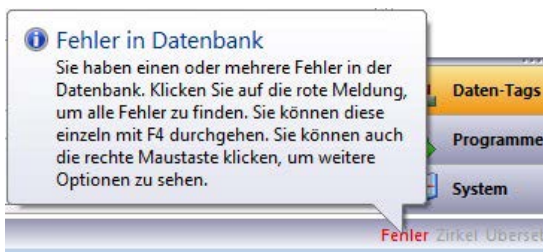


Der Konvertierungsprozess beginnt mit der Auswahl des neuen Zielgeräts im oben abgebildeten Dialogfeld. Sie werden dann aufgefordert, einen neuen Dateinamen einzugeben. Die konvertierte Datenbank wird anschließend auf der Festplatte gespeichert. Um die versehentliche Vernichtung vorhandener Datenbanken zu vermeiden, können Sie eine Datenbank erst konvertieren, nachdem Sie sie unter einem neuen Namen gespeichert haben. Nachdem die konvertierte Datenbank gespeichert wurde, wird sie automatisch zur Bearbeitung und Überprüfung geöffnet.

Beim Konvertierungsprozess wird die Größe aller Anzeigeseiten an das neue Bildschirmformat angepasst. Außerdem werden Kommunikationsgeräte auf Basis der verwendeten Bitübertragungsschicht (RS-232 oder RS-485) den entsprechenden Anschlüssen am neuen Gerät neu zugeordnet. Möglicherweise kann eine Datenbank nicht vollständig konvertiert werden, z. B. wenn das neue Gerät über weniger Kommunikationsanschlüsse verfügt als das Original. Daher müssen Sie nach der Konvertierung möglicherweise einige Anpassungen vornehmen.

3.7.7 Suchen nach Datenbankfehlern

Bestimmte Vorgänge können zu Fehlern in der Datenbank führen. Dies geschieht beispielsweise, wenn Sie ein Kommunikationsgerät löschen oder wenn Sie ein Tag einem Ausdruck gleichsetzen, der auf sich selbst basiert, wodurch Sie einen Zirkelbezug erzeugen. In Crimson 3.1 werden Sie bei derartigen Fehlern durch eine rote Sprechblase gewarnt, die über der Statusleiste eingeblendet wird:



Die Sprechblase wird nach einigen Sekunden wieder ausgeblendet, aber der rote Hinweis verbleibt in der Statusleiste und erinnert Sie weiterhin an die Fehlerbedingung. Wenn Sie auf den Hinweis klicken, wird nach allen Fehlern oder Zirkelbezügen gesucht und diese werden in der globalen Suchergebnisliste aufgeführt. Mit der Taste **F4** und der Tastenkombination **UMSCHALT+F4** können Sie die Fehlermeldungen überprüfen. Sie können auch mit der rechten Maustaste auf den Hinweis klicken, um auf Befehle zuzugreifen, mit denen die gesamte Datenbank neu kompiliert oder die Organisation der Gerätekommunikation optimiert wird. Eine manuelle Neukompilierung

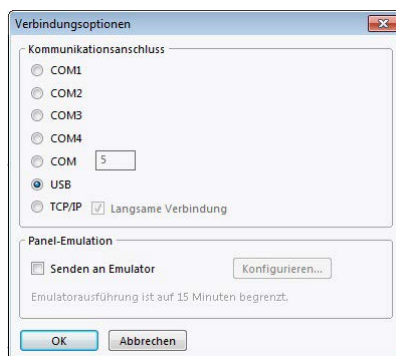
einer Datenbank ist selten erforderlich, da Crimson 3.1 die erforderlichen Schritte in der Regel ohne Benutzereingriff durchführt.

3.8 Download auf ein Gerät

Crimson 3.1-Datenbankdateien werden über das Menü „Link“ auf das Zielgerät heruntergeladen. Der Download dauert in der Regel nur wenige Sekunden. Er kann allerdings beim ersten Download etwas länger dauern, wenn Crimson die Firmware auf dem Gerät aktualisieren muss oder wenn das Gerät keine ältere Version der aktuellen Datenbank enthält. Nach diesem ersten Download nutzt Crimson einen Prozess, der als „inkrementeller Download“ bezeichnet wird und sicherstellt, dass nur Änderungen an der Datenbank übertragen werden. Updates können somit in wenigen Sekunden durchgeführt werden. Für Sie bedeutet dies kürzere Entwicklungszyklen und eine Vereinfachung des Debugging-Prozesses.

3.8.1 Konfigurieren der Verbindung

Die Programmierverbindung zwischen dem PC und dem Zielgerät kann über einen RS-232-Anschluss, einen USB-Anschluss oder eine TCP/IP-Verbindung hergestellt werden. TCP/IP-Verbindungen werden zwar in der Regel über den Ethernet-Anschluss der Konsole hergestellt, können aber auch über eine Wählverbindung aufgebaut werden. Bevor Sie mit dem Herunterladen beginnen, vergewissern Sie sich mithilfe des Befehls „Link-Options“ (Verbindungsoptionen), dass Sie die richtige Methode ausgewählt haben:



Beachten Sie, dass es in diesem Dialogfeld keine Möglichkeit gibt, die Ziel-IP-Adresse auszuwählen, wenn TCP/IP für den Download verwendet wird. Diese Informationen werden in der Datenbankdatei gespeichert und über die Registerkarte „Download“ des Elements „Netzwerkconfiguration“ konfiguriert. Diese Methode macht es einfacher, zwischen mehreren Datenbanken zu wechseln, ohne jedes Mal die Ziel-IP neu konfigurieren zu müssen. Beachten Sie auch, dass Crimson 3.1 bei der Arbeit mit mehreren Produktfamilien unterschiedliche Download-Einstellungen speichert. Es ist dann leichter, USB für den Download auf USB-fähige Produkte zu verwenden und bei weniger leistungsfähigen Geräten auf serielle Downloads zurückzugreifen.

3.8.2 Senden der Datenbank

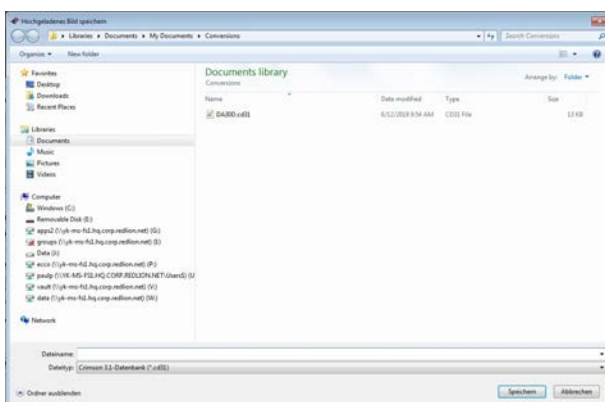
Sobald die Verbindung konfiguriert ist, kann die Datenbank mit dem Befehl „Link-Send“ (Verknüpfung – Senden) oder „Link-Update“ (Verknüpfung – Aktualisieren) heruntergeladen werden. Mit „Link-Send“ (Verknüpfung – Senden) wird die gesamte Datenbank gesendet, unabhängig davon, ob einzelne Objekte innerhalb der Datei geändert wurden. Mit „Link-Update“ (Verknüpfung – Aktualisieren) werden nur Änderungen gesendet. Dieser Vorgang nimmt in der Regel sehr viel weniger Zeit in Anspruch. Normalerweise benötigen Sie nur den Aktualisierungsbefehl, weil Crimson 3.1 automatisch die vollständige Datenbank sendet, wenn der inkrementelle Download aus irgendeinem Grund fehlschlägt. Der Befehl „Link-Update“ (Verknüpfung – Aktualisieren) kann auch schnell über das Blitzsymbol in der Symbolleiste oder über die Taste F9 auf der Tastatur aufgerufen werden.



Beachten Sie, dass der Download über TCP/IP auf einige Modelle vorsieht, dass eine Speicherkarte installiert ist, wenn für die Firmware des Geräts ein Upgrade durchgeführt werden soll. Da Sie derartige Upgrades irgendwann durchführen werden wollen, empfiehlt es sich, in jedem Gerät eine Speicherkarte zu installieren, auf das wahrscheinlich Downloads über TCP/IP durchgeführt werden. Beachten Sie auch, dass die Downloadoption TCP/IP über die Netzwerkeinstellungen in der Kategorie „Kommunikationen“ aktiviert werden muss.

3.8.3 Extrahieren von Datenbanken

Mit dem Befehl „Link-Support Upload“ können die zur Unterstützung von Datenbank-Uploads erforderlichen Informationen beim Senden einer Datenbank an ein Zielgerät eingeschlossen werden. Diese Einstellung wird in der Datenbank gespeichert und kann für jede Datei einzeln konfiguriert werden. Durch Aktivieren des Datenbank-Uploads wird der Download ein wenig verlangsamt, und bei extrem großen Datenbanken, die viele eingebettete Bilder enthalten, kann der Vorgang fehlschlagen. Der Vorteil ist jedoch, dass es Ihnen beim Verlust Ihrer Datenbankdatei möglich ist, ein bearbeitbares Abbild aus dem Gerät zu extrahieren.



Falls Sie Ihre Datenbankdatei verlieren und der Upload-Support nicht aktiviert ist, können Sie Ihre Datei nur rekonstruieren, indem Sie sie von Grund auf neu erstellen. Zum Extrahieren einer Datenbank aus einer Konsole verwenden Sie den Befehl „Link-Extract“ (Verknüpfung – Entpacken). Mit diesem Befehl wird die Datenbank hochgeladen, und Sie werden aufgefordert, einen Namen einzugeben, unter dem die Datei gespeichert werden soll. Die Datei wird dann zur Bearbeitung geöffnet. Wenn die Datenbank durch ein Kennwort geschützt war, müssen Sie möglicherweise das Kennwort eingeben, damit sie geöffnet werden kann. Anders ausgedrückt: Der Kennwortschutz lässt sich nicht durch die Aktivierung des Uploads umgehen.

3.8.4 Senden von Datum und Uhrzeit

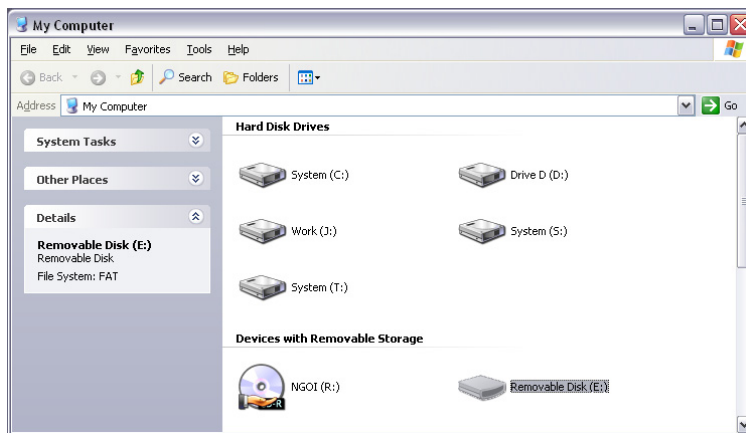
Mit dem Befehl „Link-Send Time“ (Verknüpfung – Sende Uhrzeit) können Sie die Uhr des Zielgeräts mit der Uhr des Computers synchronisieren, auf dem Crimson 3.1 ausgeführt wird. Mit diesem Befehl werden auch die aktuellen Einstellungen für Zeitzone und Sommerzeit an das Zielgerät gesendet, sodass die erweiterten Funktionen des Zeitmanagers verwendet werden können. Beachten Sie, dass bestimmte Funktionen nur zuverlässig funktionieren, wenn die Uhr richtig eingestellt ist. Dies gilt insbesondere für Funktionen im Zusammenhang mit SSL-TLS-Sicherheit.

3.9 Die Speicherkarte

Wenn Ihr Zielgerät über eine Speicherkarte verfügt, stehen verschiedene zusätzliche Funktionen zur Verfügung.

3.9.1 Anmelden der Speicherkarte

Wenn Sie über den USB-Anschluss mit einem geeigneten Gerät verbunden sind, können Sie Crimson 3.1 anweisen, die Speicherkarte des Geräts als Laufwerk in Windows Explorer anzumelden. Mit dieser Funktionalität können Sie Dateien auf der Karte speichern oder Informationen aus dem Datenlogger auslesen. Das Laufwerk wird durch Senden von Befehlen mit den Optionen „Mount Flash“ (Binde Flash ein) und „Dismount Flash“ (Entferne Flash) im Menü „Link“ angemeldet bzw. entfernt. Sobald ein Befehl gesendet wurde, wird das Zielgerät zurückgesetzt, und Windows aktualisiert die entsprechenden Explorer-Fenster.



Beim Anmelden der Speicherkarte ist Folgendes zu beachten:

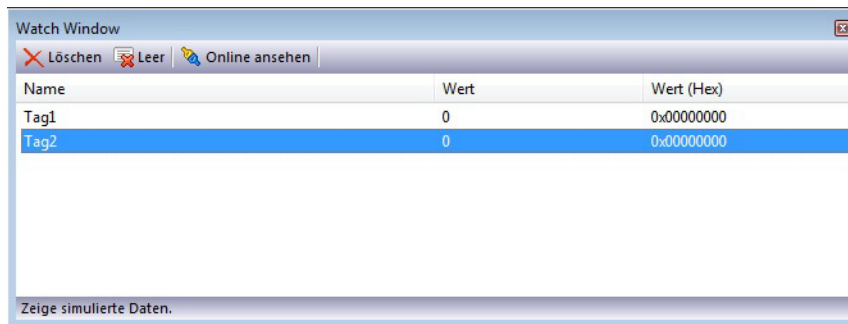
- Wenn die Karte angemeldet ist, informiert das Zielgerät den PC regelmäßig, ob Daten auf der Karte geändert wurden. Dies bedeutet, dass sowohl der PC als auch das Gerät kleine Leistungseinbußen verzeichnen, wenn die Karte während der Datenprotokollierung länger als nötig angemeldet ist.
- Wenn Sie von Ihrem PC auf die Speicherkarte schreiben, kann das Zielgerät nicht auf die Karte zugreifen, bis Windows die Sperre für die Karte aufhebt. Dies kann einige Sekunden dauern. Während dieser Zeit ist die Datenprotokollierung eingeschränkt und der Zugriff auf benutzerdefinierte Webseiten wird verhindert. Crimson 3.1 verwendet den Arbeitsspeicher des Geräts, um sicherzustellen, dass keine Daten verloren gehen. Wenn aber zu viele Schreibvorgänge durchgeführt werden und die Karte dadurch vier Minuten oder länger gesperrt ist, werden möglicherweise Daten verworfen.
- Sie sollten niemals versuchen, eine über Crimson 3.1 angemeldete Speicherkarte mit Windows zu formatieren, gleichgültig, ob Sie dafür den Explorer oder die Eingabeaufforderung verwenden. Die Karte wird während der Formatierung in Windows nicht richtig gesperrt, und das resultierende Format kann unzuverlässig sein und zu Datenverlusten führen. Im Folgenden erfahren Sie, wie Sie eine Karte auf zuverlässige Weise formatieren.

3.9.2 Formatieren der Karte

Es wird nur eine Methode zum Formatieren von Karten unterstützt, nämlich der Befehl „Format Flash“ (Formatiere Flash) im Menü „Link“. Bei Auswahl dieses Befehls wird Ihnen mitgeteilt, dass durch die Formatierung alle auf der Speicherkarte gespeicherten Daten vernichtet werden. Sie haben die Möglichkeit, den Vorgang abzubrechen. Wenn Sie den Vorgang fortsetzen, wird die Bedienkonsole angewiesen, die Karte zu formatieren. Beachten Sie, dass dieser Vorgang bei einer großen Karte einige Minuten dauern kann. Eine langsame Formatierung auf Bedienkonsolen, die eine Datenprotokollierung durchführen, kann daher dazu führen, dass die aufgezeichneten Daten lückenhaft sind.

3.10 Fernüberwachung

Crimson 3.1 unterstützt eine so genannte Beobachtungsliste, mit der Sie den Inhalt der Tags und Zuordnungsblöcke in der Datenbank anzeigen können. Die Beobachtungsliste wird im Beobachtungsfenster angezeigt. Dieses Fenster kann mit der Taste F7 oder dem Befehl im Menü „View“ (Ansicht) ein- oder ausgeblendet werden.



Wenn das Beobachtungsfenster zum ersten Mal angezeigt wird, enthält es die simulierte Daten, die beim Erstellen der Tags definiert wurden. Durch einen Klick auf die Schaltfläche „View Online“ (Online anzeigen) wird sichergestellt, dass die aktuelle Datenbank mit der Datenbank auf dem Zielgerät übereinstimmt. Dann beginnt die Anzeige von Live-Daten. Die Tag-Daten werden gemäß dem entsprechenden Formatobjekt angezeigt.

Sie können der Beobachtungsliste Elemente hinzufügen, indem Sie mit der rechten Maustaste klicken und im Menü den entsprechenden Befehl auswählen. Sie können ein oder mehrere Tags auf einmal hinzufügen, ebenso wie den Inhalt eines Zuordnungsblocks. Ferner gibt es einen Befehl, mit dem Sie alle Tags hinzufügen können, auf die eine bestimmte Anzeigeseite verweist. Dies erleichtert das Debugging der Seite, an der Sie gerade arbeiten. Mithilfe der Schaltflächen oben im Beobachtungsfenster können ein oder alle Elemente aus der Beobachtungsliste entfernt werden.

3.11 Systemmenü

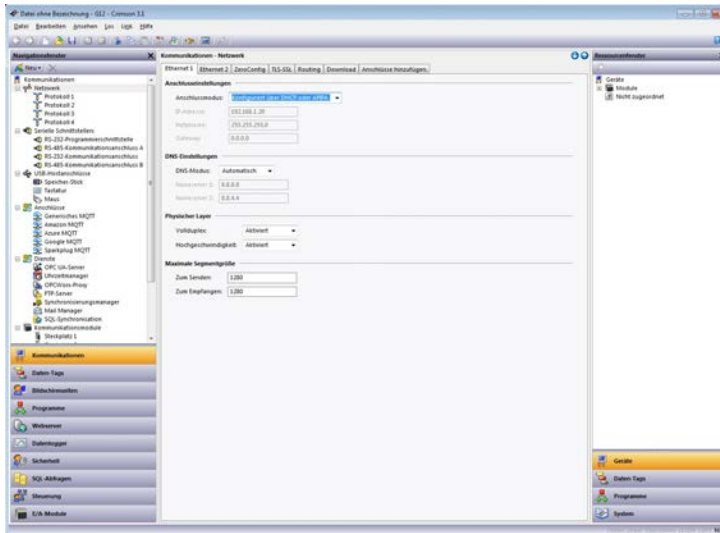
Die Crimson 3.1-Laufzeit wurde so erweitert, dass HMIs, die Crimson ausführen, jetzt einen Systemmodus unterstützen, mit dem verschiedene Wartungsfunktionen durchgeführt werden können. Dazu gehören:

- Löschen der Datenbank
- Laden der Datenbank von der Speicherkarte
- Kalibrieren oder Testen des Touchscreens
- Formatieren der Speicherkarte
- Formatieren des USB-Sticks
- Einstellen der Echtzeituhr

Sie greifen auf das Systemmenü zu, wenn Sie beim Starten des Geräts die kleine, versenkte Taste an der Rückseite des Geräts drücken. Halten Sie die Taste so lange gedrückt, bis das Menü angezeigt wird. Im Systemmenü werden die Standardwerte für die Touchscreen-Kalibrierung verwendet, damit die Wiederherstellung auch dann durchgeführt werden kann, wenn die Kalibrierung vollkommen falsch ist. Um auch bei falsch kalibriertem Touchscreen den korrekten Betrieb zu gewährleisten, erfordert die Auswahl einer Option besonders viel Umsicht. Bei der Berührung des Bildschirms wird die ausgewählte Taste rot dargestellt. Wenn das Gerät die richtige Taste erkannt hat, berühren Sie die Taste, bis sie rot ist. Wenn Sie dann loslassen, wird die zugehörige Option ausgewählt. Wenn das System die falsche Taste erkannt hat, lassen Sie sie sofort los, um die Auswahl abzubrechen und es erneut zu versuchen. Eine Beschreibung der Systemmenüfunktionen finden Sie im Technischen Hinweis TNO146, der von der Red Lion-Webseite heruntergeladen werden kann.

Kapitel 4 Verwenden von Kommunikationen

Die erste Phase beim Erstellen einer Crimson® 3.1-Datenbank besteht darin, die Kommunikationsports des Zielgeräts zu konfigurieren und anzugeben, welche Protokolle Sie verwenden und auf welche externen Geräte Sie zugreifen möchten. Diese Vorgänge werden in der Kategorie „Kommunikationen“ durchgeführt.



Wie Sie sehen, werden die verfügbaren Ports des Geräts in der Kategorie „Kommunikationen“ in Form einer Baumstruktur aufgeführt. Im obigen Beispiel gibt es zwei Ethernet-Ports, die über den Abschnitt „Netzwerk“ konfiguriert sind, und zu Beginn werden vier TCP/IP-Protokolle unterstützt. Außerdem gibt es drei primäre serielle Ports, und es besteht die Möglichkeit, weitere Ports in Form von Erweiterungsmodulen hinzuzufügen.

4.1 Netzwerkconfiguration

Die IP-Netzwerkconfiguration des Zielgeräts wird über das Symbol „Netzwerk“ im Navigationsfenster bearbeitet. Wenn das Symbol ausgewählt ist, werden im Bearbeitungsfenster mehrere Registerkarten angezeigt, über die jeweils bestimmte Eigenschaftssätze konfiguriert werden können.

4.1.1 Zero Config

Die erste Registerkarte dient zum Konfigurieren der neuen Netzwerkunterstützung „Zero Config“ von Crimson 3.1:



Mit „Zero Config“ kann ein Gerät über DHCP eine IP-Adresse abrufen und dann anhand eines einfachen Namens von überall im lokalen Subnetz referenziert werden. Wenn Sie dem Gerät beispielsweise den Namen `test` geben und bei den restlichen Einstellungen von Crimson die Standardwerte beibehalten, ruft das Gerät eine IP-Adresse ab und reagiert dann von jedem Ort im lokalen Subnetz auf Namensauflösungsanforderungen für den Namen

test.local. Sie können eine Eingabeaufforderung auf dem PC öffnen und `ping test.local` eingeben, um die IP-Adresse des Geräts zu bestätigen. Sie können auch den Webbrowser aufrufen und dort in der Adressleiste `test.local` eingeben, um die Standard-Webseite des Geräts zu besuchen. Beachten Sie hierbei, dass in Chrome und möglicherweise auch in einigen anderen Browsern das Präfix `http://` verwendet werden muss.

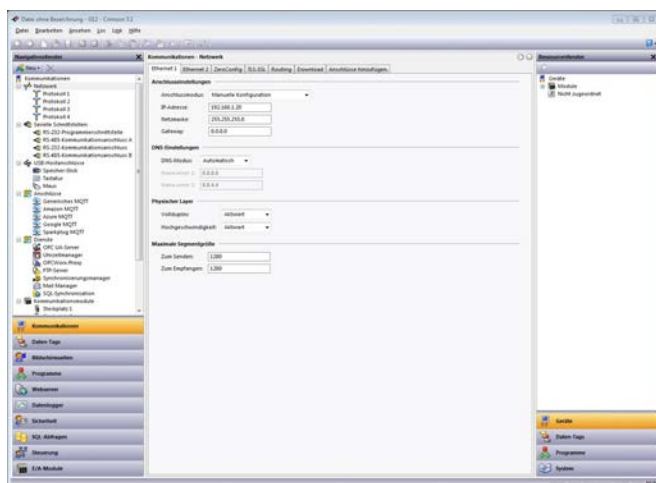
„Zero Config“ wird über die folgenden Eigenschaften gesteuert:

- Mit der Eigenschaft *Local Name* (Lokaler Name) wird der Name festgelegt, anhand dessen das Gerät referenziert wird. Der Name muss im lokalen Subnetz eindeutig sein. Wenn Sie keinen Namen eingeben, reagiert das Gerät auf `red-xx-yy-zz.local`, wobei jedes Buchstabenpaar durch Ziffern aus der letzten Hälfte der MAC-Adresse ersetzt wird. Beispielsweise reagiert ein Gerät mit der MAC-Adresse 00-05-E4-05-39-62 auf `red-05-39-62.local`.
- Mit der Eigenschaft *Responders* (Antwortsender) werden die lokalen Namensauflösungsprotokolle festgelegt, auf die das Gerät reagiert. Sie sollten idealerweise alle verfügbaren Protokolle aktiviert lassen, um sicherzustellen, dass Clients das Gerät finden können.
- Die Verknüpfung *Install Crimson Local Root* (Lokalen Stamm installieren) dient zur Installation eines Stammzertifikats. Dieses Zertifikat überprüft das SSL-Standardzertifikat, das von Crimson 3.1 automatisch für seinen Webserver generiert wird, wenn dieser im sicheren Modus ausgeführt wird. Wenn dieses Stammzertifikat installiert ist, werden alle von Crimson erzeugten Zertifikate vom PC als vertrauenswürdig eingestuft, sofern die Namen auf `.local` enden. Weitere Informationen finden Sie im Kapitel über den Webserver.

Beachten Sie, dass „Zero Config“ nur in einem einzelnen Subnetz funktioniert. Ein Gerät mit der IP-Adresse 192.168.1.200 und der Netzmaske 255.255.255.0 befindet sich im Subnetz 192.168.1.0. Geräte in anderen Subnetzen können das Gerät nicht über seinen lokalen Namen erreichen. Beachten Sie außerdem, dass die lokale Namensauflösung nicht für die Bereitstellung in Produktionsumgebungen empfohlen wird.

4.1.2 Ethernet-Einstellungen

Auf den nächsten ein oder zwei Registerkarten werden die Ethernet-Ports des Zielgeräts konfiguriert:



4.1.2.1 Anschlüsseinstellungen

Das Feld „Port Mode“ (Portmodus) steuert, ob der Port aktiviert ist, und mit welcher Methode der Port seine IP-Konfiguration abrufen soll. Wenn der DHCP-Modus ausgewählt ist, versucht das Zielgerät, eine IP-Adresse und zugehörige Parameter von einem DHCP-Server im Netzwerk abzurufen. Wenn DHCP fehlschlägt, wird mit APIPA automatisch eine IP-Adresse zugewiesen.

Wenn das Gerät dem Netzwerk Dienste anbietet, ist DHCP nur dann sinnvoll, wenn „Zero Config“ für den benannten Zugriff aus dem lokalen Subnetz verwendet wird oder wenn der DHCP-Server so konfiguriert ist, dass

der MAC-Adresse des Geräts eine wohl bekannte IP-Adresse zugewiesen wird. Bei Bedienerkonsolen besteht auch die Möglichkeit, mit der Funktion `GetNetIP()` die IP-Adresse des Geräts auf dem Bildschirm anzuzeigen, Verweise direkt über diese Adresse zuzulassen oder die Erstellung eines HOSTS-Dateieintrags zu gestatten.

Wenn der Modus „Manual Configuration“ (Manuelle Konfiguration) ausgewählt ist, müssen die Felder „IP Address“ (IP-Adresse), „Network Mask“ (Netzwerkmaske) und „Gateway“ mit entsprechenden Informationen ausgefüllt werden. Die für diese Felder angebotenen Standardwerte dürften so gut wie nie für Ihre Anwendung geeignet sein. Fragen Sie den Netzwerkadministrator, wie die richtigen Werte lauten. Achten Sie darauf, diese Werte einzugeben und herunterzuladen, bevor Sie das Zielgerät mit Ihrem Netzwerk verbinden. Andernfalls könnten Sie Probleme im Netzwerk verursachen.

Wenn Sie den Modus „IEEE 802.3 Only“ (Nur IEEE 802.3) auswählen, wird der Port für Low-Level-Kommunikation aktiviert. Es wird jedoch keine IP-Adresse zugewiesen und TCP oder UDP können nicht verwendet werden. Dieser Modus ist nur sinnvoll, wenn Treiber verwendet werden, die Raw Ethernet nutzen, etwa bestimmte Protokolle für die Gebäudeautomatisierung.

4.1.2.2 DNS-Einstellungen

Unter „DNS Settings“ (DNS-Einstellungen) wird die Fähigkeit des Geräts konfiguriert, Hostnamen aufzulösen. Crimson 3.1 arbeitet zwar überwiegend mit IP-Adressen, aber einige Konfigurationsoptionen können stattdessen auf Hostnamen eingestellt werden. Um diese Hostnamen in IP-Adressen umzuwandeln, ist der Zugriff auf einen oder mehrere DNS-Server erforderlich. Die Server können manuell angegeben oder vom DHCP-Server abgerufen werden. Die standardmäßige manuelle Auswahl sind die öffentlichen DNS-Server von Google unter „8.8.8.8“ und „8.8.4.4“.

4.1.2.3 Bitübertragungsschicht

Die Optionen unter „Physical Layer“ (Bitübertragungsschicht) steuern die Art der Verbindung, die das Gerät mit dem Hub oder Switch, mit dem es verbunden ist, auszuhandeln versucht. Im Allgemeinen können Sie hier die Standardeinstellungen übernehmen. Wenn aber keine zuverlässige Verbindung zustande kommt, insbesondere bei direkten Verbindungen mit einem PC ohne zwischengeschaltetem Hub oder Switch, sollten Sie erwägen, sowohl den „Full Duplex“ (Voll duplex) als auch den „High Speed“ (Hochgeschwindigkeits-)Betrieb zu deaktivieren. Hierdurch lässt sich das Problem möglicherweise beheben.

4.1.2.4 Maximale Segmentgröße

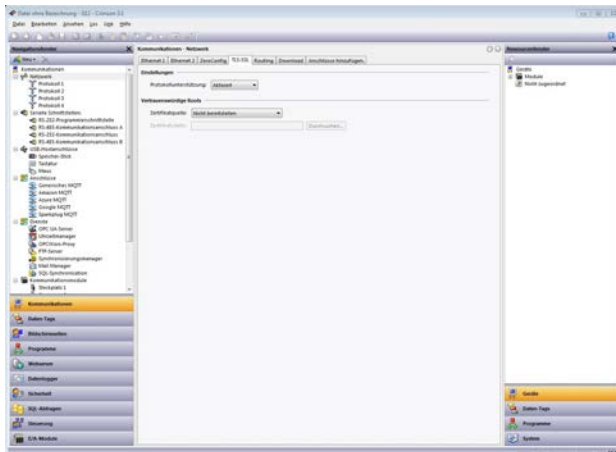
Die Optionen unter „Maximum Segment Size“ (Maximale Segmentgröße) steuern die MSS-Einstellungen für TCP Send and Receive (Senden und Empfangen). Diese Einstellungen müssen in der Regel nicht geändert werden, da die Standardwerte für praktisch alle Anwendungen und Netzwerke geeignet sind.

4.1.3 Mehrere Ports

Wenn Sie mehrere Ethernet-Ports verwenden, sollte nur für einen Port ein Standard-Gateway definiert sein, und jeder Port sollte über eine eindeutige Netzwerkadresse verfügen. Crimson 3.1 sendet ein bestimmtes IP-Paket nur an eine einzige Schnittstelle. In einer Konfiguration also, bei der z. B. der erste Ethernet-Port als 192.168.100.1 definiert ist und der zweite als 192.168.100.2, werden alle Pakete für das Netzwerk 192.168.100.0 an den ersten Port gesendet. Dadurch wird verhindert, dass der zweite Port ordnungsgemäß funktioniert.

4.1.4 TLS-SSL-Einstellungen

Auf der nächsten Registerkarte wird die Unterstützung für die TLS- und SSL-Protokolle konfiguriert, die für sichere Verbindungen über TCP/IP-Verbindungen verwendet werden.



Diese Protokolle sind erforderlich, wenn Sie beabsichtigen, HTTPS mit dem Webserver von Crimson zu verwenden oder auf sichere Mailserver zuzugreifen.

4.1.4.1 Einstellungen

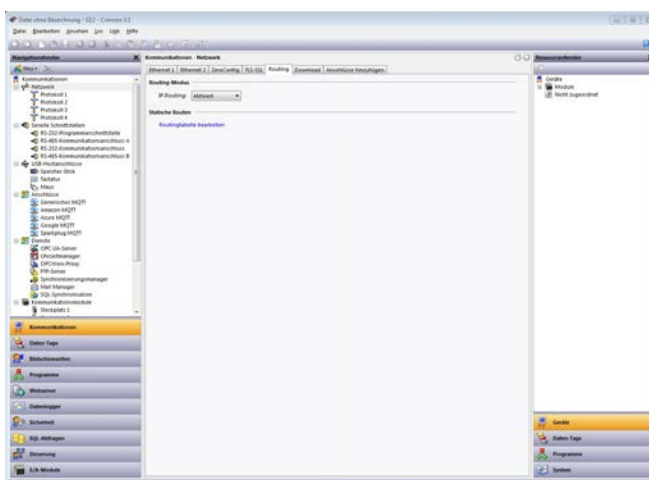
Die Option „Protocol Support“ (Protokollunterstützung) dient zur Aktivierung oder Deaktivierung der SSL-Unterstützung. Das Protokoll ist standardmäßig aktiviert, und es gibt kaum Gründe, es zu deaktivieren.

4.1.4.2 Vertrauenswürdige Roots

Wenn Crimson über TLS-SSL eine Verbindung mit einem Remote-Server herstellt, kann es optional die Identität des Servers validieren, indem es das Zertifikat des Servers anhand einer Sammlung vertrauenswürdiger Stammzertifikate prüft. Wenn Sie vorhaben, die Zertifikatüberprüfung zu verwenden, müssen Sie diese Stammzertifikate bereitstellen. Unter „Certificate Source“ (Zertifikatquelle) können Sie wählen, wie diese Zertifikate angegeben werden sollen. Sie können entweder eine Datei angeben, die ein oder mehrere Zertifikate enthält, oder Crimson anweisen, die derzeit auf Ihrem PC installierten vertrauenswürdigen Roots zu verwenden. Diese letzte Option bietet eine schnelle Methode zur Bereitstellung eines validierten Satzes von Stammzertifikaten, die in den meisten Situationen funktionieren.

4.1.5 Routing-Einstellungen

Auf der nächsten Registerkarte werden die TCP/IP-Routingoptionen konfiguriert, wie in der Abbildung gezeigt.

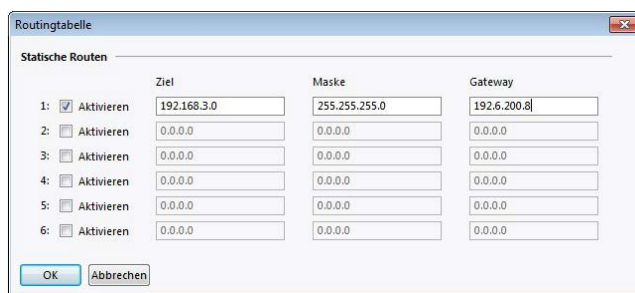


4.1.5.1 Routing-Modus

Die Option „IP Routing“ dient zum Aktivieren oder Deaktivieren der Paketweiterleitung zwischen Schnittstellen. Wenn diese Option aktiviert ist, werden IP-Pakete, die an einem Ethernet- oder Modemanschluss empfangen werden und für Geräte vorgesehen sind, die mit einem anderen Port verbunden sind, bei Bedarf weitergeleitet. Wenn Sie diese Option deaktivieren, wird die Weiterleitung verhindert. Welche Einstellung erforderlich ist, hängt von der Topologie Ihres Netzwerks ab.

4.1.5.2 Routingtabelle

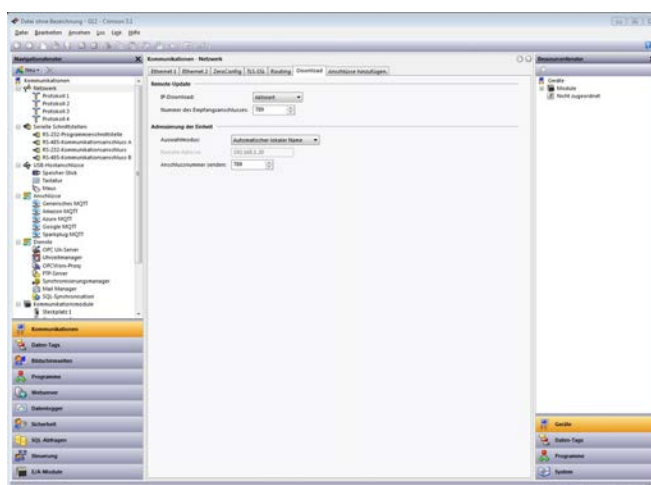
Die Routingtabelle definiert zusätzliche statische Routen für den TCP/IP Stack von Crimson 3.1.



Im obigen Beispiel wurde eine einzelne Route angegeben, die Crimson anweist, alle Pakete für IP-Adressen ab 192.168.3 an den Router im lokalen Netzwerk unter der Adresse 192.9.200.8 weiterzuleiten. Auch hier gilt, dass die genauen Einstellungen, die erforderlich sind, von der Topologie des Netzwerks abhängen, mit dem das Zielgerät verbunden ist.

4.1.6 Download-Einstellungen

Die nächste Registerkarte dient zum Konfigurieren von Downloads auf das Zielgerät über TCP/IP.



4.1.6.1 Remote Update

Mit der Option „IP Download“ können Sie TCP/IP-Downloads aktivieren oder deaktivieren. Mit der Option „Port Number“ (Portnummer) geben Sie an, welcher TCP-Port für derartige Downloads verwendet werden soll. Verwenden Sie den Standardwert 789, es sei denn, es gibt einen guten Grund, einen anderen Wert zu verwenden. Mit der Option „Download to Different Port“ (Download an anderen Port) wird Crimson 3.1 darüber informiert, dass

das Zielgerät zwar den unter „Port Number“ (Portnummer) angegebenen Port überwacht, die Daten aber an einen anderen Port gesendet werden sollen, weil ein zwischengeschalteter Router eine Portzuordnung durchführt.

4.1.6.2 Einheitenadressierung

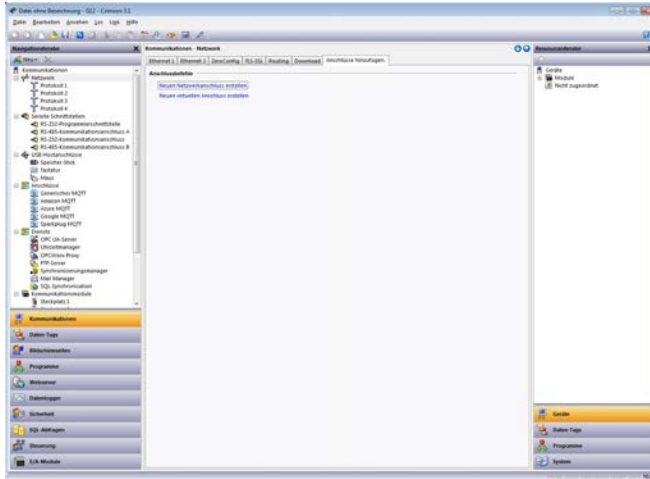
Mit diesen Einstellungen wird die IP-Adresse angegeben, die von der Crimson 3.1 Konfigurationssoftware verwendet werden soll, wenn im Dialogfeld „Link-Options“ (Link-Optionen) die Downloadmethode TCP ausgewählt ist:

- Im Modus „Auto Local Name“ (Autom. lokaler Name) wird das Gerät mit dem auf der Registerkarte „Zero Config“ definierten Namen referenziert. Dies funktioniert nur einwandfrei, wenn ein Name eingegeben wurde und Sie mit demselben Subnetz verbunden sind wie das Gerät.
- Im Modus „Auto Ethernet“ wird die IP-Adresse verwendet, die für den ausgewählten Ethernet-Port konfiguriert wurde. Selbstverständlich muss der Port hierfür manuell konfiguriert sein, damit dies Sinn ergibt, sonst liegen der Software im Vorhinein keine Informationen über die IP-Adresse vor.
- Im Modus „Manual“ (Manuell) kann eine IP-Adresse in das Feld „Remote Address“ (Remoteadresse) eingegeben werden.

Beachten Sie, dass diese Informationen als Teil der Datenbank gespeichert werden. Damit können Sie problemlos zwischen Geräten im selben Netzwerk wechseln, wenn Sie zwischen Datenbanken navigieren oder mehrere Crimson 3.1 Instanzen gleichzeitig geöffnet haben.

4.1.7 Hinzufügen von Ports

Die letzte Registerkarte dient zum Hinzufügen zusätzlicher Netzwerkprotokolle.



Mithilfe der Verknüpfung „Create New Network Port“ (Neuen Netzwerkanschluss erstellen) wird ein weiteres Netzwerkprotokoll hinzugefügt, bis die maximale Anzahl der vom Zielgerät unterstützten Ports erreicht ist. Mithilfe der Verknüpfung „Create New Virtual Port“ (Neuen virtuellen Port erstellen) wird ein ähnlicher Vorgang durchgeführt, doch dabei wird ein Port hinzugefügt, der eine serielle Verbindung über TCP/IP emulieren kann. Beide Porttypen können im Navigationsfenster gelöscht werden. Wählen Sie dazu den gewünschten Port aus, und drücken Sie ALT+ENTF oder klicken Sie in der Symbolleiste auf das Symbol „Löschen“.

4.1.8 Protokollauswahl

Sobald das Netzwerk konfiguriert wurde, können Sie die Protokolle auswählen, die Sie für die Kommunikation verwenden möchten. Verwenden Sie dazu die unten beschriebene Methode. Es können mehrere Protokolle gleichzeitig verwendet werden, und viele dieser Protokolle unterstützen mehrere externe Geräte. Das bedeutet,

dass Sie verschiedene Möglichkeiten haben, Protokolle und Geräte zu kombinieren, um die gewünschten Ergebnisse zu erzielen.

Angenommen, Sie möchten mit Modbus über TCP/IP eine Verbindung zwischen zwei externen Slave-Geräten herstellen. Ihre erste Option besteht darin, zwei Netzwerkprotokolle zu verwenden und beide Protokolle als Modbus-Master mit jeweils einem angeschlossenen Gerät zu konfigurieren. Bei den meisten Protokollen wird dadurch eine höhere Leistung erzielt, da die gleichzeitige Kommunikation mit den beiden Geräten ermöglicht wird. Es werden jedoch zwei der verfügbaren Protokolle verbraucht, wodurch Ihre Möglichkeit eingeschränkt wird, in komplexen Anwendungen Verbindungen über zusätzliche Protokolle herzustellen.

Ihre zweite Option besteht darin, ein einzelnes Protokoll zu verwenden, das als Modbus TCP/IP-Master konfiguriert ist, aber ein weiteres Gerät hinzuzufügen, sodass über denselben Treiber auf beide Slaves zugegriffen wird. Hierdurch wird die Leistung in der Regel leicht reduziert, weil Crimson 3.1 jedes Gerät nacheinander abfragt und nicht mit beiden Geräten gleichzeitig kommunizieren kann. Es werden jedoch Netzwerkprotokolle geschont, sodass für komplexere Anwendungen genügend Ressourcen verfügbar sind.

4.1.9 Verwenden virtueller Ports

Wie oben erwähnt, unterstützt Crimson 3.1 das Hinzufügen virtueller Ports zur Netzwerkkonfiguration. Ein virtueller Port verwendet das Crimson Kommunikationssystem genau wie ein serieller Port, sendet und empfängt seine Daten aber über eine TCP/IP-Verbindung. Virtuelle Ports können im aktiven oder im passiven Modus konfiguriert werden. Im ersteren Fall versucht Crimson, eine TCP/IP-Verbindung mit einem bestimmten externen Gerät herzustellen. Im letzteren Fall überwacht Crimson einen bestimmten TCP/IP-Port auf eingehende Verbindungen. Virtuelle Ports werden in der Regel zur Kommunikation mit Geräten über serielle Remote-Server verwendet: Es wird zwar ein serielles Standardprotokoll eingesetzt, aber die Daten dieses Protokolls sind in TCP/IP-Paketen eingekapselt.

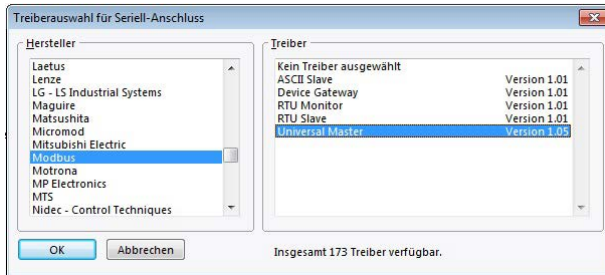
4.2 Auswahl eines seriellen Ports

Beachten Sie bei der Entscheidung, welche seriellen Ports des Zielgeräts für die Kommunikation verwendet werden sollen, dass einige Geräte und bestimmte Optionskarten einen einzelnen seriellen Controller zwischen mehreren Ports gleichzeitig übertragen. Wenn also einer der Ports für ein Slave-Protokoll verwendet wird, ist der andere Port nicht verfügbar. Gleichermaßen wird bei Verwendung eines Token-Passing-Protokolls wie DH-485 der andere Port ebenfalls deaktiviert. Bei dem Versuch, eine Konfiguration zu erstellen, die gegen diese Regeln verstößt, wird von Crimson 3.1 eine Warnung angezeigt.

Beachten Sie auch, dass der Programmieranschluss eines Zielgeräts zwar als zusätzlicher Kommunikationsanschluss eingesetzt werden kann, dass er jedoch unter diesen Umständen nicht für Downloads verfügbar ist. Dies ist kein Problem, sofern das Gerät auch über einen USB-Port verfügt. Die USB-basierte Methode wird dringend empfohlen, wenn Sie Geräte über den Programmieranschluss anschließen möchten. Wenn Sie den Programmieranschluss sowohl für Downloads als auch für die Kommunikation verwenden möchten, müssen Sie eine Methode bereitstellen, mit der serielle Downloads erneut aktiviert werden, indem der Befehl `StopSystem()` als Reaktion auf eine Benutzeraktion ausgeführt wird.

4.3 Auswählen eines Protokolls

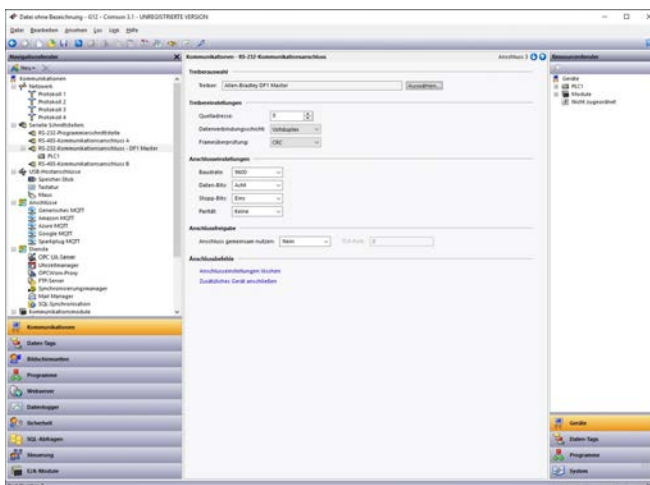
Um ein Protokoll für einen Port auszuwählen, klicken Sie im Navigationsfenster auf das Symbol des gewünschten Ports und anschließend im Bearbeitungsfenster neben dem Feld „Driver“ (Treiber) auf die Schaltfläche „Pick“ (Auswählen). Das folgende Dialogfeld wird angezeigt:



Wählen Sie den entsprechenden Hersteller und Treiber aus und klicken Sie auf „OK“, um das Dialogfeld zu schließen. Der Port wird dann mit dem entsprechenden Protokoll konfiguriert, und im Navigationsfenster wird ein Gerätesymbol erstellt. Wenn Sie einen seriellen Port konfigurieren, werden die verschiedenen Felder für die Porteinstellungen (Baudrate, Datenbits, Stoppbits und Parität) auf Standardwerte gesetzt, die dem betreffenden Protokoll entsprechen. Sie sollten diese Einstellungen überprüfen, um sicherzustellen, dass sie den Einstellungen für das zu adressierende Gerät entsprechen.

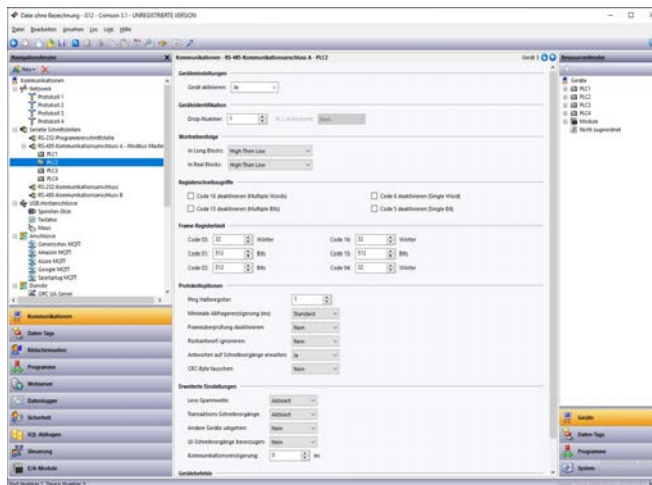
4.4 Protokolloptionen

Bei einigen Protokollen müssen weitere Parameter konfiguriert werden, die für dieses Protokoll spezifisch sind. Diese werden im Bearbeitungsfenster angezeigt, wenn das entsprechende Portsymbol ausgewählt wird. Im folgenden Beispiel sehen Sie die zusätzlichen Parameter für den Treiber „Allen-Bradley DF-1“, die im Bearbeitungsfenster im Bereich „Driver Settings“ (Treibereinstellungen) angezeigt werden:



4.5 Arbeiten mit Geräten

Wie oben erwähnt wird bei der Auswahl eines Kommunikationsprotokolls unter dem entsprechenden Portsymbol ein einzelnes Gerät erstellt. Im Fall eines Master-Protokolls stellt dies das erste externe Gerät dar, das über das Protokoll angesprochen wird. Wenn das Protokoll den Zugriff auf mehrere Geräte unterstützt, können Sie über die Schaltfläche „Add Additional Device“ (Zusätzliches Gerät hinzufügen) im Bearbeitungsfenster weitere Zielgeräte hinzuzufügen. Sie können auch den Befehl „New Comms Device“ (Neues Kommunikationsgerät) in der Symbolleiste des Navigationsfensters verwenden. Jedes Gerät wird im Navigationsfenster anhand eines Symbols dargestellt und hat je nach Protokoll eine Reihe von konfigurierbaren Eigenschaften.



Im obigen Beispiel ist das Modbus Universal Master-Protokoll ausgewählt. Es wurden drei zusätzliche Geräte erstellt, d. h. es soll auf insgesamt vier externe Geräte zugegriffen werden. Das Bearbeitungsfenster zeigt die Eigenschaften für die einzelnen Geräte an. Die Eigenschaft „Enable Device“ (Gerät aktivieren) ist für alle Geräte vorhanden. Die Einstellungen in den übrigen Feldern sind spezifisch für das jeweils ausgewählte Protokoll. Beachten Sie, dass neu erstellten Geräten in Crimson 3.1 Standardnamen zugewiesen werden. Sie können diese Namen ändern. Wählen Sie dazu im Navigationsfenster das entsprechende Symbol aus, drücken Sie die Taste **F2**, und geben Sie dann den neuen Gerätenamen ein.

4.6 Erweiterte Einstellungen

Zusätzlich zu den oben genannten Geräteeinstellungen bieten bestimmte Master-Geräte eine Reihe erweiterter Einstellungen, mit denen das Kommunikationsverhalten optimiert werden kann:

- Mit der Option *Spanning Reads* legen Sie fest, ob Crimson 3.1 Lesevorgänge durch Lesen von Datenblöcken optimiert, selbst wenn diese Blöcke Register umfassen, die sich derzeit nicht in der Kommunikationsprüfung befinden bzw. nicht in der Datenbank referenziert sind. Wenn beispielsweise „Spanning Reads“ für eine Datenbank aktiviert ist, welche auf die Register D1, D2 und D4 verweist, wird ein einzelner Kommunikationsbefehl ausgegeben, um von D1 an vier Register auszulesen. Wenn „Spanning Reads“ deaktiviert ist, werden zwei Lesevorgänge durchgeführt, und zwar einer für zwei Register ab D1 und einer für ein einzelnes Register ab D4.
- Mit den Optionen unter *Transactional Writes* (Transaktionale Schreibvorgänge) geben Sie an, wie nach einer Reihe von Änderungen an einem Datenwert in Crimson 3.1 zu verfahren ist: Entweder wird eine entsprechende Reihe von Schreibvorgängen ausgeführt oder nur der letzte geschriebene Wert wird übertragen. Transaktionale Schreibvorgänge erleichtern beispielsweise den Austausch von Tasten.
- Mit der Option *Preempt Other Devices* (Vorrang vor anderen Geräten) geben Sie an, ob Lesevorgänge von anderen Geräten unterbrochen werden sollen, damit Schreibvorgänge an dieses Gerät verarbeitet werden. Wenn diese Option aktiviert ist, verringert sich die Schreibleistung auf Kosten weniger vorhersehbarer Datenaktualisierungen von anderen Geräten.
- Mit der Option *Favor UI Writes* (UI-Schreibvorgänge bevorzugen) geben Sie an, ob Schreibvorgänge, die sich direkt aus Benutzeraktionen ergeben, Priorität haben sollen. Dies ist nützlich, wenn Sie mit einer Datenbank arbeiten, die aufgrund von Protokollkonvertierungen oder programmgesteuerten Aktivitäten viele Hintergrundkommunikationen ausführt.
- Mit der Option *Comms Delay* (Kommunikationsverzögerung) geben Sie eine Verzögerung an, die jeweils zwischen zwei Kommunikationstransaktionen für dieses Gerät eingefügt wird. Dies ist nützlich, wenn Sie mit externen Geräten arbeiten, die nicht mit der Leistung von Crimson 3.1 Schritt halten können, oder wenn einem Gerät eine geringere Kommunikationspriorität zugewiesen werden soll.

4.7 Erstellen von Tags

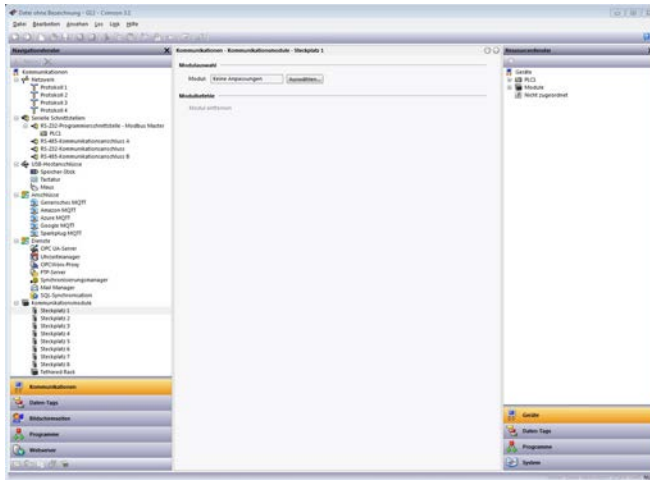
Einige Treiber bieten eine Option zum Erstellen von Tags in Crimson, die den Datenelementen entsprechen, die im externen Gerät vorhanden sind. Diese Option wird über den Link „Make Data Tags“ (Daten-Tags erstellen) auf der Gerätekonfigurationsseite aufgerufen. Der genaue Ablauf hängt vom jeweiligen Treiber ab, aber in der Regel werden Sie aufgefordert, eine Konfigurationsdatei auszuwählen, die von der Programmiersoftware des Geräts exportiert wurde. Beim Importvorgang werden alle Tags aus einem früheren Import desselben Geräts gelöscht, aber Tag-Einstellungen wie Formate, Auslöser, Sicherheit usw. bleiben erhalten.

4.8 Port- und Gerätenutzung

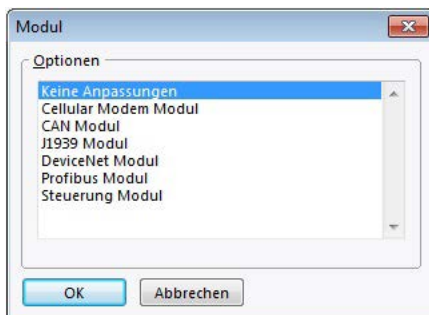
Wenn Sie eine Liste aller Elemente anzeigen möchten, die sich auf ein bestimmtes Kommunikationsgerät oder auf eines der mit einem bestimmten Port verbundenen Geräte beziehen, klicken Sie im Navigationsfenster mit der rechten Maustaste auf das Element, und wählen Sie den Befehl „Find Usage“ (Nutzung suchen) aus. Die gefundenen Elemente werden in der globalen Suchergebnisliste aufgeführt, und Sie können über die Taste **F4** bzw. die Tastenkombination **UMSCHALT+F4** darauf zugreifen. Die Liste selbst kann durch Drücken der Taste **F8** ein- oder ausgeblendet werden.

4.9 Verwenden von Erweiterungsmodulen

Einige Geräte unterstützen die Installation externer Kommunikationsmodule, um zusätzliche Kommunikationsmöglichkeiten bereitzustellen. Es stehen verschiedene Karten zur Verfügung, darunter Modelle, die Busprotokolle wie CANOpen, Profibus oder DeviceNet unterstützen. Diese Module können an das Gerät selbst oder über ein Erweiterungsrack angeschlossen werden. Geräte, die diese Module unterstützen, haben in der Navigationsliste ihres Abschnitts „Kommunikationen“ einen Abschnitt namens „Comms Modules“ (Kommunikationsmodule). Das Gerät unten unterstützt z. B. sechs direkt verbundene Module und die Option zum Hinzufügen von angebundnen Erweiterungs racks:



Um das Modul auszuwählen und es in einem bestimmten Steckplatz zu installieren, wählen Sie diesen Steckplatz aus. Klicken Sie dann im Abschnitt „Module Selection“ (Modulauswahl) neben der Eigenschaft „Module“ (Modul) auf die Schaltfläche „Pick“ (Auswählen). Ein Dialogfeld wie das unten abgebildete wird angezeigt:



Wenn Sie das entsprechende Modul auswählen, werden dem Navigationsfenster ein oder mehrere Symbole hinzugefügt. Diese Symbole stellen die zusätzlichen Ports dar, die von der Karte zur Verfügung gestellt werden. Die zusätzlichen Ports können wie in den vorherigen Abschnitten beschrieben konfiguriert werden. Beachten Sie, dass die für einen Port verfügbaren Treiber sich nach den unterstützten Verbindungstypen richten. Zum Beispiel zeigt die CANOpen-Erweiterungskarte einen Port an, der nur die für den CAN-Kommunikationsstandard konzipierten Treiber unterstützt.

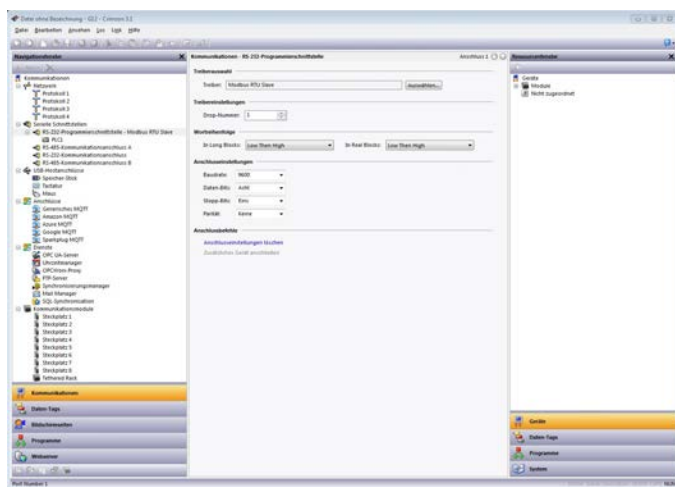
Wenn ein Erweiterungs rack an das Zielgerät angeschlossen ist, können Sie dieses konfigurieren, indem Sie in der Navigationsstruktur auf die Option „Tethered Rack“ (Angebundenes Rack) klicken und dann im Navigationsfenster links oben unter der Schaltfläche „New“ (Neu) die Option „Expansion Rack“ (Erweiterungs rack) auswählen. Daraufhin werden Symbole hinzugefügt, die das Rack und die davon bereitgestellten Steckplätze darstellen. Diese können genau wie die Steckplätze des Zielgeräts selbst konfiguriert werden.

4.10 Slave-Protokolle

Für Master-Protokolle (d. h. Protokolle, bei denen das Crimson-Gerät die Kommunikation initiiert) ist in der Kategorie „Kommunikationen“ keine weitere Konfiguration erforderlich. Für Slave-Protokolle (d. h. Protokolle, bei denen das Crimson-Gerät externe Anforderungen empfängt und darauf reagiert), ist der Vorgang ein wenig komplizierter, weil Sie auch angeben müssen, welche Daten Sie verfügbar machen möchten.

4.10.1 Auswählen des Protokolls

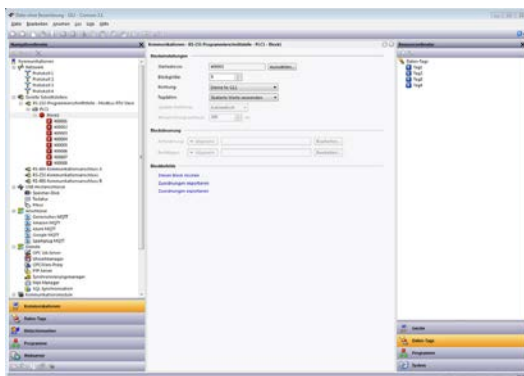
Genau wie bei den Master-Protokollen müssen Sie zuerst das Protokoll für den Kommunikationsanschluss auswählen, den Sie verwenden möchten. Im folgenden Beispiel ist der Port RS-485 des Zielgeräts für den Betrieb mit dem Slave-Protokoll Modbus ASCII konfiguriert:



Beachten Sie, dass ein einzelnes Gerät für das Protokoll erstellt wurde. Bei Master-Protokollen stellt dies das externe Gerät dar, auf das Crimson zugreifen wird. Im vorliegenden Fall jedoch stellt das Gerät den Modbus-Slave dar, den die Hardware selbst verkörpert. Dies bedeutet, dass nur ein einziges Gerät erforderlich ist und dass Elemente wie die Stationsnummer, auf die die Hardware reagieren wird, normalerweise über die Porteinstellungen konfiguriert werden und nicht über das Gerät.

4.10.2 Hinzufügen von Gateway-Blöcken

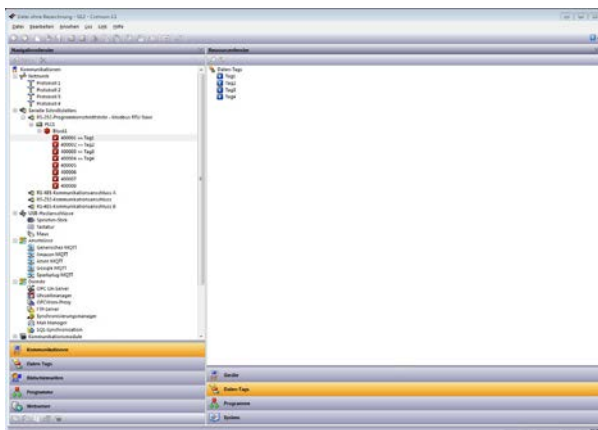
Nachdem Sie das Protokoll konfiguriert haben, müssen Sie entscheiden, welchen Adressbereich das Slave-Protokoll verfügbar machen soll. In diesem Beispiel möchten wir die Modbus-Register 40001 bis 40008 verwenden, um Lese- und Schreibzugriff auf bestimmte Datenelemente in unserer Datenbank zu ermöglichen. Zunächst klicken wir im Navigationsfenster auf das Gerätesymbol und dann im Bearbeitungsfenster auf die Schaltfläche „Add Gateway Block“ (Gateway-Block hinzufügen). Unter dem Gerät wird daraufhin ein Symbol für den Block angezeigt:



Im obigen Beispiel haben wir 40001 als Startadresse konfiguriert, um anzugeben, dass der Block hier beginnen soll. Außerdem haben wir die Blockgröße auf acht festgesetzt, um jedem Tag, das verfügbar gemacht werden soll, jeweils ein Modbus-Register zuzuordnen. Unter „Direction“ (Richtung) wurde „Device to Crimson“ (Gerät an Crimson) gewählt. Hierdurch wird externen Geräten ermöglicht, die über diesen Block verfügbar gemachten Datenelemente zu lesen und zu schreiben. Zum Schluss haben wir bei der Eigenschaft „Tag Data“ (Tag-Daten) die Standardeinstellung „Use Scaled Values“ (Skalierte Werte verwenden) übernommen. Dies bedeutet, dass alle Skalierungen auf Tag-Daten angewendet werden, bevor diese Daten an den Gateway-Block übertragen werden.

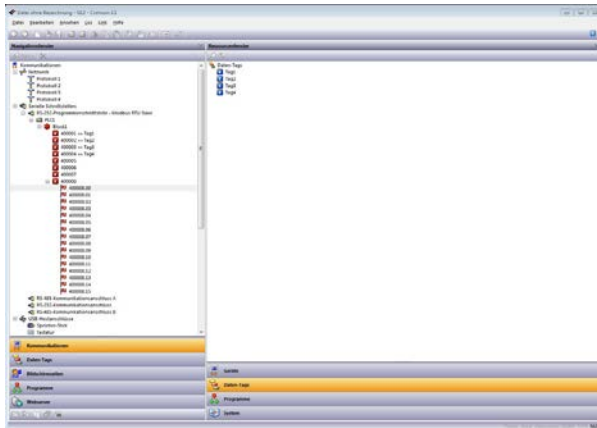
4.10.3 Hinzufügen von Elementen zu einem Block

Nachdem der Block erstellt und seine Größe definiert wurde, werden im Navigationsfenster Einträge für die einzelnen Register angezeigt, die der Block für den externen Zugriff verfügbar macht. Bei der Auswahl eines Eintrags wird ein erweitertes Ressourcenfenster angezeigt, in dem Sie Zugriff auf verfügbare Datenelemente haben. Diese Elemente umfassen sowohl Tags aus der Datenbank als auch Datenregister von beliebigen Master-Kommunikationsgeräten, die Sie konfiguriert haben.



Um anzugeben, dass ein Register in Ihrem Gateway-Block einem bestimmten Datenelement entsprechen soll, ziehen Sie dieses Element einfach aus dem Ressourcenfenster in das Navigationsfenster und legen es auf dem gewünschten Eintrag des Gateway-Blocks ab. Im obigen Beispiel wurden die ersten vier Register im Block den Tags namens Tag1 bis Tag4 zugeordnet. Damit wird angegeben, dass Zugriffe auf 40001 bis 40004 den jeweiligen Variablen zugeordnet werden sollen.

4.10.4 Zugriff auf einzelne Bits

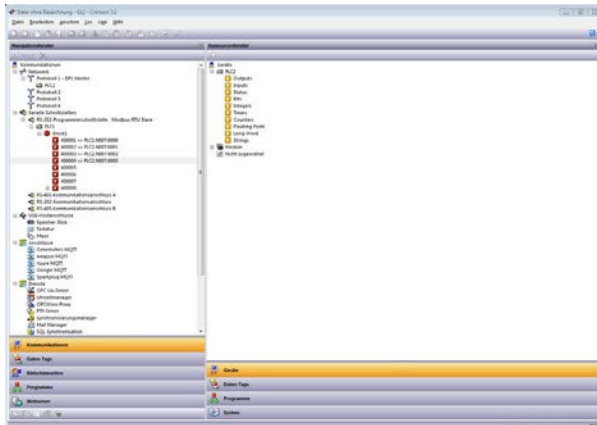


Falls für Ihre Anwendung erforderlich, können Sie einzelne Elemente in einem Gateway-Block bis auf ihre einzelnen Bits erweitern und dann jedem Bit ein anderes Datenelement zuordnen. Klicken Sie dazu mit der rechten Maustaste auf das betreffende Element, und wählen Sie im daraufhin angezeigten Menü die Option „Expand Bits“ (Bits erweitern) aus. Im Navigationsfenster werden nun die einzelnen Bits angezeigt, aus denen das Register besteht. Mit dem oben beschriebenen Drag-and-Drop-Vorgang können diese Bits zugeordnet werden.

4.11 Protokollkonvertierung

Gateway-Blöcke dienen nicht nur dazu, interne Daten-Tags über Slave-Protokolle verfügbar zu machen. Mit Gateway-Blöcken können auch Daten verfügbar gemacht werden, die von anderen Geräten abgerufen werden. Ferner ist es möglich, Daten zwischen zwei Master-Geräten zu verschieben. Diese einzigartige Protokollkonvertierungsfunktion ermöglicht eine deutlich engere Integration von Elementen eines Steuerungssystems, selbst wenn einfache, kostengünstige Geräte verwendet werden.

4.11.1 Master und Slave



Die Offenlegung von Daten von anderen Geräten über ein Slave-Protokoll ist einfach eine Erweiterung des oben beschriebenen Zuordnungsprozesses. Anstatt jedoch ein Tag aus dem Ressourcenfenster zu ziehen, wählen Sie die Kategorie „Comms Devices“ (Kommunikationsgeräte) aus, erweitern das entsprechende Master-Gerät, und ziehen das Symbol für die Register hinüber, die Sie verfügbar machen wollen. Sie werden dann aufgefordert, im Master-Gerät eine Startadresse sowie die Anzahl der zuzuordnenden Register anzugeben. Die Zuordnungen werden dann wie gezeigt erstellt.

In diesem Beispiel wurden die Register $N7:0$ bis $N7:3$ einer Allen-Bradley SPS als Register 40001 bis 40004 für den Zugriff über Modbus verfügbar gemacht. Crimson stellt automatisch sicher, dass diese Datenelemente aus dem Allen-Bradley SPS gelesen werden, um Modbus-Anforderungen zu erfüllen. Schreibvorgänge in die Modbus-Register werden automatisch in Schreibvorgänge in die SPS konvertiert. Wenn Sie ein TCP/IP-Slave-Protokoll auswählen, kann selbst das einfachste Legacy-Gerät mithilfe dieses Mechanismus mit einem Ethernet-Netzwerk verbunden werden.

4.11.2 Master und Master

Um Daten zwischen zwei Master-Geräten zu verschieben, wählen Sie einfach eines der Geräte aus und erstellen für dieses Gerät einen Gateway-Block. Anschließend können Sie Referenzen zu den Registern des anderen Geräts hinzufügen, genau wie bei der Freigabe von Daten mit einem Slave-Protokoll. Auch hier werden die Daten von Crimson bei Bedarf automatisch gelesen oder geschrieben sowie transparent zwischen den Geräten verschoben. Im obigen Beispiel sehen Sie, wie Daten von einem Modbus-Gerät in einen SLC-500 verschoben werden.

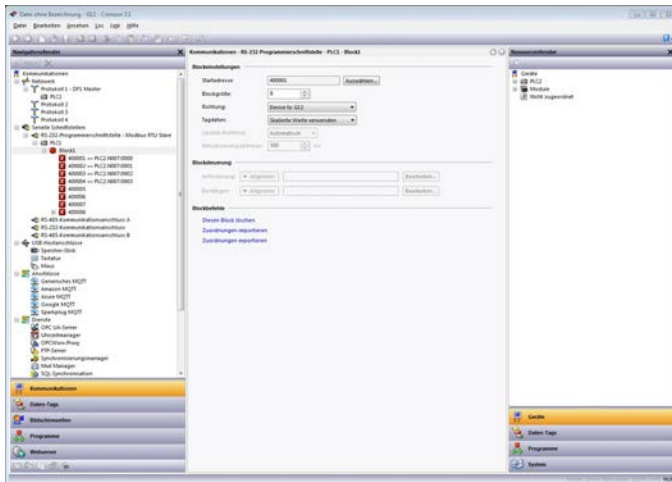
4.11.3 Die richtige Vorgehensweise

Sie mögen sich fragen, ob es besser ist, den Gateway-Block wie in diesem Beispiel im Allen-Bradley-Gerät zu erstellen oder doch lieber im Modbus-Gerät. Dazu zwei Anmerkungen: Erstens ist es unnötig, mehr als einen Block zu erstellen, wenn Übertragungen nur in eine Richtung durchgeführt werden sollen. Wenn Sie einen Block in AB erstellen, um Daten aus MOD zu lesen, und einen zweiten Block in MOD, um in AB zu schreiben, führen Sie ein und dieselbe Übertragung zweimal durch und beeinträchtigen hierdurch die Gesamtleistung. Zweitens ist die Entscheidung, auf welchem Gerät sich der Gateway-Block befinden soll, im Wesentlichen eine Frage des Ermessens. Im Allgemeinen sollten Sie beim Erstellen von Blöcken darauf achten, die Anzahl der Blöcke in der

Datenbank möglichst gering zu halten. Angenommen, die Register im Allen-Bradley-Gerät befinden sich alle in einem einzigen Bereich, doch die Register im Modbus-Gerät sind über die SPS verteilt. In diesem Fall sollte der Gateway-Block im Allen-Bradley-Gerät erstellt werden, denn dadurch entfällt die Notwendigkeit, mehrere Blöcke für den Zugriff auf die verschiedenen Bereiche des Modbus-Adressraums zu erstellen.

4.11.4 Steuern von Master-Blöcken

Gateway-Blöcke in Master-Geräten verfügen über mehrere zusätzliche Eigenschaften.



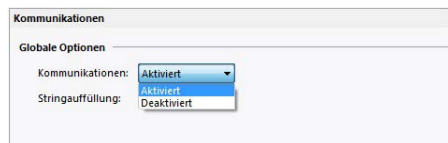
- Wie bei Slave-Blöcken wird auch hier mit der Option *Tag Data* (Tag-Daten) angegeben, wie die Zuordnung von Daten-Tags zum und vom Block erfolgen soll. Wie im nächsten Kapitel ausführlicher erläutert, können Tag-Daten verschiedene Transformationsphasen durchlaufen. Mit dieser Eigenschaft legen Sie fest, an welcher Stelle der Transformation der Gateway-Block Daten abrufen und einfügt.
- Mit der Eigenschaft *Update Policy* (Aktualisierungsrichtlinien) legen Sie fest, wie der Block aktualisiert wird. Wenn Sie die Standardeinstellung „Automatic“ (Automatisch) übernehmen, werden Leseblöcke kontinuierlich aktualisiert, und Schreibblöcke übertragen nur Werte, die geändert wurden. Bei Auswahl der Einstellung „Continuous“ (Fortlaufend) werden alle Blöcke kontinuierlich aktualisiert. Bei der Einstellung „Timed“ (Zeitlich festgelegt) werden alle Blöcke in dem Zeitintervall aktualisiert, das unter „Update Period“ (Aktualisierungszeitraum) angegeben ist. Bei der Aktualisierung wird jedes Mal der gesamte Inhalt eines Schreibblocks geschrieben.
- Mit den Eigenschaften *Request* (Anforderung) und *Acknowledge* (Bestätigen) können Sie den Zeitpunkt von Block-Aktualisierungen über Tags oder andere Datenelemente steuern. Wenn Sie die Eigenschaft „Acknowledge“ (Bestätigen) leer lassen, verhält sich „Request“ (Anforderung) wie ein Aktivierungsfeld. Mit dem Wert Null wird der Block dann deaktiviert und mit einem Wert von nicht Null bleibt er aktiv. Wenn Sie „Acknowledge“ (Bestätigen) definieren, funktionieren „Request“ (Anforderung) und „Acknowledge“ (Bestätigen) wie ein normaler Zwei-Wege-Handshake. Dabei wird der Block bei jeder steigenden Flanke der Anforderung jeweils einmal aktualisiert, und die Bestätigung wird nach Abschluss der Transaktion festgelegt.

4.12 Datentransformation

Mithilfe von Gateway-Blöcken können Sie auch Rechenoperationen ausführen, die eine SPS sonst möglicherweise nicht bewältigen kann. Angenommen, Sie möchten ein Register aus der SPS lesen, es skalieren, die Quadratwurzel berechnen und in ein anderes SPS-Register schreiben. Wie Sie hierzu vorgehen müssen, erfahren Sie im Abschnitt zu Daten-Tags. Erstellen Sie ein zugeordnetes Tag, das den Eingabewert darstellt, der aus dem Gerät gelesen wird. Erstellen Sie dann ein Tag, das den Ausgabewert darstellt, und stellen Sie den Ausdruck zur Durchführung der erforderlichen Rechenoperation ein. Sie können dann einen Gateway-Block erstellen, der auf das erforderliche Ausgaberegister ausgerichtet ist, und die Formel hinüberziehen, um Crimson anzuweisen, den abgeleiteten Wert wieder in die SPS zu schreiben.

4.13 Deaktivieren der Kommunikation

Crimson bietet die Möglichkeit, die gesamte treiberbasierte Kommunikation mit einer Eigenschaft in der obersten Ebene der Kategorie „Kommunikation“ zu deaktivieren.



Die Deaktivierung der Kommunikation kann während der Entwicklung nützlich sein, wenn Sie nicht über die erforderlichen Zielgeräte verfügen. Beim Betrieb im deaktivierten Modus setzt Crimson zunächst alle Tags gleich ihren simulierten Werten und gestattet dann die Änderung dieser Werte, als würden sie in die zugehörigen Geräte geschrieben. Wenn Sie feststellen, dass die Kommunikation grundlos unterbrochen wurde, prüfen Sie, ob Sie diese Einstellung deaktiviert haben.

Kapitel 5 Arbeiten mit Tags

Nachdem Sie die Kommunikationsoptionen für die Datenbank konfiguriert haben, besteht der nächste Schritt darin, die Datenelemente zu definieren, die Sie anzeigen oder anderweitig bearbeiten möchten. Hierzu wählen Sie im Navigationsfenster die Kategorie „Daten-Tags“ aus. Tags können anhand der in diesem Handbuch bereits erwähnten Standardverfahren erstellt, gelöscht und anderweitig bearbeitet werden.

5.1 Alles über Tags

Daten-Tags sind benannte Einheiten, die Datenelemente darstellen.

5.1.1 Datenquellen

Tags können Daten aus drei möglichen Quellen beziehen:

- Ein Tag kann einem oder mehreren Registern in einem externen Gerät *zugeordnet* werden. In diesem Fall wird das entsprechende Register von automatisch von Crimson® 3.1 gelesen, wenn das Tag referenziert oder angezeigt wird. Wenn Sie ein zugeordnetes Tag ändern, wird der neue Wert von Crimson automatisch in das Gerät geschrieben.
- Ein Tag kann *intern* sein. In diesem Fall stellt es ein oder mehrere Datenelemente im Crimson-basierten Gerät dar. Interne Tags können als remanent gekennzeichnet sein. In diesem Fall behalten sie ihre Werte während eines Aus- und Einschaltvorgangs bei. Als flüchtig gekennzeichnete interne Tags hingegen werden beim Einschalten des Geräts auf Null zurückgesetzt.
- Ein Tag kann ein *Ausdruck sein*. In diesem Fall stellt es eine Berechnung dar, die auf anderen Datenelementen basiert. Die Berechnung kann optional mathematische Operatoren und eine oder mehrere der internen oder benutzerdefinierten Funktionen von Crimson umfassen. Ausdrucks-Tags können abgeleitete Werte für den internen Gebrauch oder zur Übertragung an externe Geräte berechnen.

5.1.2 Typen von Tags

Crimson unterstützt drei Haupttypen von Tags:



















- *Numerische Tags* stellen Ganzzahlen oder Gleitkommawerte dar.
- *Markierungs-Tags* stellen einen Ein- oder Aus-Wert dar.
- *String-Tags* stellen Zeichenfolgen von Unicode-Zeichen dar.

Jeder der drei Haupttypen kann einen einzelnen Wert oder ein Array von Werten darstellen. Ein Array ist eine Sammlung von Elementen mit ähnlichen Eigenschaften, die gruppiert und über einen Indexwert aufgerufen werden. Zugeordnete Arrays entsprechen mehreren Registern im Zielgerät.

Ein vierter Tag-Typ ist das *Basis-Tag*. Dies ist eine vereinfachte Version eines Tags, die nur Zeichenfolgen oder numerische Ausdrücke darstellen kann. Diesem Tag fehlen viele der leistungsstarken Funktionen der Standard-Tags. Es wird in der Regel zum Codieren von einfachen Datenelementen wie Konstanten verwendet.

5.1.3 Tag-Tasten

In der folgenden Grafik sind die in Crimson 3.1 verfügbaren Tag-Tasten und deren Verwendung aufgeführt:

	Integer; Internal		When seen in gateway blocks; Read Block (Device to G3)
	Integer; Read Only		When seen in slave device gateway blocks; Read and Write Block
	Integer; Writeable (Read and Write or Write Only)		Binary Value; Internal
	Integer; Array		Binary Value; Read Only
	Integer; Basic Tag		Binary Value; Writeable (Read and Write or Write Only)
	String; Internal		Binary Value; Array
	String; Read Only		Binary Value; Standard Flag
	String; Writeable (Read and Write or Write Only)		
	String; Array		
	String; Basic Tag		
	Floating Point; Internal		
	Floating Point; Read Only		
	Floating Point; Writeable (Read and Write or Write Only)		
	Floating Point; Array		
	Floating Point; Basic Tag		
	When seen in gateway blocks; Write Block (G3 to Device)		

5.1.4 Tag-Attribute

Tags in Crimson sind reichhaltige Objekte, die verschiedene allgemeine Eigenschaften definieren:

- Die *Bezeichnung* eines Tags ist eine übersetzbare, von Menschen lesbare Zeichenfolge. Sie dient der automatischen Bezeichnung von Datenfeldern, die auf dieses Datenelement verweisen. Sie wird auch vom Webserver und vom Datenlogger verwendet, um zugehörige Datenelemente zu kennzeichnen.
- Die *Beschreibung* eines Tags ist eine nicht übersetzbare Zeichenfolge, mit der eine Erläuterung bezüglich des Zwecks des Tags bereitgestellt wird. Sie wird vom Benutzer des Zielgeräts normalerweise nicht eingesehen, kann aber zu Diagnosezwecken angezeigt werden.
- Das *Format* eines Tags ist eine Sammlung von Einstellungen, die die Methode definieren, mit der die Tag-Daten für die Anzeige bereitgestellt werden. Für das Format können Sie entweder die Einstellung „General“ (Allgemein) übernehmen, wobei Crimson dann die Standardformatierungsregeln verwendet, oder einen der vielen Formatierungstypen auswählen. Ein numerischer Wert kann z. B. im wissen-

schaftlichen Format angezeigt oder zur Auswahl einer Reihe verschiedener Textzeichenfolgen verwendet werden.

- Die *Farbgebung* eines Tags ist eine Sammlung von Einstellungen, mit denen definiert wird, wie der Text des Tags angezeigt werden soll oder welche Farben zur Darstellung des Zustands des Tags verwendet werden sollen. Auch hier steht eine Reihe verschiedener Optionen zur Auswahl, mit denen das Erscheinungsbild eines Tags in Abhängigkeit von einer Vielzahl von Bedingungen geändert werden kann. Vorder- und Hintergrundfarben werden paarweise definiert und können von Anzeigeprimitiven separat aufgerufen werden.
- Mit dem *Security Descriptor* (der Sicherheitsbeschreibung) wird definiert, welche Zugriffsregeln beim Ändern des Tags verwendet werden und ob diese Änderungen protokolliert werden sollen.

Bei Basis-Tags fehlen Format-, Farbgebungs- und Sicherheitsinformationen. Numerische Tags und Markierungs-Tags definieren ferner Alarme und Auslöser, damit unter bestimmten Bedingungen Alarme aktiviert oder bestimmte Aktionen ausgeführt werden können.

5.2 Vorteile von Tags

Da Crimson Ihnen ermöglicht, ein SPS-Register direkt auf einer Anzeigeseite zu platzieren, ohne Daten-Tags definieren zu müssen, soll an dieser Stelle kurz auf die Vorteile hingewiesen werden, die der mit dem Einsatz von Tags verbundene minimale zusätzliche Arbeitsaufwand mit sich bringt:

- Mit Tags können Sie Datenelemente benennen, damit Sie wissen, auf welche Datenelemente in der SPS sie sich beziehen. Wenn sich die Daten im SPS verschieben oder wenn Sie zu einer ganz anderen SPS-Familie wechseln, können Sie die Tags einfach neu zuordnen und vermeiden, weitere Änderungen an der Datenbank vornehmen zu müssen.
- Mit Tags können Sie vermeiden, die gleichen Informationen immer wieder erneut eingeben zu müssen. Wenn Sie ein Tag erstellen, geben Sie an, wie das Tag angezeigt werden soll. Bei numerischen Tags bedeutet dies, dass Sie Crimson anweisen, wie viele Dezimalstellen zu verwenden sind und welche Einheiten gegebenenfalls an den Wert angehängt werden sollen. Wenn Sie ein Tag auf einer Anzeigeseite platzieren, wird es von Crimson automatisch und ohne weiteren Benutzereingriff formatiert. Wenn Sie die Formatierung ändern und vielleicht zu einer anderen Einheitengruppe wechseln möchten, können Sie dies an einem zentralen Ort durchführen, ohne jede Anzeigeseite einzeln zu bearbeiten.
- Tags werden als einfache Methode für die Farbanimation verwendet. Mit den verschiedenen Farben, die für ein Tag definiert sind, kann angegeben werden, wie andere Animationsprimitive angezeigt werden. Es gibt zwar andere Methoden, doch Tags bieten eine einfache Möglichkeit, die Farbe von Anzeigeprimittiven zu ändern.
- Tags sind der Schlüssel zur Implementierung von Slave-Protokollen. Crimson behandelt diese Protokolle als Mechanismen zur Freigabe von Datenelementen im Zielgerät. Dadurch können dieselben Daten über mehrere Anschlüsse abgerufen werden. Somit kann eine Geräteeinstellung z. B. sowohl von einem lokalen SCADA-Paket als auch von einem ähnlichen Paket über Ethernet von einem externen Standort aus geändert werden. Ohne Tags gäbe es nichts zugänglich zu machen, und dieser Mechanismus könnte nicht implementiert werden.
- Tags werden in Crimson zur Implementierung vieler erweiterter Funktionen verwendet. Wenn Sie Funktionen wie Alarmer, Auslöser, Datenprotokollierung oder Webserver verwenden möchten, müssen Sie Tags einsetzen. Alle genannten Funktionen benötigen in der Regel die Formatierungsdaten aus der Tag-Definition, d. h. Tags sind für die Ausführung dieser Funktionen unverzichtbar.

Mit anderen Worten: Tags automatisieren viele Aufgaben, die bei der Programmierung anfallen, und hierdurch sparen Sie Zeit. Selbst wenn Sie keine Tags verwenden, beziehen sich viele der nachfolgenden Kapitel dieses Handbuchs auf Konzepte, die in diesem Kapitel erörtert werden. Lesen Sie sie gründlich durch, bevor Sie fortfahren.

5.3 Bearbeiten von Eigenschaften

Die meisten Eigenschaften lassen sich auf eine Weise bearbeiten, die erfahrenen Benutzern des Windows-Betriebssystems geläufig sein dürfte. Dazu gehören beispielsweise die Eingabe eines numerischen Werts oder die Auswahl eines Elements in einer Dropdown-Liste. Bestimmte Arten von Eigenschaften bieten jedoch komplexere Bearbeitungsoptionen, und diese werden im Folgenden beschrieben.

5.3.1 Ausdruckseigenschaften

Ausdruckseigenschaften können wie folgt festgelegt werden:

- als konstanter Wert
- als Inhalt eines Daten-Tags
- als Inhalt eines Registers in einem externen Kommunikationsgerät
- als Kombination dieser Elemente, die mit verschiedenen mathematischen Operatoren verknüpft sind
- als Rückgabewert eines lokalen Programms

Die Pfeilschaltfläche direkt neben der Eigenschaftsbezeichnung gibt in ihrem Standardzustand an, dass das Feld sich im Modus „General“ (Allgemein) befindet. Das Bearbeitungsfeld rechts neben der Schaltfläche enthält möglicherweise eine abgeblendete Zeichenfolge, die das Standardverhalten der Eigenschaft angibt. Die unten stehende Abbildung enthält ein Beispiel für eine leere Ausdruckseigenschaft ohne Standardwert:



Wenn Sie mit der Ausdruckssyntax von Crimson vertraut sind, können Sie die Eigenschaft bearbeiten, indem Sie einen Ausdruck direkt in das Bearbeitungsfeld eingeben. Eine vollständige Beschreibung dieser Syntax finden Sie im Kapitel „Schreiben von Ausdrücken“. Ausdruckseigenschaften können außerdem auf eine Vielzahl von weiteren Elementen eingestellt werden, die in der Regel über die Liste ausgewählt werden können, die unter der Pfeilschaltfläche angezeigt wird. Sie können diese Liste ganz normal aufrufen, indem Sie auf die Schaltfläche klicken. Alternativ können Sie bei ausgewählter Schaltfläche die Leertaste drücken, um das Menü einzublenden. Sie können auch die Taste mit dem ersten Buchstaben des Elements drücken, das Sie auswählen möchten, und so das Anzeigen der Liste umgehen. Zum Beispiel können Sie bei ausgewählter Schaltfläche die Taste **T** drücken, statt das Menü einzublenden und die Option „Tag“ auszuwählen.

5.3.1.1 Auswählen eines Tags

Es gibt vier Möglichkeiten, eine Ausdruckseigenschaft für ein vorhandenes Tag festzulegen. Erstens können Sie, nachdem Sie sich vergewissert haben, dass das Zielfeld ausgewählt ist, im Ressourcenfenster doppelt auf das gewünschte Tag klicken. Zweitens können Sie das Tag aus dem Ressourcenfenster ziehen und im Zielfeld ablegen. Drittens können Sie im Dropdown-Menü, welches durch die Pfeilschaltfläche aktiviert wird, die Option „Tag“ auswählen. Dabei fällt Ihnen womöglich auf, dass Sie das Ziel einfach auf das Feld hätten ziehen können. Viertens können Sie den Tag-Namen einfach in die Ausdruckseigenschaft eingeben.

5.3.1.2 Erstellen von Tags

Es gibt drei Möglichkeiten, eine Ausdruckseigenschaft für ein neues Tag festzulegen. Erstens können Sie für Ausdrücke, die die Quelle eines Datenelements definieren, im Dropdown-Menü, das durch die Pfeilschaltfläche aktiviert wird, die Option „New Tag“ (Neues Tag) auswählen. Zweitens können Sie, wenn Sie bereits ein Tag ausgewählt haben, auf die Schaltfläche „Pick“ (Auswählen) klicken und im daraufhin angezeigten Dialogfeld „New Tag“ (Neues Tag) auswählen. Drittens können Sie den Namen des neuen Tags als Teil eines Ausdrucks eingeben. Von Crimson wird dann ein Dialogfeld angezeigt, das in etwa der nachstehenden Abbildung entspricht:



In diesem Beispiel wurde ein Ausdruck mit einem Verweis auf Tag1 eingegeben, doch in der Datenbank ist kein derartiges Tag vorhanden. Der Fehler wird von Crimson erkannt, und Sie werden gefragt, ob Sie dieses Tag automatisch erstellen möchten. In der Dropdown-Liste können Sie den Typ des neuen Tags auswählen. Sie enthält geeignete Optionen für den Kontext, in dem das Tag verwendet wurde. Mit der Schaltfläche „Yes to All“ (Ja zu allem) können Sie Crimson anweisen, zum Erstellen anderer fehlender Tags in diesem Ausdruck den Standarddatentyp zu verwenden, ohne weitere Informationen von Ihnen anzufordern.

5.3.1.3 Kommunikationsreferenzen

Um ein Register in einem Kommunikationsgerät auszuwählen, wählen Sie im Dropdown-Menü ein Gerät aus. Ein Dialogfeld wird angezeigt, in dem Sie ein Register in diesem externen Kommunikationsgerät auswählen können. Die verschiedenen Kommunikationsgeräte werden am Ende des Menüs in der Reihenfolge aufgeführt, in der sie erstellt wurden. Sie können auch die Option „Next“ (Weiter) im Dropdown-Menü auswählen. Dadurch wird das aktuelle Tag mit der zuletzt verwendeten SPS-Registrierung plus der Anzahl der der Adresse zugeordneten Register gleichgesetzt. Wenn Sie z. B. ein 32-Bit-Tag dem Modbus-Register 40001 zuordnen und dann „Next“ (Weiter) wählen, wird das nachfolgende Tag dem Register 40003 zugeordnet.

5.3.1.4 Bearbeiten eines Ausdrucks

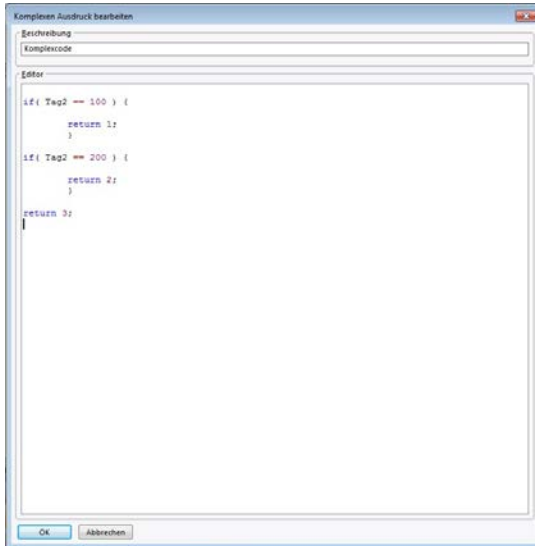
Wie oben erwähnt, werden allgemeine Ausdrücke in der Regel direkt im Bearbeitungsfeld der Eigenschaft bearbeitet. Alternativ können Sie neben dem Feld auf die Schaltfläche „Edit“ (Bearbeiten) klicken. Daraufhin wird ein spezielles Dialogfeld eingeblendet, in dem der Ausdruck wie folgt detaillierter angezeigt wird:



Der in diesem Dialogfeld verwendete Editor ist der gleiche Editor, der zum Erstellen globaler Programme verwendet wird. Er bietet daher eine Syntaxfärbung. Sie können außerdem Hilfeinformationen zu Systemfunktionen aufrufen, indem Sie den Cursor in oder am Ende des Funktionsnamens platzieren und F1 drücken.

5.3.1.5 Komplexe Ausdrücke

Wenn ein Ausdruck so komplex ist, dass er nicht in eine einzelne Zeile passt, oder wenn Sie ihn einfach lieber als Programm schreiben möchten, können Sie im Dropdown-Menü die Option „Complex“ (Komplex) auswählen. Sie können dann wie folgt ein lokales Programm erstellen:



Mit der Anweisung `return` wird der Wert des Ausdrucks bereitgestellt, als hätten Sie ein globales Programm aufgerufen. Beachten Sie, dass auch hier der Programmeditor verwendet wird, der Syntaxfärbung und automatischen Einzug bietet. Über den oben beschriebenen `F1`-Mechanismus können Hilfeinformationen zu Systemfunktionen abgerufen werden. Unter „Description“ (Beschreibung) können Sie rasch eine Notiz zur Funktion des Programms aufzeichnen. Dieser Text wird neben der Eigenschaft als Referenz angezeigt. Informationen zum Schreiben und Bearbeiten von Programmen finden Sie in späteren Kapiteln.

5.3.2 Übersetzbare Zeichenfolgen

Crimson-Datenbanken sind für den mehrsprachigen Betrieb konzipiert. Dies bedeutet, dass jede Zeichenfolge, die dem Benutzer des Zielgeräts präsentiert wird, in einer von vielen verschiedenen Sprachen angezeigt werden kann. Damit Sie diese Übersetzungen definieren können, befindet sich rechts neben Eigenschaften, die derartige Zeichenfolgen enthalten, eine Schaltfläche namens „Translate“ (Übersetzen).

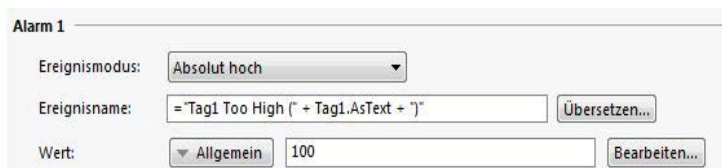


Klicken Sie auf die Schaltfläche, um die Übersetzungen einzugeben. Daraufhin wird folgendes Dialogfeld angezeigt:



Die im Dialogfeld aufgeführten Sprachen werden auf Datenbankebene definiert. Das Kapitel über Lokalisierung enthält weitere Informationen dazu, wie Sie Sprachen auswählen, die Funktion „Auto-Translate“ (Automatisch übersetzen) verwenden sowie während der Laufzeit die Sprache wechseln. Wenn Sie keinen Text für eine Sprache eingeben und diese Sprache anschließend vom Bediener ausgewählt wird, wird von Crimson stattdessen der Text in der Standardsprache angezeigt.

Übersetzbare Zeichenfolgen können auch als Ausdrücke definiert werden, damit sie zur Laufzeit geändert werden können. Beispielsweise wird ein Alarmname zwar normalerweise während der Konfiguration festgelegt, aber u. U. möchte ein Datenbank-Designer, dass der Alarm den Wert des Tags enthält, das den Alarm ausgelöst hat. Ausdrücke können mit einem Gleichheitszeichen als Präfix eingegeben werden, genau wie beim Bearbeiten einer Tabelle:



Beachten Sie, dass die Eigenschaft `AsText` des Tags zulässt, dass der Wert gemäß seiner Formateinstellung als Zeichenfolge abgerufen wird. Weitere Informationen finden Sie im Kapitel „Schreiben von Ausdrücken“.

5.3.3 Zwei-Wege-Eigenschaften

Eigenschaften wie übersetzbare Zeichenfolgen, die auf einen konstanten Wert oder einen Ausdruck eingestellt werden können, werden als Zwei-Wege-Eigenschaften bezeichnet. Sie akzeptieren nicht nur Ausdrücke mit einem Gleichheitszeichen als Präfix, sondern können auch einfach durch Ziehen des entsprechenden Tags aus dem Ressourcenfenster auf ein Feld auf Tag-Werte gesetzt werden.

5.3.4 Aktionseigenschaften

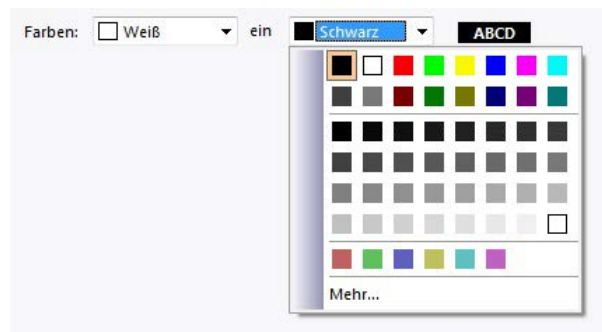
Aktionseigenschaften werden in Tags verwendet, um die Aktion zu definieren, die ausgeführt werden soll, wenn bestimmte Bedingungen erfüllt sind oder wenn ein Tag-Wert geändert wird. Für deren Bearbeitung stehen ein Dropdown-Menü und ein Bearbeitungsfeld zur Verfügung, ähnlich wie bei der Bearbeitung von Ausdrücken:



Genau wie bei Ausdrücken kann auch hier über die Schaltfläche „Edit“ (Bearbeiten) ein größeres Bearbeitungsfenster aufgerufen werden. Komplexe Aktionen können mithilfe lokaler Programme erstellt werden.

5.3.5 Farbeigenschaften

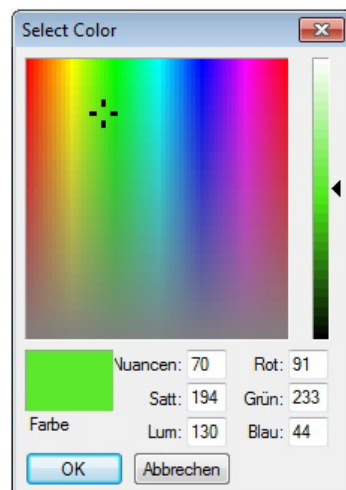
Farbeeigenschaften in Tags stellen ein Farbpaar dar. Dabei handelt es sich um die Vorder- und Hintergrundfarbe, die bei der Anzeige des Tag-Status in Textform verwendet werden. Im folgenden Beispiel sehen Sie ein Farbpaar, das bearbeitet wird:



Das Dropdown-Menü enthält die folgenden Farben:

- die sechzehn VGA-Standardfarben
- zweiunddreißig Grautöne zwischen Schwarz und Weiß
- alle anderen in der Datenbank verwendeten Farben, bis maximal vierundzwanzig

Mit der Option „More“ (Mehr) am Ende der Liste können Sie das Farbauswahl-Dialogfeld aufrufen:



Dieses Dialogfeld bietet mehrere Möglichkeiten zum Definieren einer Farbe. Sie können eine Farbe in der Palette oder im „Regenbogen“-Fenster auswählen oder die expliziten HSL- oder RGB-Parameter eingeben. Wenn die ausgewählte Farbe noch nicht in der Datenbank verwendet wurde und nicht zu den Standardfarben oder Graustufen gehört, wird sie den benutzerdefinierten Farben im Dropdown-Menü hinzugefügt.

5.4 Protokolleigenschaften

Wenn Sie zum ersten Mal die Kategorie „Daten-Tags“ des Navigationsfensters aufrufen, sehen Sie eine Reihe von Eigenschaften rund um die Ereignisprotokollierung. Diese Eigenschaften steuern, ob und wie Ereignisse, die von Tags oder deren Alarmen generiert werden, auf der Speicherkarte gespeichert werden. Sie sind analog zu den von Datenprotokollen definierten Eigenschaften. Weitere Informationen zu deren Verwendung finden Sie Kapitel „Verwenden des Daten-Loggers“.

5.5 Erstellen von Tags

Daten-Tags werden im Navigationsfenster anhand der üblichen Methoden erstellt und bearbeitet. Sie können Ordner erstellen, um Ihre Tags zu organisieren. Die Schaltfläche „New“ (Neu) in der Symbolleiste enthält einen Dropdown-Pfeil, über den Sie den Typ des einzufügenden Tags auswählen können. Über den linken Bereich der Schaltfläche „New“ (Neu) wird ein Tag des gleichen Typs wie das zuletzt erstellte Tag generiert. Sie können also einfach mehrere Tags erstellen, ohne die Dropdown-Liste zu öffnen.



5.6 Duplizieren von Tags

Mit dem Befehl „Smart Duplicate“ (Smart duplizieren) im Menü „Edit“ (Bearbeiten) können Sie eine neue Kopie eines vorhandenen Tags erstellen und dabei dessen Datenquelle automatisch so hochzählen, dass auf das nächste Datenelement verwiesen wird.

Was unter dem „nächsten“ Datenelement zu verstehen ist, ist vom genauen Typ des Datenelements abhängig. Crimson kann das nächste Register in einem Kommunikationsgerät, das nächste Mitglied eines Arrays oder das nächste Tag in einer Sequenz auswählen. Wenn Sie „Smart Duplicate“ (Smart duplizieren) beispielsweise bei einem 16-Bit-Tag verwenden, das dem Modbus-Register 40001 zugeordnet ist, wird ein Tag erstellt, das 40002 zugeordnet ist. Wenn Sie den Befehl bei einem Tag verwenden, das `Array[2]` zugeordnet ist, wird ein Tag erstellt, das `Array[3]` zugeordnet ist.

Diese Funktion erleichtert das Erstellen ganzer Sätze von Tags, die auf aufeinander folgende Datenelemente verweisen.

5.7 Bearbeiten mehrerer Tags

Möglicherweise möchten Sie die Eigenschaften mehrerer Tags gleichzeitig bearbeiten. In Crimson ist dies möglich. Sie bearbeiten ein Tag und können dann die Eigenschaften der anderen Tags auf die Eigenschaften des zuerst bearbeiteten Tags setzen. Crimson bietet hierfür zwei Methoden, die beide auf demselben Mechanismus beruhen.

5.7.1 Verwenden von Kopieren von

Mit dem Befehl „Copy From“ (Kopieren von) können die ausgewählten Eigenschaften eines Tags in ein oder mehrere Tags in der Navigationsliste kopiert werden. Zur Verwendung des Befehls wählen Sie die Zieltags aus und klicken dann mit der rechten Maustaste, um das Kontextmenü aufzurufen. (Die Navigationsliste für Tags unterstützt die Mehrfachauswahl über die üblichen Tastenkombinationen mit **UMSCHALT** und **STRG**.) Wählen Sie einen der Befehle „Copy From“ (Kopieren von) aus. Der Cursor ändert sich, damit Sie das Tag auswählen können, von dem aus der Kopiervorgang durchgeführt werden soll. Je nach ausgewähltem Befehl werden dann eine oder mehrere Eigenschaften aus dem Quelltag auf die Zieltags angewendet.

5.7.2 Verwenden von Einfügen spezial

Mit dem Befehl „Paste Special“ (Einfügen spezial) können Sie dasselbe Ergebnis erzielen. Bei dieser Methode ist es jedoch möglich, Eigenschaften auch zwischen Datenbanken oder zwischen mehreren Instanzen von Crimson zu kopieren. Wählen Sie zuerst das Quelltag und dann den Befehl „Copy“ (Kopieren) aus, um das Tag und seine Eigenschaften in die Zwischenablage zu kopieren. Wählen Sie dann in der Navigationsliste die Zieltags aus. Denken Sie daran, dass Sie mehrere Tags auswählen können. Klicken Sie zum Schluss mit der rechten Maustaste auf die Auswahl, um das Kontextmenü aufzurufen, und wählen Sie den Befehl „Paste Special“ (Einfügen spezial) aus. Das folgende Dialogfeld wird angezeigt:



Die ausgewählten Eigenschaften des Quelltags werden auf die Zieltags angewendet.

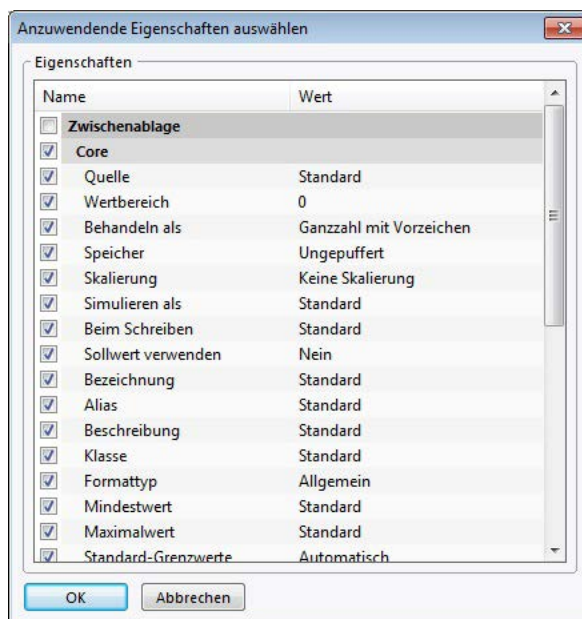
5.7.3 Eigenschaftsauswahl

Bei beiden oben beschriebenen Methoden können Sie festlegen, welche Eigenschaften kopiert werden sollen:

- *Mapping* (Zuordnung) kopiert die Quelleigenschaft. Außerdem werden alle Eigenschaften kopiert, die die Kommunikationsoptionen des Tags steuern, wie z. B. Umfang, Zugriff und alle weiteren Eigenschaften, die im Abschnitt „Datenquelle“ enthalten sind.
- *Scaling* (Skalierung) kopiert die Eigenschaft „Scale To“ (Skalieren auf) und die zugehörigen Skalierungsgrenzwerte.
- *Format* kopiert der Formattyp und das zugehörige Formatobjekt.
- *Coloring* (Färbung) kopiert den Farbtyp und das zugehörige Färbungsobjekt.
- *Alarms* (Alarme) kopiert alle Eigenschaften von Alarm 1 und Alarm 2.
- *Triggers* (Auslöser) kopiert alle Eigenschaften von Trigger 1 und Trigger 2.

- *Security* (Sicherheit) kopiert alle Eigenschaften von der Sicherheitsseite des Tags.

Darüber hinaus kann die Option *Selective* (Selektiv) wie folgt zur Auswahl der zu kopierenden Eigenschaften verwendet werden:



Die Liste enthält eine hierarchische Aufstellung aller Eigenschaften, die vom Quelltag definiert werden. Die Eigenschaften sind anhand des Layouts angeordnet, das bei der Bearbeitung des Tags verwendet wurde. Für jede Eigenschaft wird zudem der ihr zugewiesene Wert angezeigt. Die Eigenschaften oder Gruppen von Eigenschaften können mithilfe der entsprechenden Kontrollkästchen aktiviert oder deaktiviert werden. Nur die aktivierten Eigenschaften werden angewendet, sodass Sie im Detail steuern können, was von einem Tag zum andern kopiert wird.

5.8 Importieren und Exportieren

Durch Auswahl von „Daten-Tags“ in der Navigationsliste können Sie auf Schaltflächen zugreifen, mit denen Daten-Tags aus der Datenbank exportiert und in die Datenbank importiert werden können. Tags können entweder in Unicode-Textdateien oder in ANSI-CSV-Dateien (Comma Separated Variable) exportiert werden, die jeweils in Anwendungen wie Microsoft Excel bearbeitet werden können. Die Exportdatei ist auf der Grundlage von Tag-, Format- und Farbtyp in Abschnitte unterteilt. Jeder Abschnitt enthält mehrere Spalten, deren Bedeutungen anhand der nachfolgenden Abschnitte bestimmt werden können.

Beachten Sie, dass bestimmte Kommunikationstreiber z. B. eine SPS-Konfigurationsdatei importieren und Daten-Tags erstellen können, die den SPS-Registern entsprechen. Diese Funktion wird auf der Gerätekonfigurationsseite mit dem Befehl „Make Data Tags“ (Daten-Tags erstellen) aufgerufen.

5.9 Suchen nach der Tag-Nutzung

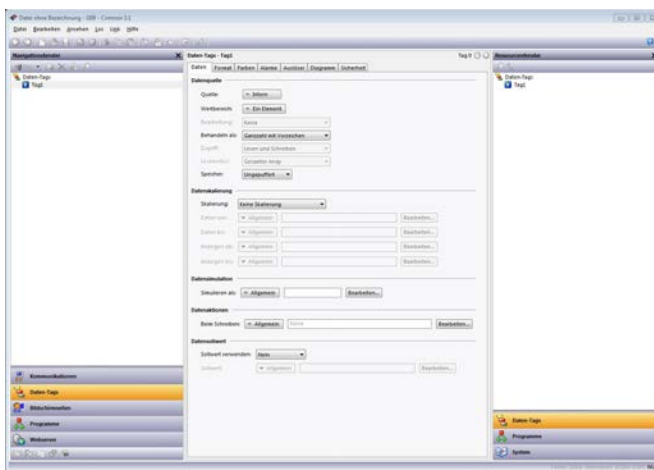
Wenn Sie alle Elemente anzeigen möchten, die auf ein bestimmtes Tag verweisen, klicken Sie im Navigationsfenster mit der rechten Maustaste auf das Element und wählen Sie den Befehl „Find Usage“ (Nutzung suchen) aus. Die gefundenen Elemente werden in der globalen Suchergebnisliste aufgeführt, und Sie können über die Taste F4 bzw. die Tastenkombination UMSCHALT+F4 darauf zugreifen. Die Liste selbst kann durch Drücken der Taste F8 ein- oder ausgeblendet werden.

5.10 Numerische Tags

Ein numerisches Tag stellt eine oder mehrere Ganzzahlen bzw. einen oder mehrere Gleitkommawerte dar. Crimson führt alle internen Berechnungen entweder mit vorzeichenbehafteten 32-Bit-Ganzzahlen oder Gleitkommawerten mit einfacher Genauigkeit durch, sodass alle Daten vor der Verarbeitung in eine dieser Formen konvertiert werden. (Es gibt bestimmte Funktionen, mit denen 64-Bit-Mathematik über Paare von 32-Bit-Array-Elementen durchgeführt werden kann. Weitere Informationen finden Sie im Referenzhandbuch.) Zugeordnete numerische Tags unterstützen mehrere Transformationen, die zwischen den Rohdaten und den von Crimson verwendeten Daten auftreten. Der genaue Vorgang wird später in diesem Kapitel näher erläutert.

5.10.1 Dateneigenschaften

Für ein numerisches Tag werden auf der Registerkarte „Data“ (Daten) die folgenden Eigenschaften angezeigt:



5.10.1.1 Datenquelle

- Die Eigenschaft *Source* (Quelle) gibt an, woher die Daten des Tags stammen. Die Standardeinstellung führt zu einem internen Tag, aber in der Dropdown-Liste kann auch ein allgemeiner Ausdruck, ein anderes Daten-Tag oder ein Element von einem externen Gerät ausgewählt werden.
- Die Eigenschaft *Extent* (Umfang) dient zur Wahl zwischen einem einzelnen Element-Tag oder einem Array. Wenn Sie ein Array auswählen, müssen Sie die erforderliche Anzahl von Elementen eingeben. Arrays sind bei Tags, deren Quelle ein Ausdruck ist, nicht zulässig. Bei zugeordneten Elementen ist die Anzahl der aus dem externen Gerät zu lesenden Register abhängig vom Datentyp, der für die Adresse definiert ist. Beispielsweise wird bei einem Array aus zwei Elementen, das einem Register des Typs „Word as Long“ (Wort als Lang) zugeordnet ist, auf vier Register zugegriffen, wobei für jeden Long-Wert jeweils zwei Wörter benötigt werden. Bei einem ähnlichen Array, das dem Datentyp „Word as Word“ (Wort als Wort) zugeordnet ist, sind nur zwei Register erforderlich.

- Die Eigenschaft *Manipulation* definiert die erste Transformationsphase, die bei der Übertragung von Kommunikationsdaten in ein zugeordnetes Tag angewendet wird. Je nach verwendetem Datentyp sind die folgenden Optionen verfügbar:

Manipulation	Beschreibung
Keine	Die Daten werden nicht verändert.
Bits invertieren	Der Status jedes Bits in den Daten wird umgekehrt.
Reverse Bits	Das höchstwertige Bit in den Daten wird mit dem niedrigstwertigen Bit vertauscht, und alle dazwischen liegenden Bits werden in ähnlicher Weise behandelt.
Reverse Byte	Das höchstwertige Byte in den Daten wird mit dem niedrigstwertigen Byte vertauscht usw. Nur verfügbar für Datenelemente mit einer Größe ab 16 Bit.
Wörter umdrehen	Das höchstwertige Wort in den Daten wird mit dem niedrigstwertigen Wort vertauscht usw. Nur verfügbar für Datenelemente mit einer Größe von genau 32 Bit.
BCD zu Binär	Jede 4-Bit-Gruppe in den Daten wird als einzelne Dezimalziffer interpretiert. Bei Auswahl dieser Option werden die Daten in eine Ganzzahl ohne Vorzeichen übertragen.

- Die Eigenschaft *Treat As* (Behandeln als) für interne Tags definiert den Datentyp des Tags. Bei zugeordneten Tags gibt sie an, wie die manipulierten Daten von Crimson zu interpretieren sind. Die Eigenschaft wird ausgehend vom zugrunde liegenden Datentyp auf den Standardwert gesetzt, wenn das Tag zugeordnet wird, kann jedoch geändert werden. Beachten Sie, dass die Eigenschaft „Treat As“ (Behandeln als) bei den meisten Tags nicht endgültig über den tatsächlichen Datentyp des Tags entscheidet, weil die Daten mit den Skalierungseigenschaften weiter konvertiert werden können. Je nach dem genauen Datentyp der Kommunikationsdaten sind die folgenden Optionen verfügbar:

Behandeln als	Beschreibung
Signierter Integer	Die Daten werden als vorzeichenbehafteter 32-Bit-Wert behandelt, wobei kleinere Datenwerte mit Vorzeichen erweitert werden. Beispielsweise wird der 16-Bit-Wert 0x8000 in 0xFFFF8000 konvertiert.
Unsignierter Integer	Die Daten werden als vorzeichenbehafteter 32-Bit-Wert behandelt, wobei kleinere Datenelementen mit Nullen erweitert werden. Beispielsweise wird der 16-Bit-Wert 0x8000 in 0x00008000 konvertiert. Nur verfügbar für Datenelemente mit einer Größe von weniger als 32 Bit.
Standard-Integer	Die Daten werden je nach Einstellung des Kommunikationstreibers, von dem die Daten abgerufen werden, entweder mit Nullen oder mit Vorzeichen erweitert. Nur verfügbar für Datenelemente mit einer Größe von weniger als 32 Bit.
Gleitkomma	Die Daten werden als 32-Bit-Gleitkommawert mit einfacher Genauigkeit behandelt. Nur verfügbar für Datenelemente mit einer Größe von genau 32 Bit.

- Mit der Eigenschaft *Access* (Zugriff) wird bei zugeordneten Tags definiert, welche Kommunikationsvorgänge zulässig sein sollen. Interne Tags sind immer auf Lese- und Schreibzugriff gesetzt, und Ausdrucks-Tags sind immer schreibgeschützt.

- Die Eigenschaft *Read Mode* (Lesemodus) wird nur für Array-Tags verwendet. Sie gibt die Elemente an, die gelesen werden sollen, wenn das Array referenziert wird. Folgende Einstellungen können verwendet werden:

Lesemodus	Beschreibung
Gesamtes Array	Alle Elemente im Array werden gelesen, wenn das Array referenziert wird, und der Zugriff wird blockiert, bis der Lesevorgang abgeschlossen ist. Dadurch wird zwar sichergestellt, dass Daten verfügbar sind, aber die Durchführung ist am langsamsten.
Manueller Modus	Das Array wird erst gelesen, wenn ein Aufruf an die Funktion <code>ReadData</code> erfolgt, um ein einmaliges manuelles Update zu erzwingen.
Auf Anforderung	Array-Elemente werden gelesen, sobald sie referenziert werden. Dadurch wird die schnellste Durchführung erzielt, aber möglicherweise werden veraltete Daten zurückgegeben, wenn ein Element zum ersten Mal aufgerufen wird.
N auf beiden Seiten	Die Operation „On Demand“ (Auf Anforderung) wird zwar verwendet, aber N Register auf beiden Seiten des referenzierten Registers werden ebenfalls gelesen, wodurch angrenzende Daten schneller verfügbar gemacht werden.

- Mit der Eigenschaft *Storage* (Speicher) wird angegeben, ob das Tag bei einem Aus- und Einschaltvorgang des Zielgeräts beibehalten wird. Sie wird in der Regel für interne Tags verwendet, aber auch die Werte von zugeordneten Nur-schreiben-Tags können beibehalten werden.

5.10.1.2 Datenskalierung

- Die Eigenschaft *Scaling* (Skalierung) wird für zugeordnete Tags verwendet, um einen abschließenden Skalierungsschritt für die Daten zu definieren. Die Daten können auf Ganzzahlen oder Gleitkommawerte skaliert werden, unabhängig davon, wie Crimson die manipulierten Kommunikationsdaten behandelt. Ein ganzzahliger Wert kann beispielsweise auf einen Gleitkommawert skaliert werden. In diesem Fall wird das Tag von Crimson als Gleitkommazahl betrachtet. Ebenso kann ein Gleitkommawert zurück in eine Ganzzahl konvertiert werden, möglicherweise ohne dabei dessen Größe zu verändern.
- Mit den Eigenschaften *Data From* (Daten von) und *Data To* (Daten zu) wird der Bereich der Transformation beim Lesen und der Umfang der Transformation beim Schreiben definiert. Die Werte müssen mit dem unter „Treat As“ (Behandeln als) angegebenen Datentyp übereinstimmen, damit in diese Felder nur für Daten, die als Gleitkommawert behandelt werden, nicht integrale Werte eingegeben werden können. Beim Lesen werden Werte außerhalb dieser Grenzwerte weiterhin akzeptiert und auf die entsprechenden Werte außerhalb der Anzeigegrenzen skaliert. Mit anderen Worten: Es werden keine Werte abgeschnitten.
- Mit den Eigenschaften *Display From* (Anzeigen von) und *Display To* (Anzeigen zu) wird der Umfang der Transformation beim Lesen und der Bereich der Transformation beim Schreiben definiert. Die Werte müssen mit dem unter „Scale As“ (Skalieren als) angegebenen Datentyp übereinstimmen, damit in diese Felder nur für Daten, die auf Gleitkommawerte skaliert werden, nicht integrale Werte eingegeben werden können. Beim Schreiben werden Werte außerhalb dieser Grenzwerte weiterhin akzeptiert und auf die entsprechenden Werte außerhalb der Datenlimits skaliert. Mit anderen Worten: Es werden keine Werte abgeschnitten.

5.10.1.3 Datensimulation

- Die Eigenschaft *Simulate As* (Simulieren als) definiert den angenommenen Wert, der bei der Arbeit im Anzeigeseiten-Editor für das Tag verwendet werden soll. Durch die Eingabe eines sinnvollen Werts stimmt die Darstellung der Seite stärker mit ihrem tatsächlichen Erscheinungsbild überein. Dieser Wert wird auch als Standardwert des Tags verwendet, wenn die Kommunikation global deaktiviert ist.

5.10.1.4 Datenaktivitäten

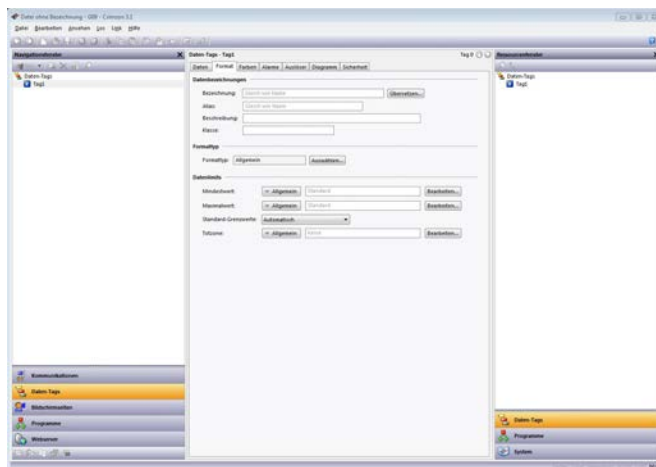
- Die Eigenschaft *On Write* (Bei Schreiben) definiert eine Aktion, die bei einer Änderung des Tags aufgerufen werden soll. Die Systemvariable *Data* (Daten) enthält den neuen Datenwert, wenn der Schreibvorgang erfolgt und wenn die Aktion ausgeführt wird. Die Verwendung der Eigenschaften „On Write“ (Bei Schreiben) wird später in diesem Kapitel behandelt.

5.10.1.5 Daten Sollwert

- Die Eigenschaft *Use Setpoint* (Sollwert verwenden) dient zum Aktivieren oder Deaktivieren eines Sollwerts für dieses Tag.
- Die Eigenschaft *SP Value* (SP-Wert) definiert einen Ausdruck oder ein anderes Tag, dem dieses Tag nominell folgen soll. Dieser Sollwert kann dann in Alarmen oder Primitiven verwendet werden, um verschiedene Funktionen zu implementieren.

5.10.2 Formateigenschaften

Für ein numerisches Tag werden auf der Registerkarte „Format“ die folgenden Eigenschaften angezeigt:



5.10.2.1 Datenbezeichnungen

- Die Eigenschaft *Label* (Bezeichnung) wurde bereits unter „Tag-Attribute“ erörtert.
- Die Eigenschaft *Description* (Beschreibung) wurde bereits unter „Tag-Attribute“ erörtert.
- Die Eigenschaft *Class* (Klasse) ist für zukünftige Erweiterungen reserviert.

5.10.2.2 Formattyp

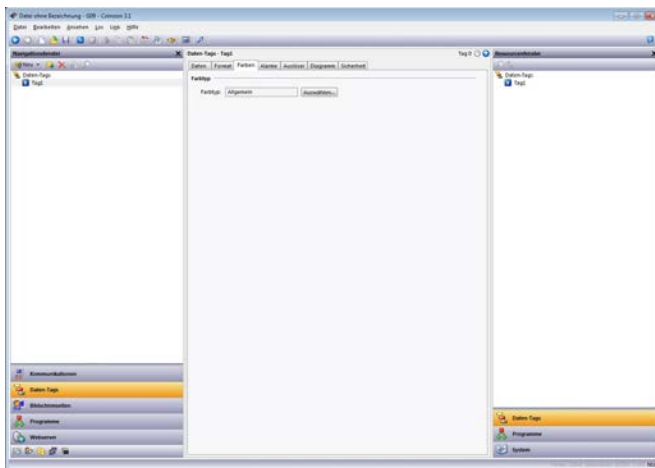
- Die Eigenschaft *Format Type* (Formattyp) wählt das Format für dieses Tag aus. Die verschiedenen Arten von Formaten sowie weitere Eigenschaften, die je nach ausgewähltem Formattyp angezeigt werden können, werden in einem späteren Kapitel ausführlich beschrieben.

5.10.2.3 Datenlimits

- Mit den Eigenschaften *Minimum Value* (Mindestwert) und *Maximum Value* (Maximalwert) können Sie manuell Grenzwerte für die Dateneingabe festlegen. Wenn keine Werte angegeben werden, basieren die Standardwerte auf der unten beschriebenen Eigenschaft *Default Limits* (Standardlimits).
- Die Eigenschaft *Default Limits* (Standardlimits) definiert, wie die Grenzwerte für die Dateneingabe bei einem Tag bestimmt werden, wenn nichts angegeben wurde. Bei der Einstellung „Automatic“ (Automatisch) wird der auf der Registerkarte „Data“ (Daten) angegebene Anzeigebereich als primäre Quelle verwendet. Alternativ wird das Formatobjekt verwendet, wenn die Skalierung nicht aktiviert ist. Wenn keine Quelle einen Bereich definieren kann, wird der maximal unterstützte Bereich für den Datentyp des Tags verwendet. Die Einstellung „From Format“ (Aus Format) erzwingt die Verwendung des Formatobjekts.
- Mit der Eigenschaft *Deadband* (Totzone) wird definiert, inwieweit der Wert des Tags geändert werden muss, damit Updates an OPC UA-Clients gesendet werden. Normale Kommunikationstreiber verwenden diese Einstellung nicht und verarbeiten daher Schreibvorgänge, sobald der Wert sich ändert. Die Eigenschaft wird nicht auf Geräten angezeigt, die den OPC UA-Server nicht unterstützen.

5.10.3 Farbeigenschaften

Für ein numerisches Tag werden auf der Registerkarte „Colors“ (Farben) die folgenden Eigenschaften angezeigt:



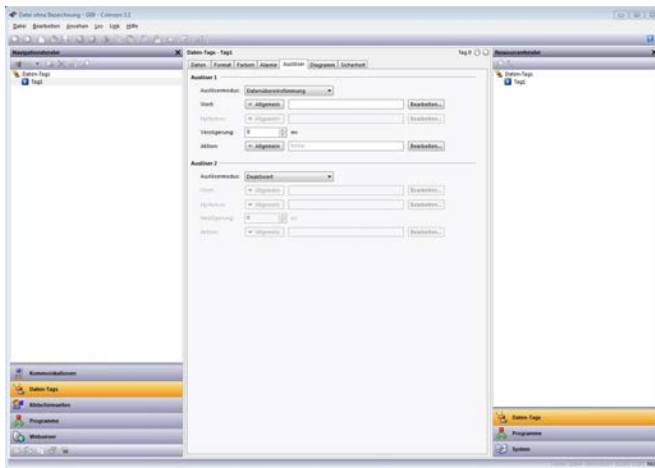
5.10.3.1 Farbtyp

- Die Eigenschaft *Color Type* (Farbtyp) definiert die Farbe für dieses Tag. Die verschiedenen Arten von Färbungen sowie weitere Eigenschaften, die je nach ausgewählter Option angezeigt werden können, werden in einem späteren Kapitel ausführlich beschrieben.

- Die Eigenschaft *Event Name* (Ereignisname) definiert den Namen, der bei einem Verweis auf dieses Ereignis im Alarmbetrachter oder im Ereignisprotokoll angezeigt wird.
- Die Eigenschaft *Value* (Wert) definiert entweder den absoluten Wert, bei dem der Alarm aktiviert wird, die Abweichung vom Sollwert oder die Wertänderung, die seit der letzten Alarmauslösung stattfinden muss. Die genaue Interpretation hängt vom Ereignismodus ab, wie oben beschrieben.
- Die Eigenschaft *Hysteresis* (Hysterese) verhindert, dass ein Alarm zwischen den Zuständen „Ein“ und „Aus“ hin und her wechselt, wenn sich der Prozess in der Nähe der Alarmbedingung befindet. Bei einem Alarm mit einem absoluten Höchstwert bedeutet dies z. B. Folgendes: Der Alarm wird aktiv, wenn das Tag den Wert des Alarms überschreitet. Er wird aber nur deaktiviert, wenn das Tag um einen Betrag unter den Wert fällt, der größer oder gleich der Hysterese des Alarms ist. Denken Sie daran, dass die Eigenschaft stets darauf ausgelegt ist, einen Alarm aufrecht zu erhalten, wenn er einmal aktiviert wurde. Sie ist nicht darauf ausgelegt, den Punkt zu ändern, an dem die Aktivierung erfolgt.
- Die Eigenschaft *Enable* (Aktivieren) definiert einen Ausdruck, der den Alarm aktiviert oder deaktiviert. Bei einem Wert von nicht Null oder einem leeren Wert wird der Alarm aktiviert. Bei einem Nullwert wird der Alarm deaktiviert.
- Mit der Eigenschaft *Trigger* wird angegeben, ob der Alarm flankengesteuert oder pegelabhängig ausgelöst werden soll. Im ersteren Fall wird der Alarm ausgelöst, wenn die durch den Ereignismodus definierte Bedingung zum ersten Mal wahr wird. Im zweiten Fall verbleibt der Alarm im aktiven Zustand, solange die Bedingung besteht. Mit dieser Eigenschaft können Sie auch angeben, dass dieser Alarm nur als Ereignis verwendet werden soll. In diesem Fall wird der Alarm zwar flankengesteuert ausgelöst, führt aber nicht zu einer Alarmbedingung. Stattdessen wird ein Ereignis im internen Speicher und optional auf der Speicherkarte protokolliert.
- Mit der Eigenschaft *Delay* (Verzögerung) wird angegeben, wie lange die Alarmbedingung bestehen muss, damit der Alarm aktiv wird. Im Falle eines flankengesteuerten Alarms oder Ereignisses gibt diese Eigenschaft auch an, wie lange die Alarmbedingung nicht mehr bestehen muss, damit nachfolgende Reaktivierungen zu einem weiteren Alarm führen. Angenommen, ein Alarm soll gemäß Einstellung aktiviert werden, wenn ein Geschwindigkeitsschalter anzeigt, dass ein Motor nicht läuft, obwohl der Start dieses Motors angefordert wurde. In diesem Fall kann dem Motor mithilfe dieser Eigenschaft eine Anlaufzeit zugewiesen werden, bevor der Alarm aktiviert wird.
- Mit der Eigenschaft *Accept* (Akzeptieren) wird angegeben, ob der Benutzer einen Alarm explizit akzeptieren muss, damit er nicht mehr angezeigt wird. Flankengesteuerte Alarme müssen immer manuell akzeptiert werden.
- Die Eigenschaft *Priority* (Priorität) dient zur Steuerung der Reihenfolge, in der Alarme im Alarmbetrachter von Crimson angezeigt werden. Je niedriger der numerische Wert im Feld „Priority“ (Priorität), desto weiter oben wird der Alarm angezeigt.
- Mit der Eigenschaft *Siren* (Sirene) wird angegeben, ob die Aktivierung dieses Alarms auch den Signalgeber des Zielgeräts aktivieren soll. Während der akustische Signalgeber aktiv ist, blinkt auch das Display des Bedienfelds, um auf die Alarmbedingung aufmerksam zu machen.
- Die Eigenschaft *Mail To* (Senden an) gibt den Eintrag im E-Mail-Adressbuch an, an den eine Nachricht gesendet werden soll, wenn dieser Alarm aktiviert wird. Informationen zur E-Mail-Konfiguration finden Sie im Kapitel „Verwenden von Diensten“.
- Mit den Eigenschaften *On Accept* (Bei Akzeptieren), *On Active* (Bei Aktiv), *On Clear* (Bei Löschen) und *On Event* (Bei Ereignis) werden Aktionen angegeben, die bei der Änderung des Zustands ausgeführt werden sollen. Welche Aktionen verfügbar sind, richtet sich nach dem Auslösemodus des Alarms und der unter „Accept“ (Akzeptieren) gewählten Option.

5.10.5 Auslösereigenschaften

Für ein numerisches Tag werden auf der Registerkarte „Trigger“ (Auslöser) die folgenden Eigenschaften angezeigt:



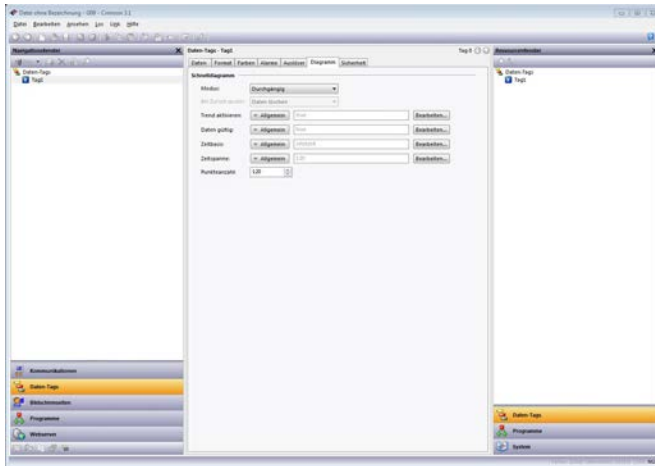
5.10.5.1 Für jeden Auslöser

- Die Eigenschaft *Trigger Mode* (Auslösemodus) stimmt mit der Beschreibung für die Registerkarte „Alarms“ (Alarmer) überein.
- Die Eigenschaften *Value* (Wert) und *Hysteresis* (Hysterese) stimmen mit der Beschreibung für die Registerkarte „Alarms“ (Alarmer) überein.
- Die Eigenschaft *Delay* (Verzögerung) stimmt mit der Beschreibung für die Registerkarte „Alarms“ (Alarmer) überein.
- Mit der Eigenschaft *Action* (Aktion) wird angegeben, welche Aktion ausgeführt werden soll, wenn der Auslöser aktiviert ist. Eine Beschreibung der Syntax, die zum Definieren der verfügbaren Aktionen verwendet wird, finden Sie im Kapitel „Schreiben von Aktionen“.

5.10.6 Diagrammeigenschaften

„Quick Plot“ ist eine zusätzliche Funktion für numerische Tags, die eine einfache grafische Verfolgung von Tag-Werten ermöglicht. Sobald diese Option aktiviert und konfiguriert ist, kann das Tag-Diagramm im Ressourcenfenster von der Kategorie „Core Primitives“ (Kernprimitive) aus einer Anzeigeseite hinzugefügt werden. Klicken Sie auf das Quick Plot-Primitiv, ziehen Sie es auf die gewünschte Anzeigeseite, und passen Sie bei Bedarf die Größe an.

Die Registerkarte „Plot“ (Diagramm) enthält die folgenden Eigenschaften:



- Mit der Eigenschaft *Mode* (Modus) wird festgelegt, wie Daten aufgezeichnet werden. Im Modus *Continuous* (Durchgängig) erfolgt die Aufzeichnung in einem kreisförmigen Puffer, wobei alte Werte verworfen werden. Dies ist der am häufigsten verwendete Modus. *One-Shot Relative* (Einmalig relativ) startet die Aufzeichnung, wenn die Aktivierung „true“ wird, und stoppt, wenn der Puffer voll ist oder wenn die Aktivierung zu „false“ wechselt. Der Zeitwert, der jeder Probe zugeordnet ist, steht in Bezug zu dem Zeitpunkt, zu dem das Diagramm gestartet wurde. *OneShot Absolute* (Einmalig absolut) ist ähnlich, doch hier alle Zeitwerte sind Null-basiert.
- *On Rewind* (Bei Rücklauf) gibt an, was zu tun ist, wenn die Zeit rückwärts läuft. Dies mag merkwürdig klingen, ist aber durchaus möglich, weil die Zeitbasis eine Variable sein kann und möglicherweise nicht mit der tatsächlichen Tageszeit verknüpft ist. Es könnte sich beispielsweise um einen Positionswert handeln, sodass das Diagramm das Profil zeigt, das von einem Teil einer Maschine nachgezeichnet wird. Es gibt dann zwei Möglichkeiten: Entweder werden die Daten nach dem Zeitpunkt, zu dem wir zurückgegangen sind, gelöscht, oder alle Daten im Puffer werden verschoben, sodass die alten Daten beibehalten, aber zeitlich betrachtet nach hinten verschoben werden.
- *Enable* (Aktivieren) startet und stoppt den Trend.
- *Data Valid* (Daten gültig) gibt an, ob die aufgezeichneten Daten gültig sind. Wenn dieser Ausdruck auf Null gesetzt wird, können Lücken in den Daten aufgezeichnet werden, ohne den Trend zu stoppen und damit beim erneuten Start alle Daten zu verlieren.
- *Time Base* (Zeitbasis) ist standardmäßig die Systemzeit. Sie wird zur Definition der Zeitbasis verwendet. Bei einigen Anwendungen (z. B. Aufzeichnung der Ramp/Soak-Leistung eines externen Controllers) kann eine externe Zeitbasis verwendet werden. Bei anderen Anwendungen wiederum (z. B. Verfolgung eines Bearbeitungsprofils) ist sie möglicherweise gar kein Zeitwert.
- *Time Span* (Zeitspanne) ist die Anzahl der Zeitbasismarkierungen, die im Puffer aufgezeichnet werden sollen. Beachten Sie, dass dieser Wert in der Regel größer ist als die Anzahl der Punkte und zusammen mit dieser Variablen definiert, wie viele Markierungen jeder Slot aufnimmt.
- *Point Count* (Punktzahl) ist die Anzahl der im Puffer zu speichernden Punkte. Da „Quick Plot“ darauf ausgelegt ist, die Änderungen eines Tags im Zeitverlauf auf einfache Weise darzustellen, ist dieser Wert in der Regel geringer als die Anzahl der Pixel im Display.

5.10.7 Sicherheitseigenschaften

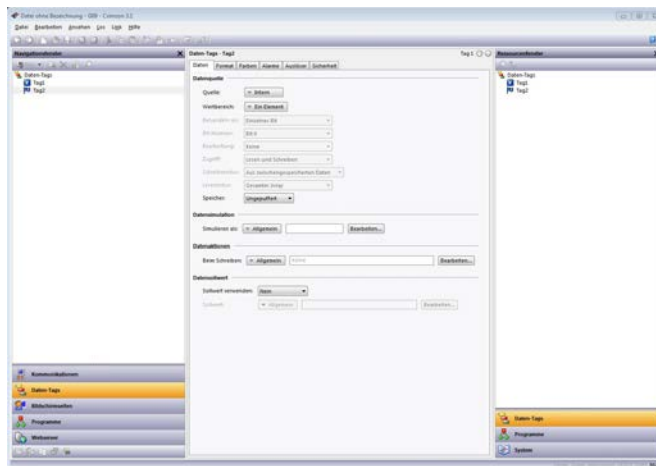
Informationen zu Sicherheitsbeschreibungen finden Sie im Kapitel „Verwenden von Sicherheit“.

5.11 Markierungs-Tags

Ein Markierungs-Tag stellt einen oder mehrere Ein- oder Aus-Werte dar. Unabhängig von der Art der zugrunde liegenden Daten wird davon ausgegangen, dass es vom internen Datentyp „Ganzzahl“ ist. Zugeordnete Markierungs-Tags ermöglichen einfache Transformationen zwischen Rohdaten und den von Crimson verwendeten Daten.

5.11.1 Dateneigenschaften

Für ein Markierungs-Tag werden auf der Registerkarte „Data“ (Daten) die folgenden Eigenschaften angezeigt:



5.11.1.1 Datenquelle

- Die Eigenschaft *Source* (Quelle) gibt an, woher die Daten des Tags stammen. Die Standardeinstellung führt zu einem internen Tag, aber in der Dropdown-Liste kann auch ein allgemeiner Ausdruck, ein anderes Daten-Tag oder ein Element von einem externen Gerät ausgewählt werden.
- Die Eigenschaft *Extent* (Umfang) dient zur Wahl zwischen einem einzelnen Element-Tag oder einem Array. Wenn Sie ein Array auswählen, müssen Sie die erforderliche Anzahl von Elementen eingeben. Arrays sind bei Tags, deren Quelle ein Ausdruck ist, nicht zulässig. Bei zugeordneten Tags ist die genaue Anzahl der von externen Geräten zu lesenden Register vom Typ der Register abhängig, denen das Tag zugeordnet ist, sowie von der Eigenschaft „Treat As“ (Behandeln als).
- Mit der Eigenschaft *Treat As* (Behandeln als) wird bei zugeordneten Tags festgelegt, wie der Ein-oder-Aus-Wert aus den Rohdaten der Kommunikation und umgekehrt abgeleitet werden soll. Je nach dem zugrunde liegenden Datentyp sind die folgenden Einstellungen verfügbar:

Behandeln als	Ergebnis
Unsignierter Integer	Das Tag ist „true“, wenn die Daten nicht Null sind, oder „false“, wenn sie Null sind. Der Wert „true“ wird als ganzzahliger Wert 1 geschrieben und der Wert „false“ als Null. Bei einem zugeordneten Array entspricht jedes Array-Element einem einzelnen Kommunikationsdatenelement. Diese Einstellung ist für alle Kommunikationsdaten mit einer Größe von 8 Bit und mehr verfügbar.
Gleitkomma	Das Tag ist „true“, wenn der Wert nicht Null ist, oder „false“, wenn er Null ist. Der Wert „true“ wird als 32-Bit-Gleitkommawert 1 geschrieben und der Wert „false“ als Null. Bei einem zugeordneten Array entspricht jedes Array-Element einem einzelnen Kommunikationsdatenelement. Diese Einstellung ist für Kommunikationsdaten mit einer Größe von genau 32 Bit verfügbar.
Bit-Array Little-Endian	Ein einzelnes Bit wird aus den Daten extrahiert. Bei einem einzelnen Element wählt das Feld „Bit Number“ (Bitnummer) das Bit aus, wobei „Bit 0“ das niedrigstwertige Bit ist. Bei einem Array ist jedes Element ein einzelnes Bit, d. h. die Bits sind sozusagen in den Datenelementen „verpackt“. Das erste Element des Arrays ist das niedrigstwertige Bit, das zweite Element ist das Bit mit dem nächsthöheren Stellenwert usw. Ein aus 8 Elementen bestehendes Array, das einem Byte-Datentyp in einem SPS zugeordnet ist, liest daher alle 8 Bits aus einem einzelnen Register.
Bit-Array Big-Endian	Hier gilt das Gleiche wie oben, nur sind die Bits umgekehrt. Wenn im Feld „Bit Number“ (Bitnummer) „Bit 0“ steht, wird auf das höchstwertige Bit zugegriffen. Das erste Element eines Arrays wird aus dem höchstwertigen Bit bezogen; danach steigt die Reihenfolge ab.

- Wenn Sie einen der Bit-Array-Modi verwenden, lesen Sie den nachstehenden Abschnitt über den Schreibmodus bezüglich der Gefahr von Schreibzugriffen auf einzelne Bits innerhalb größerer Datentypen, wenn andere Bits im gleichen Wert vom externen Gerät geändert werden können.
- Die Eigenschaft *Bit Number* (Bitnummer) extrahiert aus Multi-Bit-Datenelementen ein einzelnes Bit für zugeordnete Nicht-Array-Tags. Die Eigenschaft wird nicht für andere Konfigurationen verwendet.
- Die Eigenschaft *Manipulation* definiert die Transformation, die beim Lesen von Daten nach der Logik „Treat As“ (Behandeln als) bzw. beim Schreiben von Daten vor der Logik „Treat As“ (Behandeln als) auf den Tag-Zustand angewendet wird. Die einzig verfügbare Option ist, den Zustand des Tags umzukehren. Mit einem einzelnen Bitwert lässt sich sonst wenig anfangen.
- Mit der Eigenschaft *Access* (Zugriff) wird bei zugeordneten Tags definiert, welche Kommunikationsvorgänge zulässig sein sollen. Interne Tags sind immer auf Lese- und Schreibzugriff gesetzt, und Ausdrucks-Tags sind immer schreibgeschützt.
- Die Eigenschaft *Write Mode* (Schreibmodus) wird verwendet, wenn ein Markierungstag einem Bit innerhalb eines größeren Datentyps zugeordnet und der Schreibzugriff aktiviert ist. Wenn Sie Crimson anweisen, Daten aus zwischengespeicherten Daten zu schreiben, wird der letzte bekannte Wert (entweder aus einem Lese- oder dem letzten Schreibvorgang) für die übrigen Bits im größeren Datentyp verwendet. Wenn Sie Crimson anweisen, einen Lesen-Ändern-Schreiben-Vorgang auszuführen, wird der größere Datentyp gelesen, das ausgewählte Bit aktualisiert und der neue Wert zurückgeschrieben.

Dieser Vorgang kann in einer einzigen Kommunikationstransaktion auftreten, wenn der zugrunde liegende Treiber dies unterstützt, wird jedoch normalerweise in drei einzelnen Vorgängen ausgeführt. Es besteht weiterhin die Möglichkeit, dass die Daten in den anderen Bits des zugrunde liegenden Typs während des Vorgangs geändert werden, sodass ungültige Werte zurückgeschrieben werden. Sie sollten daher Schreibvorgänge in einzelne Bits in größeren Datentypen vermeiden, wenn das externe Gerät weitere Bits am gleichen Speicherort ändern kann.

- Die Eigenschaft *Read Mode* (Lesemodus) wird nur für Array-Tags verwendet. Sie gibt die Elemente an, die gelesen werden sollen, wenn das Array referenziert wird. Folgende Einstellungen können verwendet werden:

Lesemodus	Beschreibung
Gesamtes Array	Alle Elemente im Array werden gelesen, wenn das Array referenziert wird, und der Zugriff wird blockiert, bis der Lesevorgang abgeschlossen ist. Dadurch wird zwar sichergestellt, dass Daten verfügbar sind, aber die Durchführung ist am langsamsten.
Manueller Modus	Das Array wird erst gelesen, wenn ein Aufruf an die Funktion <code>ReadData</code> erfolgt, um ein einmaliges manuelles Update zu erzwingen.
Auf Anforderung	Array-Elemente werden gelesen, sobald sie referenziert werden. Dadurch wird die schnellste Durchführung erzielt, aber möglicherweise werden veraltete Daten zurückgegeben, wenn ein Element zum ersten Mal aufgerufen wird.
N auf beiden Seiten	Die Operation „On Demand“ (Auf Anforderung) wird zwar verwendet, aber N Register auf beiden Seiten des referenzierten Registers werden ebenfalls gelesen, wodurch angrenzende Daten schneller verfügbar gemacht werden.

- Mit der Eigenschaft *Storage* (Speicher) wird angegeben, ob das Tag bei einem Aus- und Einschaltvorgang des Zielgeräts beibehalten wird. Sie wird in der Regel für interne Tags verwendet, aber auch die Werte von zugeordneten Nur-schreiben-Tags können beibehalten werden.

5.11.1.2 Datensimulation

- Die Eigenschaft *Simulate as* (Simulieren als) definiert den angenommenen Wert, der bei der Arbeit im Seiteneditor für das Tag verwendet werden soll. Durch die Eingabe eines sinnvollen Werts stimmt die Darstellung der Seite stärker mit ihrem tatsächlichen Erscheinungsbild überein. Dieser Wert wird auch vom Zielgerät als Standardwert des Tags verwendet, wenn die Kommunikation global deaktiviert ist.

5.11.1.3 Datenaktivitäten

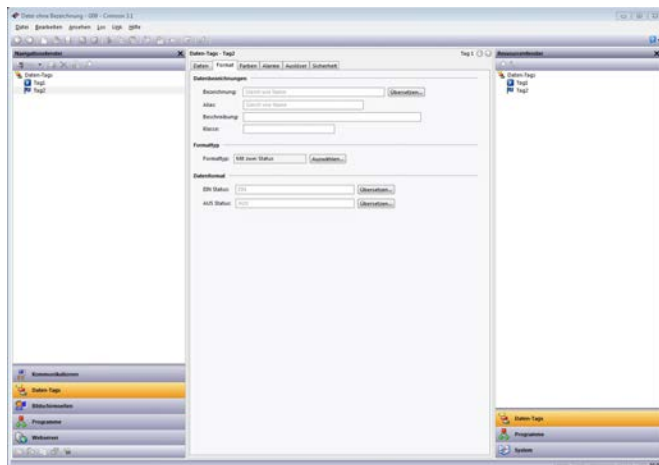
- Die Eigenschaft *On Write* (Bei Schreiben) definiert eine Aktion, die bei einer Änderung des Tags aufgerufen werden soll. Die Systemvariable *Data* (Daten) enthält den neuen Datenwert, wenn der Schreibvorgang erfolgt und wenn die Aktion ausgeführt wird. Die Verwendung der Eigenschaften „On Write“ (Bei Schreiben) wird später in diesem Kapitel behandelt.

5.11.1.4 Daten Sollwert

- Die Eigenschaft *Use Setpoint* (Sollwert verwenden) dient zum Aktivieren oder Deaktivieren eines Sollwerts für dieses Tag.
- Die Eigenschaft *SP Value* (SP-Wert) definiert einen Ausdruck oder ein anderes Tag, dem dieses Tag nominell folgen soll. Dieser Sollwert kann dann in Alarmen oder Primitiven verwendet werden, um verschiedene Funktionen zu implementieren.

5.11.2 Formateigenschaften

Für ein Markierungstag werden auf der Registerkarte „Format“ die folgenden Eigenschaften angezeigt:



5.11.2.1 Datenbezeichnungen

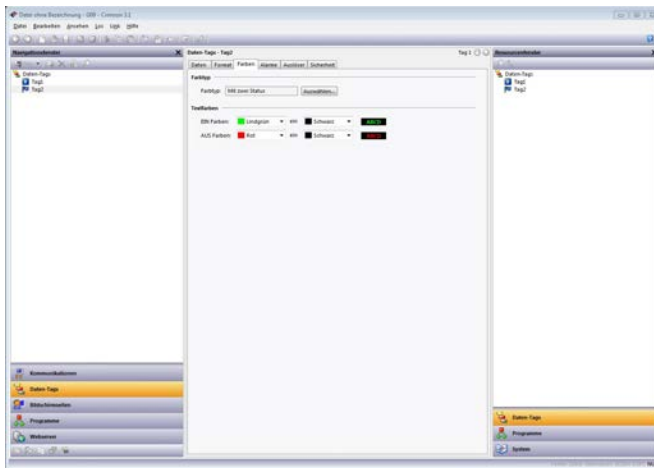
- Die Eigenschaft *Label* (Bezeichnung) wurde bereits unter „Tag-Attribute“ erörtert.
- Die Eigenschaft *Description* (Beschreibung) wurde bereits unter „Tag-Attribute“ erörtert.
- Die Eigenschaft *Class* (Klasse) ist für zukünftige Erweiterungen reserviert.

5.11.2.2 Formattyp

- Die Eigenschaft *Format Type* (Formattyp) wählt das Format für dieses Tag aus. Standardmäßig wird ein Format mit zwei Zuständen verwendet; dies kann jedoch durch ein verknüpftes Format ersetzt werden. Die verschiedenen Arten von Formaten sowie weitere Eigenschaften, die je nach ausgewähltem Formattyp angezeigt werden können, werden in einem späteren Kapitel ausführlich beschrieben.

5.11.3 Farbeigenschaften

Für ein numerisches Tag werden auf der Registerkarte „Colors“ (Farben) die folgenden Eigenschaften angezeigt:

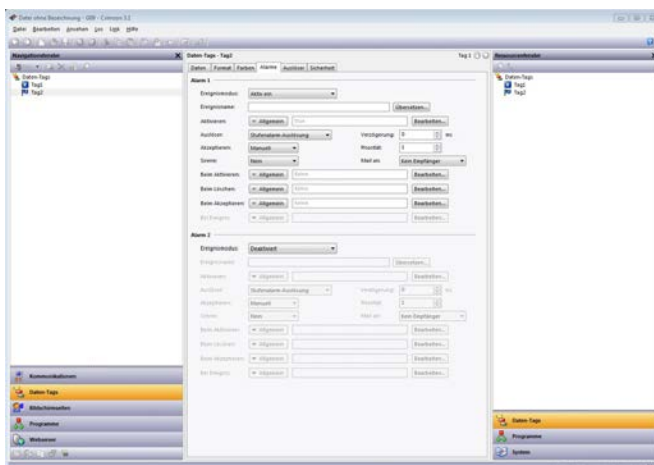


5.11.3.1 Farbtyp

- Die Eigenschaft *Color Type* (Farbtyp) definiert die Farbe für dieses Tag. Standardmäßig wird eine Färbung mit zwei Zuständen ausgewählt, die aber durch eine Färbung des Typs „General“ (Allgemein), „Linked“ (Verknüpft) oder „Fixed“ (Fixiert) ersetzt werden kann. Die verschiedenen Färbungen, sowie weitere Eigenschaften, die je nach ausgewählter Option angezeigt werden können, werden in einem späteren Kapitel ausführlich beschrieben.

5.11.4 Alarmeigenschaften

Für ein Markierungstag werden auf der Registerkarte „Alarms“ (Alarme) die folgenden Eigenschaften angezeigt:



5.11.4.1 Für jeden Alarm

- Mit der Eigenschaft *Event Mode* (Ereignismodus) wird die Logik angegeben, die zur Aktivierung des Alarms verwendet wird. Die verfügbaren Modi sind in der nachstehende Tabelle aufgeführt.

Modus	Alarm wird aktiviert, wenn Folgendes zutrifft:
Aktiv Ein	Das Tag ist „true“.
Aktiv Aus	Das Tag ist „false“.
Statusänderung	Das Tag wurde verändert.

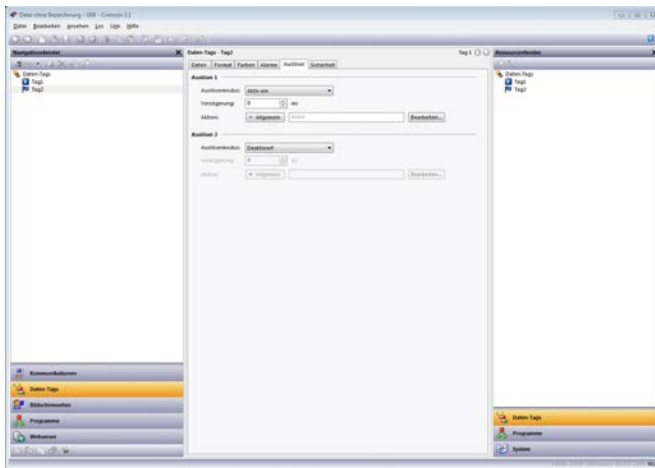
Die folgenden Modi sind nur verfügbar, wenn ein Sollwert definiert ist:

Modus	Alarm wird aktiviert, wenn Folgendes zutrifft:
Entspricht nicht SP	Das Tag entspricht nicht seinem Sollwert.
Aus, wenn SP Ein	Das Tag reagiert nicht auf einen EIN-Sollwert.
Ein, wenn SP Aus	Das Tag reagiert nicht auf einen AUS-Sollwert.
Entspricht SP	Das Tag entspricht seinem Sollwert.

- Die Eigenschaft *Event Name* (Ereignisname) definiert den Namen, der entweder im Alarmbetrachter oder im Ereignisprotokoll angezeigt wird. Crimson schlägt einen Standardnamen vor, der auf der Bezeichnung des Tags und dem ausgewählten Ereignismodus beruht.
- Die Eigenschaft *Enable* (Aktivieren) definiert einen Ausdruck, der den Alarm aktiviert oder deaktiviert. Bei einem Wert von nicht Null oder einem leeren Wert wird der Alarm aktiviert. Bei einem Nullwert wird der Alarm deaktiviert.
- Mit der Eigenschaft *Trigger* wird angegeben, ob der Alarm flankengesteuert oder pegelabhängig ausgelöst werden soll. Im ersteren Fall wird der Alarm ausgelöst, wenn die durch den Ereignismodus definierte Bedingung zum ersten Mal wahr wird. Im zweiten Fall verbleibt der Alarm im aktiven Zustand, solange die Bedingung besteht. Mit dieser Eigenschaft können Sie auch angeben, dass dieser Alarm nur als Ereignis verwendet werden soll. In diesem Fall wird der Alarm zwar flankengesteuert ausgelöst, führt aber nicht zu einer Alarmbedingung. Stattdessen wird ein Ereignis im internen Speicher und optional auf der Speicherkarte protokolliert.
- Mit der Eigenschaft *Delay* (Verzögerung) wird angegeben, wie lange die Alarmbedingung bestehen muss, damit der Alarm aktiv wird. Im Falle eines flankengesteuerten Alarms oder Ereignisses gibt diese Eigenschaft auch an, wie lange die Alarmbedingung nicht mehr bestehen muss, damit nachfolgende Reaktivierungen zu einem weiteren Alarm führen. Angenommen, ein Alarm soll gemäß Einstellung aktiviert werden, wenn ein Geschwindigkeitsschalter anzeigt, dass ein Motor nicht läuft, obwohl der Start dieses Motors angefordert wurde. In diesem Fall kann dem Motor mithilfe dieser Eigenschaft eine Anlaufzeit zugewiesen werden, bevor der Alarm aktiviert wird.
- Mit der Eigenschaft *Accept* (Akzeptieren) wird angegeben, ob der Benutzer einen Alarm explizit akzeptieren muss, damit er nicht mehr angezeigt wird. Flankengesteuerte Alarme müssen immer manuell akzeptiert werden.
- Die Eigenschaft *Priority* (Priorität) dient zur Steuerung der Reihenfolge, in der Alarme im Alarmbetrachter von Crimson angezeigt werden. Je niedriger der numerische Wert im Feld „Priority“ (Priorität), desto weiter oben wird der Alarm angezeigt.
- Mit der Eigenschaft *Siren* (Sirene) wird angegeben, ob die Aktivierung dieses Alarms auch den Signalgeber des Zielgeräts aktivieren soll. Während der akustische Signalgeber aktiv ist, blinkt auch das Display des Bedienfelds, um auf die Alarmbedingung aufmerksam zu machen.
- Die Eigenschaft *Mail To* ((Senden an) gibt den Eintrag im E-Mail-Adressbuch an, an den eine Nachricht gesendet werden soll, wenn dieser Alarm aktiviert wird. Informationen zur E-Mail-Konfiguration finden Sie im Kapitel „Verwenden von Diensten“.
- Mit den Eigenschaften *On Accept* (Bei Akzeptieren), *On Active* (Bei Aktiv), *On Clear* (Bei Löschen) und *On Event* (Bei Ereignis) werden Aktionen angegeben, die bei der Änderung des Zustands ausgeführt werden sollen. Welche Aktionen verfügbar sind, richtet sich nach dem Auslösemodus des Alarms und der unter „Accept“ (Akzeptieren) gewählten Option.

5.11.5 Auslösereigenschaften

Für ein Markierungstag werden auf der Registerkarte „Trigger“ (Auslöser) die folgenden Eigenschaften angezeigt:



5.11.5.1 Für jeden Auslöser

- Die Eigenschaft *Trigger Mode* (Auslösemodus) stimmt mit der Beschreibung für die Registerkarte „Alarms“ (Alarmer) überein.
- Die Eigenschaft *Delay* (Verzögerung) stimmt mit der Beschreibung für die Registerkarte „Alarms“ (Alarmer) überein.
- Mit der Eigenschaft *Action* (Aktion) wird angegeben, welche Aktion ausgeführt werden soll, wenn der Auslöser aktiviert ist. Eine Beschreibung der Syntax, die zum Definieren der verfügbaren Aktionen verwendet wird, finden Sie im Kapitel „Schreiben von Aktionen“.

5.11.6 Sicherheitseigenschaften

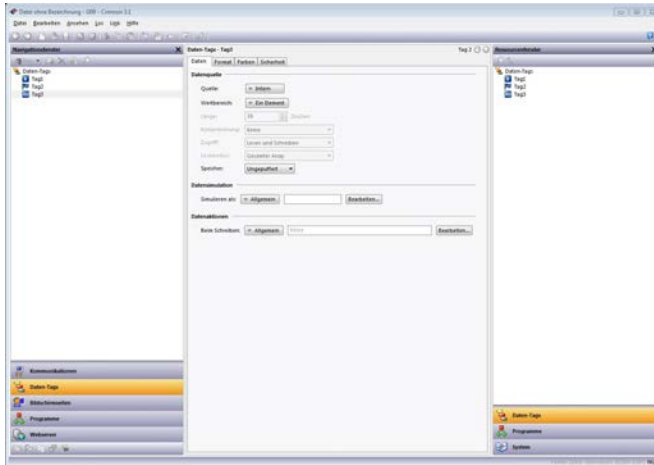
Informationen zu Sicherheitsbeschreibungen finden Sie im Kapitel „Verwenden von Sicherheit“.

5.12 String-Tags

Ein String-Tag stellt eine oder mehrere Zeichenfolgen von Unicode-Zeichen dar. Crimson arbeitet mit Unicode, kann aber auch Zeichenfolgen aus 8-Bit-Quellen lesen und schreiben. Zugeordnete String-Tags unterstützen verschiedene Codierungen, sodass ein oder mehrere Zeichen aus einem Register extrahiert werden können.

5.12.1 Dateneigenschaften

Für ein String-Tag werden auf der Registerkarte „Data“ (Daten) die folgenden Eigenschaften angezeigt:



5.12.1.1 Datenquelle

- Die Eigenschaft *Source* (Quelle) gibt an, woher die Daten des Tags stammen. Die Standardeinstellung führt zu einem internen Tag, aber in der Dropdown-Liste kann auch ein allgemeiner Ausdruck, ein anderes Daten-Tag oder ein Element von einem externen Gerät ausgewählt werden.
- Die Eigenschaft *Extent* (Umfang) dient zur Wahl zwischen einem einzelnen Element-Tag oder einem Array. Wenn Sie ein Array auswählen, müssen Sie die erforderliche Anzahl von Elementen eingeben. Arrays sind bei Tags, deren Quelle ein Ausdruck ist, nicht zulässig. Bei zugeordneten Tags hängt die genaue Anzahl der zu lesenden Register vom Typ der Register ab, denen das Tag zugeordnet ist, sowie von der Länge und der Einstellung „Packing“ (Packung).
- Die Eigenschaft *Length* (Länge) definiert die Länge der Zeichenfolge. Für nicht remanente interne Zeichenfolgen ist keine Länge definiert, da sie eine Zeichenfolge beliebiger angemessener Länge aufnehmen können.
- Mit der Eigenschaft *Packing* (Packung) wird bei zugeordneten Tags festgelegt, wie der Unicode-Zeichenfolgenwert aus den Rohdaten der Kommunikation und umgekehrt abgeleitet werden soll. Je nach dem zugrunde liegenden Datentyp sind die folgenden Einstellungen verfügbar:

Packung	Ergebnis
Keine	Jedes Kommunikationsdatenelement wird zur Bereitstellung eines einzelnen Zeichens für die Zeichenfolge verwendet. 8-Bit-Werte werden als ASCII behandelt und Werte mit 16 oder mehr Bits als Unicode.
ASCII Big-Endian	Jede 8-Bit-Einheit im Datenelement wird verwendet, um ein einzelnes ASCII-Zeichen bereitzustellen, wobei die höchstwertigen 8 Bits für das erste Zeichen verwendet werden. Nur verfügbar für Datenelemente mit einer Größe ab 16 Bit.
ASCII Little-Endian	Jede 8-Bit-Einheit im Datenelement wird verwendet, um ein einzelnes ASCII-Zeichen bereitzustellen, wobei die am niedrigstwertigen 8 Bits für das erste Zeichen verwendet werden. Nur verfügbar für Datenelemente mit einer Größe ab 16 Bit.
Unicode Big-Endian	Jede 16-Bit-Einheit im Datenelement wird verwendet, um ein einzelnes Unicode-Zeichen bereitzustellen, wobei die höchstwertigen 16 Bits für das erste Zeichen verwendet werden. Nur verfügbar für Datenelemente mit einer Größe von 32 Bit.
Unicode Little-Endian	Jede 16-Bit-Einheit im Datenelement wird verwendet, um ein einzelnes Unicode-Zeichen bereitzustellen, wobei die niedrigstwertigen 16 Bits für das erste Zeichen verwendet werden. Nur verfügbar für Datenelemente mit einer Größe von 32 Bit.
Hex String Little-Endian	Jede 4-Bit-Einheit im Datenelement wird verwendet, um ein einzelnes Hex-Zeichen im Bereich „0-9“ und „A-F“ bereitzustellen, wobei die niedrigstwertigen 4 Bits zuerst verwendet werden. Schreibvorgänge werden in Zeichenfolgen mit dieser Packungsmethode nicht unterstützt.
Hex String Big-Endian	Jede 4-Bit-Einheit im Datenelement wird verwendet, um ein einzelnes Hexadezimalzeichen im Bereich „0-9“ und „A-F“ bereitzustellen, wobei die höchstwertigen 4 Bits zuerst verwendet werden. Schreibvorgänge werden in Zeichenfolgen mit dieser Packungsmethode nicht unterstützt.

- Mit der Eigenschaft *Access* (Zugriff) wird bei zugeordneten Tags definiert, welche Kommunikationsvorgänge zulässig sein sollen. Interne Tags sind immer auf Lese- und Schreibzugriff gesetzt, und Ausdrucks-Tags sind immer schreibgeschützt.
- Die Eigenschaft *Read Mode* (Lesemodus) wird nur für Array-Tags verwendet. Sie gibt die Elemente an, die gelesen werden sollen, wenn das Array referenziert wird. Folgende Einstellungen können verwendet werden:

Lesemodus	Beschreibung
Gesamtes Array	Alle Elemente im Array werden gelesen, wenn das Array referenziert wird, und der Zugriff wird blockiert, bis der Lesevorgang abgeschlossen ist. Dadurch wird zwar sichergestellt, dass Daten verfügbar sind, aber die Durchführung ist am langsamsten.
Manueller Modus	Das Array wird erst gelesen, wenn ein Aufruf an die Funktion <code>ReadData</code> erfolgt, um ein einmaliges manuelles Update zu erzwingen.
Auf Anforderung	Array-Elemente werden gelesen, sobald sie referenziert werden. Dadurch wird die schnellste Durchführung erzielt, aber möglicherweise werden veraltete Daten zurückgegeben, wenn ein Element zum ersten Mal aufgerufen wird.
N auf beiden Seiten	Die Operation „On Demand“ (Auf Anforderung) wird zwar verwendet, aber N Register auf beiden Seiten des referenzierten Registers werden ebenfalls gelesen, wodurch angrenzende Daten schneller verfügbar gemacht werden.

- Mit der Eigenschaft *Storage* (Speicher) wird angegeben, ob das Tag bei einem Aus- und Einschaltvorgang des Zielgeräts beibehalten wird. Sie wird in der Regel für interne Tags verwendet, aber auch die Werte von zugeordneten Nur-schreiben-Tags können beibehalten werden.

5.12.1.2 Datensimulation

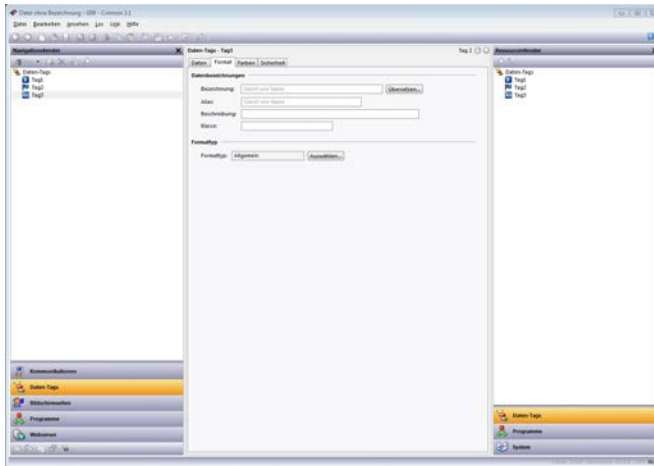
- Die Eigenschaft *Simulate as* (Simulieren als) definiert den angenommenen Wert, der bei der Arbeit im Seiteneditor für das Tag verwendet werden soll. Durch die Eingabe eines sinnvollen Werts stimmt die Darstellung der Seite stärker mit ihrem tatsächlichen Erscheinungsbild überein. Dieser Wert wird auch vom Zielgerät als Standardwert des Tags verwendet, wenn die Kommunikation global deaktiviert ist.

5.12.1.3 Datenaktivitäten

- Die Eigenschaft *On Write* (Bei Schreiben) definiert eine Aktion, die bei einer Änderung des Tags aufgerufen werden soll. Die Systemvariable *Data* (Daten) enthält den neuen Datenwert, wenn der Schreibvorgang erfolgt und wenn die Aktion ausgeführt wird. Die Verwendung der Eigenschaften „On Write“ (Bei Schreiben) wird später in diesem Kapitel behandelt.

5.12.2 Formateigenschaften

Für ein String-Tag werden auf der Registerkarte „Format“ die folgenden Eigenschaften angezeigt:



5.12.2.1 Datenbezeichnungen

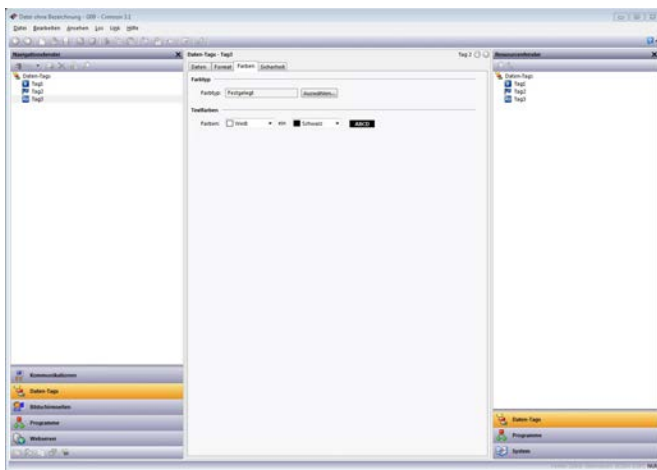
- Die Eigenschaft *Label* (Bezeichnung) wurde bereits unter „Tag-Attribute“ erörtert.
- Die Eigenschaft *Description* (Beschreibung) wurde bereits unter „Tag-Attribute“ erörtert.
- Die Eigenschaft *Class* (Klasse) ist für zukünftige Erweiterungen reserviert.

5.12.2.2 Formattyp

- Die Eigenschaft *Format Type* (Formattyp) wählt das Format für dieses Tag aus. Standardmäßig wird ein Zeichenfolgenformat verwendet, aber es kann durch ein Format des Typs „General“ (Allgemein) oder „Linked“ (Verknüpft) ersetzt werden. Die verschiedenen Arten von Formaten sowie weitere Eigenschaften, die je nach ausgewähltem Formattyp angezeigt werden können, werden in einem späteren Kapitel ausführlich beschrieben.

5.12.3 Farbeigenschaften

Für ein String-Tag werden auf der Registerkarte „Colors“ (Farben) die folgenden Eigenschaften angezeigt:



5.12.3.1 Farbtyp

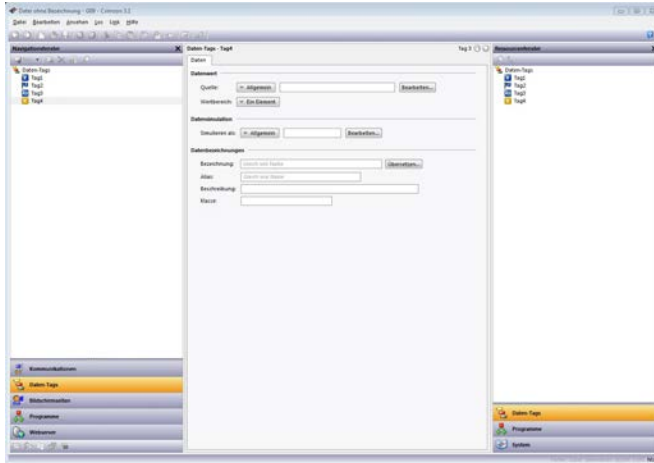
- Die Eigenschaft *Color Type* (Farbtyp) definiert die Farbe für dieses Tag. Standardmäßig ist eine Färbung des Typs „Fixed“ (Fixiert) ausgewählt, sie kann jedoch durch eine Färbung des Typs „General“ (Allgemein) oder „Linked“ (Verknüpft) ersetzt werden. Die verschiedenen Färbungen sowie weitere Eigenschaften, die je nach ausgewählter Option angezeigt werden können, werden in einem späteren Kapitel ausführlich beschrieben.

5.12.4 Sicherheitseigenschaften

Informationen zu Sicherheitsbeschreibungen finden Sie im Kapitel „Verwenden von Sicherheit“.

5.13 Basis-Tags

Basis-Tags werden zur Darstellung von Konstanten oder Ausdrücken verwendet:



5.13.1 Datenwert

Die Eigenschaft *Data Value* (Datenwert) definiert den Wert des Tags. Sie muss ein Ausdruck sein. Das Tag selbst übernimmt den Datentyp des verwendeten Ausdrucks.

5.13.1.1 Datensimulation

- Die Eigenschaft *Simulate as* (Simulieren als) definiert einen Wert, der beim Bearbeiten von Anzeigeseiten als Standard für das Tag verwendet wird. Durch die Eingabe eines sinnvollen Werts stimmt die Darstellung der Seite stärker mit ihrem tatsächlichen Erscheinungsbild überein. Dieser Wert wird auch vom Zielgerät als Standardwert des Tags verwendet, wenn die Kommunikation global deaktiviert ist.

5.13.1.2 Datenbezeichnungen

- Die Eigenschaft *Label* (Bezeichnung) wurde bereits unter „Tag-Attribute“ erörtert.
- Die Eigenschaft *Description* (Beschreibung) wurde bereits unter „Tag-Attribute“ erörtert.
- Die Eigenschaft *Class* (Klasse) ist für zukünftige Erweiterungen reserviert.

5.14 Weiterführende Themen

5.14.1 Array-Eigenschaften

Viele der Eigenschaften von Array-Tags können variabel gemacht werden, ausgehend vom tatsächlichen Element des Arrays, das referenziert wird. Hierzu wird eine Systemeigenschaft namens i bei der Bewertung dieser Eigenschaften auf den Elementindex gesetzt. Zum Beispiel könnte die Eigenschaft „Label“ (Bezeichnung) auf `=\"Element \" + AsText(i+1)` gesetzt werden, um die Array-Elemente „Element 1“, „Element 2“ usw. zu kennzeichnen.

Diese Funktion kann mit den folgenden Eigenschaften verwendet werden:

- Bezeichnung des Tags
- Skalierungswerte des Tags
- Sollwert des Tags
- Grenzwerte des Tags
- Tag-Eigenschaft „On Write“ (Bei Schreiben)
- Ereignisbezeichnungen des Tags
- Ereignis- und Auslösewerte sowie Hysterese-Einstellungen des Tags
- Auslöseaktionen des Tags

Beachten Sie, dass Auslöser und Ereignisse für jedes Element des Arrays, für das sie konfiguriert sind, separat ausgewertet werden. Dies bedeutet, dass mehrere Ereignisse oder Auslöser gleichzeitig erstellt werden können. Die einzige Einschränkung dieser Funktion ist, dass Alarmer und Ereignisse nur bei den ersten 256 Elementen des Arrays funktionieren. Auslöser funktionieren bei allen Elementen, unabhängig von der Größe des Arrays.

5.14.2 Tag-Datenfluss

Wie Sie sicher bemerkt haben, gibt es insbesondere bei numerischen Tags eine Reihe von Datentransformationen, die zwischen den Kommunikationsdaten und dem tatsächlich von Crimson verwendeten Wert stattfinden. Diese können so konfiguriert werden, dass sie nahezu jede Art von Daten auf jede erdenkliche Weise verarbeiten können. Im Folgenden soll aber auf die Besonderheiten bei numerischen Tags eingegangen werden.

5.14.2.1 Lesevorgang bei numerischen Tags

Wenn Daten aus einem Gerät gelesen werden, werden die folgenden Schritte ausgeführt:

- Der Kommunikationstreiber liest ausgehend von der Adresseinstellung, die für die Quelle des Tags definiert wurde, einen Wert. Je nach Art der Adresse kann der Treiber mehrere Register kombinieren, um den Datenwert zu erstellen. Beim Lesen eines einzelnen „Word as Long“-Werts (Wort als Lang) werden beispielsweise zwei Register gelesen und kombiniert, wobei sich der Treiber auf sein Wissen über die Wortreihenfolge des Geräts stützt.
- Die Kommunikationsdaten werden dann der Eigenschaft „Manipulation“ des betreffenden Tags entsprechend geändert. Bei diesen Vorgängen werden die Daten auf Bit- oder Byte-Ebene geändert. Damit wird in der Regel Treiberinkompatibilitäten oder anderen Situationen Rechnung getragen, in denen die Daten nicht in dem vom Kommunikationstreiber normalerweise erwarteten Format vorliegen.
- Die manipulierten Daten werden dann in Verbindung mit der Eigenschaft „Treat As“ (Behandeln als) interpretiert und entweder als 32-Bit-Ganzzahl oder 32-Bit-Gleitkommawert mit einfacher Genauigkeit betrachtet. Datenelemente, die kleiner sind als 32 Bit, werden je nach ihrer Konfiguration entweder mit

Nullen oder Vorzeichen erweitert. Wenn keine Skalierung definiert ist, bestimmt das Ergebnis dieses Schritts den endgültigen Wert und Datentyp des Tags.

- Wenn die Skalierung definiert ist, werden die interpretierten Daten gemäß dem definierten Bereich und Umfang des Tags skaliert. Das Ergebnis der Skalierung kann einem anderen Typ angehören als die interpretierten Daten, d. h. ein Gleitkommawert kann auf eine Ganzzahl skaliert werden oder umgekehrt. Unter der Voraussetzung, dass eine Skalierung definiert ist, bestimmt das Ergebnis dieses Schritts den endgültigen Wert und Datentyp des Tags.

5.14.2.2 Schreibvorgang bei numerischen Tags

Wenn Daten auf ein Gerät geschrieben werden, werden die folgenden Schritte ausgeführt:

- Wenn eine Skalierung definiert ist, werden Bereich und Umfang umgekehrt. Dabei werden die Daten in einen nicht skalierten Wert zurück konvertiert, dessen Datentyp durch die Eigenschaft „Treat As“ (Behandeln als) definiert wird.
- Wenn die nicht skalierten Daten größer sind als die Kommunikationsdaten, werden die höherwertigen Bits entfernt. Das Resultat ist eine bereinigte Version der Daten, die für den nächsten Schritt geeignet ist.
- Die bereinigten Daten werden dann gemäß der Eigenschaft „Manipulation“ geändert. Dabei wird die obige Transformation umgekehrt, und es werden Kommunikationsdaten erzeugt.
- Anschließend schreibt der Kommunikationstreiber die Kommunikationsdaten je nach der Art der Adresse in ein oder mehrere Register im Zielgerät.

5.14.3 Verwenden von Bei Schreiben

Die Eigenschaft „On Write“ (Bei Schreiben) eines Tags enthält eine Aktion, die ausgeführt wird, wenn am Tag eine Änderung vorgenommen wird. Während die Aktion ausgeführt wird, wird eine Systemeigenschaft namens `Data` auf den neuen Wert gesetzt, damit die neuen Daten geprüft werden können.

Für diese Funktion gibt es drei typische Verwendungszwecke:

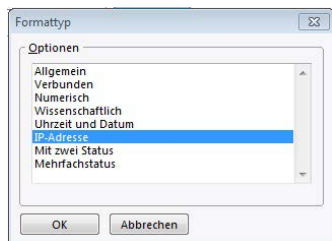
- Für reguläre Lesen-und-Schreiben-Tags kann eine Eigenschaft „On Write“ (Bei Schreiben) definiert werden, die es ermöglicht, bei Bedarf eine Aktion auszuführen. Zum Beispiel muss eine Datenbank den Wert eines Tags möglicherweise in zwei Formaten speichern, also das ursprüngliche Tag-Format und die transformierte Version. Hierzu stehen verschiedene Methoden zur Auswahl. Eine Methode besteht darin, mit der Eigenschaft „On Write“ (Bei Schreiben) den Schreibvorgang zu erfassen und dann ein Programm zum Berechnen und Speichern der transformierten Version auszuführen.
- Schreibgeschützte Tags können durch Festlegen der Eigenschaft „On Write“ (Bei Schreiben) beschreibbar gemacht werden. Das erscheint widersinnig; doch angenommen, eine PID-Schleife verfügt über die Eigenschaft „Read-Only“ (Schreibgeschützt), um die aktuelle Ausgangsleistung anzugeben, und die Eigenschaft „Read/Write“ (Lesen/Schreiben), um die manuelle Ausgangsleistung anzugeben. Sie könnten Anzeigefelder definieren, um die Dateneingabe für die Ausgangsleistung im manuellen Modus zu ermöglichen, und diese mit der Eigenschaft „On Write“ (Bei Schreiben) erfassen, wodurch die Werte in die manuelle Ausgangsleistung geschrieben werden.
- Zur Implementierung komplexer Transformationen können Sie ein Ausdrucks-Tag zur Durchführung der Vorwärtstransformation sowie eine „On Write“ (Bei Schreiben-)Aktion zur Durchführung der Umkehrung definieren. Ein Tag könnte beispielsweise auf `Sqrt([40001])` gesetzt werden, um die Quadratwurzel eines Wertes in einem Modbus-SPS zu berechnen. Da dies ein Ausdrucks-Tag ist, ist es definitionsgemäß schreibgeschützt. Schreibvorgänge können jedoch zugelassen werden, indem ein „On Write“ (Bei Schreiben) gleich `[40001] = Data*Data` gesetzt und dadurch die Quadratwurzelberechnung umgekehrt wird.

Kapitel 6 Verwenden von Formaten

Für numerische Tags stehen verschiedene Datenformate zur Auswahl. Markierungs-Tags haben immer das Format „Two-State“ (Zweifachstatus), und String-Tags haben immer das Format „General“ (Allgemein). Jeder Formattyp konvertiert einen Datenwert in eine bzw. aus einer Textzeichenfolge.

6.1 Formattypen

Folgende Formate werden unterstützt:



- Das Format *General* (Allgemein) bietet eine einfache Formatierung von Werten. Numerische Werte werden in vorzeichenbehaftete Dezimalwerte konvertiert, und Zeichenfolgen werden ohne weitere Verarbeitung übergeben. Das Format „General“ (Allgemein) hat keine Konfigurationseigenschaften und ist das Standardformat für String-Tags. Es wird auch vorbehaltlos von Basis-Tags verwendet.
- Das Format *Linked* (Verbunden) verwendet das Datenformat eines anderen Tags zum Formatieren desjenigen Tags, das Sie konfigurieren. Sie können damit Formatvorlagen erstellen und diese dann auf viele Tags in derselben Datenbank anwenden. Dadurch lassen sich Wiederholungen vermeiden, und Einstellungen wie Einheiten oder die Anzahl von Dezimalstellen lassen sich leichter anpassen.
- Das Format *Numeric* (Numerisch) konvertiert einen Gleitkomma- oder Ganzzahlwert in eine Zeichenfolge. Dabei wird eine bestimmte Zahlenbasis verwendet und die erforderliche Anzahl von Ziffern vor und nach dem Dezimalkomma ausgewählt. Das Format kann dem Wert auch eine Präfixzeichenfolge und eine Einheitenzeichenfolge hinzufügen sowie Werte mit und ohne Vorzeichen verarbeiten.
- Das Format *Scientific* (Wissenschaftlich) konvertiert einen Gleitkomma- oder Ganzzahlwert in ein Exponentialformat. Dabei wird die erforderliche Anzahl von Ziffern nach dem Dezimalkomma ausgewählt. Das Format kann dem Wert auch eine Präfixzeichenfolge und eine Einheitenzeichenfolge hinzufügen.
- Das Format *Time and Date* (Zeit und Datum) behandelt einen Ganzzahlwert als Anzahl von Sekunden, die seit dem 01.01.1997 verstrichen sind. Das Ergebnis kann als Datumswert, Zeitwert oder beides angezeigt werden. Der Wert kann auch als verstrichene Zeit behandelt werden. In diesem Fall kann der Stundenwert eine höhere Zahl als 24 enthalten. Für Zeit und Datum werden Formatierungsoptionen bereitgestellt, die verschiedene internationale Standards unterstützen.
- Das Format *IP Address* (IP-Adresse) zeigt einen Ganzzahlwert und als vier durch Punkte voneinander getrennte Dezimalbytes an. Dadurch kann eine 32-Bit-Nummer ohne weitere Konfiguration als IP-Adresse angezeigt werden.
- Das Format *Two-State* (Zweifachstatus) zeigt einen numerischen Wert als eine von zwei Zeichenfolgen an, je nachdem, ob der Wert Null oder nicht Null ist. Dies ist der Standardformattyp für Markierungs-Tags.
- Das Format *Multi-State* (Mehrfachstatus) vergleicht einen numerischen Wert mit einer Tabelle, die Werte und Zeichenfolgen enthält. Entweder wird die mit einem übereinstimmenden Datenwert verknüpfte Zei-

chenfolge angezeigt, oder das Format kann so konfiguriert werden, dass die letzte Zeichenfolge mit einem Wert angezeigt wird, der nicht höher ist als die mit der Zeichenfolge verknüpften Daten.

- Das Format *String* beschränkt die Länge eines Zeichenfolgewerts bei der Eingabe oder wendet eine Schablone an, die angibt, welche Art von Zeichen an welcher Stelle in der Zeichenfolge eingegeben werden kann. Zum Beispiel kann eine Zeichenfolge als US-amerikanische Telefonnummer formatiert werden. Die Klammern und der Bindestrich werden automatisch eingefügt und müssen nicht in den Zeichenfolgedaten gespeichert werden.

6.2 Format Allgemein

Für das Format „General“ (Allgemein) gibt es keine Eigenschaften.

6.3 Format Verbunden

Das Format „Linked“ (Verbunden) verfügt über die folgenden Eigenschaften:



Daten formatieren

Format wie:

- Mit der Eigenschaft *Format Like* (Format wie) können Sie ein Tag auswählen, dem die Formatierungsinformationen für dieses Tag entnommen werden sollen. Achten Sie darauf, ein Tag des richtigen Datentyps zu verwenden; das Format eines String-Tags muss auf einem anderen String-Tag basieren. Wird diese Anforderung nicht beachtet, werden die Standardformatierungsregeln angewendet.

6.4 Format Numerisch

Das Format „Numeric“ (Numerisch) verfügt über die folgenden Eigenschaften:

6.4.1 Festgelegtes Datenformat

- Die Eigenschaft *Number Base* (Zahlenbasis) definiert die Basis des angezeigten Werts. Die Einstellung „Passcode“ eignet sich für Dezimalzahlen, doch werden die Ziffern durch Sternchen ersetzt. Viele der anderen Optionen werden deaktiviert, wenn ein nicht-dezimaler Modus verwendet wird.
- Die Eigenschaft *Sign Mode* (Vorzeichenmodus) definiert, wie die Daten behandelt werden und wie das Vorzeichen angezeigt wird. Bei dem Wert „Unsigned“ (Kein Vorzeichen) wird der Wert als 32-Bit-Zahl ohne Vorzeichen angezeigt. Dadurch können diese Werte angezeigt und eingegeben werden, auch wenn Crimson keine Rechenoperationen mit Werten ausführen kann, die nicht mit einer vorzeichenbehafteten 32-Bit-Darstellung übereinstimmen. Bei dem Wert „Soft Sign“ (Weiches Vorzeichen) wird negativen Zahlen ein Minuszeichen vorangestellt und positiven Zahlen ein Leerzeichen. Bei dem Wert „Hard Sign“ (Hartes Vorzeichen) wird anstelle des Leerzeichens ein Pluszeichen vorangestellt.
- Die Eigenschaft *Digits Before DP* (Ziffern vor DK) definiert, wie viele Ziffern vor dem Dezimalkomma angezeigt werden sollen. Bei Werten ohne Dezimalstellen ist dies die Gesamtzahl der anzuzeigenden Ziffern, die somit die Größe des Datenfelds steuert.
- Die Eigenschaft *Digits After DP* (Ziffern nach DK) definiert, wie viele Ziffern nach dem Dezimalkomma angezeigt werden sollen. Bei Ganzzahlwerten wird der Dezimalpunkt so in die Ganzzahldarstellung eingefügt, dass beispielsweise „1234“ als „12,34“ angezeigt und eingegeben wird, wenn diese Eigenschaft auf zwei gesetzt wurde. Ein Wert von Null unterdrückt das Dezimalkomma.
- Die Eigenschaft *Lead Character* (Führungszichen) definiert, wie Werte mit führenden Nullen formatiert werden. Führende Nullen können entweder beibehalten, durch Leerzeichen ersetzt oder vollständig entfernt werden. Wenn Sie die Nullen entfernen, flackern die Werte auf dem Bildschirm möglicherweise unangenehm, wenn sich die Anzahl der Ziffern ändert. Dies gilt insbesondere für Werte, die in einem Feld zentriert sind.
- Die Eigenschaft *Group Digits* (Ziffern gruppieren) fügt bei Dezimalzahlen nach jeder dritten Ziffer ein Trennzeichen ein. Bei anderen Zahlenbasen wird ähnlich verfahren.

6.4.2 Dynamisches Datenformat

- Die Eigenschaft *Dynamic DP* (Dynamisches DK) bietet eine Möglichkeit, mit einem Ausdruck die Anzahl der nach dem Dezimalkomma angezeigten Ziffern festzulegen. Der Grenzwert wird dabei von dem Wert vorgegeben, der für *Digits after DP* (Ziffern nach DK) festgelegt wurde. Weitere Details zu dieser Eigenschaft finden Sie in „Fixed Data Format (Festgelegtes Datenformat)“.

6.4.3 Formateinheiten

- Die Eigenschaft *Prefix* (Präfix) definiert eine Zeichenfolge, die vor dem numerischen Wert angezeigt werden soll.
- Die Eigenschaft *Units* (Einheiten) definiert eine Zeichenfolge, die nach dem numerischen Wert angezeigt werden soll.

6.5 Format Wissenschaftlich

Das Format „Scientific“ (Wissenschaftlich) verfügt über die folgenden Eigenschaften:

6.5.1 Festgelegtes Datenformat

- Die Eigenschaft *Mantissa Sign Mode* (Vorzeichenmodus Mantisse) definiert, wie das Vorzeichen der Mantisse angezeigt wird. Bei dem Wert „Soft Sign“ (Weiches Vorzeichen) wird negativen Zahlen ein Minuszeichen vorangestellt und positiven Zahlen ein Leerzeichen. Bei dem Wert „Hard Sign“ (Hartes Vorzeichen) wird anstelle des Leerzeichens ein Pluszeichen vorangestellt.
- Die Eigenschaft *Exponent Sign Mode* (Vorzeichenmodus Exponent) definiert, wie das Vorzeichen des Exponenten angezeigt wird. Bei dem Wert „Soft Sign“ (Weiches Vorzeichen) wird bei negativen Werten ein führendes Minuszeichen und bei positiven Werten nichts angezeigt. Bei dem Wert „Hard Sign“ (Hartes Vorzeichen) wird bei positiven Werten ein führendes Pluszeichen angezeigt.
- Die Eigenschaft *Digits After DP* (Ziffern nach DK) definiert, wie viele Ziffern nach dem Dezimal komma angezeigt werden sollen. Definitionsgemäß steht im wissenschaftlichen Format stets eine Ziffer vor dem Dezimalwert. Ein Wert von Null unterdrückt das Dezimal komma.

6.5.2 Dynamisches Datenformat

- Die Eigenschaft *Dynamic DP* (Dynamisches DK) bietet eine Möglichkeit, mit einem Ausdruck die Anzahl der nach dem Dezimal komma angezeigten Ziffern festzulegen. Der Grenzwert wird dabei von dem Wert vorgegeben, der für *Digits after DP* (Ziffern nach DK) festgelegt wurde. Weitere Details zu dieser Eigenschaft finden Sie in „Fixed Data Format (Festgelegtes Datenformat)“.

6.5.3 Formateinheiten

- Die Eigenschaft *Prefix* (Präfix) definiert eine Zeichenfolge, die vor dem numerischen Wert angezeigt werden soll.
- Die Eigenschaft *Units* (Einheiten) definiert eine Zeichenfolge, die nach dem numerischen Wert angezeigt werden soll.

6.6 Format Uhrzeit und Datum

Das Format „Time and Date“ (Uhrzeit und Datum) verfügt über die folgenden Eigenschaften:

6.6.1 Formatmodus

- Mit der Eigenschaft *Format Mode* (Formatmodus) wird angegeben, ob das Feld die Uhrzeit, das Datum oder beides anzeigen soll. In letzterem Fall wird mit dieser Eigenschaft auch angegeben, in welcher Reihenfolge die beiden Elemente angezeigt werden sollen. Außerdem werden Optionen bereitgestellt, mit denen ein Zeitwert als verstrichener Zeitraum und nicht als Uhrzeit behandelt werden kann, die mit einem Datum kombiniert wurde. Zum Beispiel wird ein Wert von 25,5 Stunden als 25:30 angezeigt, wenn die verstrichene Zeit angegeben werden soll. Im konventionellen Zeitmodus wird 00:30 angezeigt, weil das System davon ausgeht, dass eine Uhrzeit am frühen Morgen des 02.01.1997 gemeint ist.

6.6.2 Zeitformat

- Die Eigenschaft *Time Format* (Zeitformat) gibt an, ob das 12-Stunden- oder 24-Stunden-Zeitformat verwendet werden soll. Wie bei anderen Eigenschaften bewirkt auch hier Übernahme der Einstellung „Locale Default“ (Gebietsschema Standard), dass Crimson anhand der im Bedienfeld ausgewählten Sprache ein geeignetes Format auswählt.
- Die Eigenschaft *Time Separator* (Uhrzeit-Trennzeichen) dient zur Auswahl des Zeichens, das zwischen den Elementen der Zeitanzeige platziert wird. Der Standardwert basiert auf der aktuellen Sprachauswahl, kann aber bei Bedarf überschrieben werden.
- Mit den Eigenschaften *AM Suffix* (AM-Suffix) und *PM Suffix* (PM-Suffix) wird im 12-Stunden-Modus angegeben, ob sich die Zeitangabe auf den Vormittag oder Nachmittag bezieht. Wenn Sie diese Eigenschaft nicht definieren, wird von Crimson ein Standardwert verwendet.
- Mit der Eigenschaft *Show Seconds* (Sekunden anzeigen) wird angegeben, ob die Zeitangabe im Uhrzeitfeld nur Stunden und Minuten oder zusätzlich auch Sekunden umfassen soll.

6.6.3 Datumsformat

- Mit der Eigenschaft *Date Format* (Datumsformat) wird angegeben, in welcher Reihenfolge die verschiedenen Datumselemente (d. h. Tag, Monat und Jahr) angezeigt werden.
- Mit der Eigenschaft *Date Separator* (Datumstrennzeichen) wird das Zeichen ausgewählt, das zwischen den Elementen der Datumsanzeige platziert wird. Der Standardwert basiert auf der aktuellen Sprachauswahl, kann aber bei Bedarf überschrieben werden.
- Mit der Eigenschaft *Show Year* (Jahr anzeigen) wird angegeben, ob das Datumsfeld das Jahr enthalten soll, und falls ja, wie viele Ziffern für dieses Element angezeigt werden sollen.
- Mit der Eigenschaft *Show Month* (Monat anzeigen) wird angegeben, ob die Monatsangabe in Form von Ziffern (d. h. 01 bis 12) oder als Kurzname (d. h. Jan bis Dez) angezeigt werden soll.

6.7 Format IP-Adresse

Für das Format „IP Address“ (IP-Adresse) gibt es keine Eigenschaften.

6.8 Format Zweifachstatus

Das Format „Two-State“ (Zweifachstatus) verfügt über die folgenden Eigenschaften:

Datenformat

EIN Status:	<input type="text" value="EIN"/>	<input type="button" value="Übersetzen..."/>
AUS Status:	<input type="text" value="AUS"/>	<input type="button" value="Übersetzen..."/>

- Die Eigenschaft *ON State* (EIN Status) definiert den Text, der angezeigt werden soll, wenn der Wert nicht Null ist.

Die Eigenschaft *OFF State* (AUS Status) definiert den Text, der angezeigt werden soll, wenn der Wert Null ist.

6.9 Format Mehrfachstatus

Das Format „Multi-State“ (Mehrfachstatus) verfügt über die folgenden Eigenschaften:

Formatsteuerung

Status:

Grenze:

Standard:

Übereinstimmungstyp:

Formatstatus

	Daten	Text	
1:	<input type="text" value="1"/>	<input type="text" value="S1"/>	<input type="button" value="Übersetzen..."/>
2:	<input type="text" value="2"/>	<input type="text" value="S2"/>	<input type="button" value="Übersetzen..."/>
3:	<input type="text" value="3"/>	<input type="text" value="S3"/>	<input type="button" value="Übersetzen..."/>

Formatbefehle

6.9.1 Formatsteuerung

- Die Eigenschaft *States* (Status) definiert, wie viele Status das Mehrfachstatusformat enthält. Maximal 500 Einträge sind möglich. Das Fenster, welches das Format anzeigt, wird aktualisiert, um die erforderliche Anzahl von Daten- und Texteingenschaften anzuzeigen.
- Die Eigenschaft *Limit* (Grenze) definiert, wie viele Status verwendet werden, wenn Daten mit diesem Format abgeglichen werden. Sie kann dynamisch angepasst werden. Die absolute Anzahl von Status hingegen ist statisch definiert. Diese Eigenschaft ist nützlich, wenn die Statusfelder zur Laufzeit ausgefüllt werden, da nicht verwendete Felder bei der Dateneingabe übersprungen werden können.
- Die Eigenschaft *Default* (Standard) definiert eine Zeichenfolge, die angezeigt werden soll, wenn die Daten nicht mit den definierten Status abgeglichen werden können. Wird kein Wert angegeben, wird die numerische Darstellung des nicht abgeglichen Status in Klammern angezeigt.
- Die Eigenschaft *Match Type* (Übereinstimmungstyp) definiert, wie die Daten mit den verschiedenen Status verglichen werden. Wenn „Discrete“ (Getrennt) ausgewählt ist, müssen die Tag-Daten mit dem Datenwert eines bestimmten Status übereinstimmen, damit dieser Status verwendet werden kann. Wenn „Ranged“ (Angeordnet) ausgewählt ist, geht Crimson davon aus, dass die Statusdatenwerte sich in aufsteigender numerischer Reihenfolge befinden. Es wird ein Statuswert verwendet, wenn die Tag-Daten kleiner oder gleich dem Datenwert dieses Status sind, aber größer als der Datenwert des vorherigen Status. Bei der Dateneingabe werden mit angeordneten Formatobjekten Werte zugewiesen, die gleich den tatsächlichen Datenwerten der einzelnen Status sind.

6.9.2 Formatstatus

- Mit den Eigenschaften *Data* (Daten) und *Text* (Text) werden der Datenwert und der Anzeigetext für die einzelnen Status in diesem Format definiert. Status mit leeren Textfeldern werden deaktiviert und ignoriert.

6.9.3 Formatbefehle

Objekte mit dem Format „Multi-State“ (Mehrfachstatus) bieten außerdem Optionen, mit denen ihre verschiedenen Status und die zugehörigen Eigenschaften in Unicode-Textdateien exportiert bzw. daraus importiert werden können. Diese Dateien können dann in einer Anwendung wie Microsoft Excel bearbeitet werden.

6.10 Format Zeichenfolge

Das Format „String“ (Zeichenfolge) verfügt über die folgenden Eigenschaften:

Daten formatieren

Schablone:

Max. Länge:

- Die Eigenschaft *Template* (Vorlage) dient zur Eingabe eines optionalen „Bildes“ vom Erscheinungsbild der Zeichenfolge. Eine Vorlage besteht aus einer Reihe von speziellen Formatierungszeichen, mit denen angegeben wird, welche Art von Zeichen an der jeweiligen Stelle zulässig ist. Folgende Formatierungszeichen werden unterstützt:

Zeichen In Vorlage	Zulässige Zeichen				
	A-Z	a-z	0-9	Leerzeichen	Andere
A	Ja	-	-	-	-
a	Ja	Ja	-	-	-
S	Ja	-	-	Ja	-
s	Ja	Ja	-	Ja	-
N	Ja	-	Ja	-	-
n	Ja	Ja	Ja	-	-
M	Ja	-	Ja	Ja	-
m	Ja	Ja	Ja	Ja	-
0,00	-	-	Ja	-	-
X	Ja	Ja	Ja	Ja	Ja

Die zusätzlichen Zeichen, auf die die Spalte „Andere“ verweist, lauten:

,.;+ -=! ?%/\$

Alle Zeichen, die keine Formatierungszeichen sind, werden als Buchstabensymbole interpretiert und angezeigt, ohne dass sie in den zugrunde liegenden Daten vorhanden sein müssen. Zum Beispiel ermöglicht die Vorlage „(000) 000-0000“ die Eingabe von US-amerikanischen Standardtelefonnummern, ohne dass der Benutzer die Satzzeichen eingeben muss und ohne dass diese Zeichen in jeder Zeichenfolge gespeichert werden.

- Die Eigenschaft *Max Length* (Max. Länge) kann als Alternative zur Eigenschaft „Template“ (Vorlage) verwendet werden, um einen Eintrag in freier Form bis zu einer maximalen Anzahl von Zeichen zu ermöglichen. Beachten Sie, dass die Formatlänge und die zugrunde liegende Datenlänge zwar voneinander unabhängige Werte sind, dass aber der erste Wert in der Regel nicht größer als der zweite sein sollte.

Kapitel 7 Verwenden von Färbungen

Für numerische Tags stehen verschiedene sogenannte „Färbungen“ zur Auswahl. Für Markierungs-Tags und Zeichenfolgen-Tags ist immer die Farbe „Two-State“ (Zweifachstatus), bzw. „General“ (Allgemein) festgelegt. Jede Färbung nimmt einen Datenwert und konvertiert diesen in ein Farbpaar aus Vorder- und Hintergrundfarbe.

7.1 Arten von Färbungen

Die folgenden Färbungen werden unterstützt:



- Die Färbung *General* (Allgemein) gibt immer Weiß auf Schwarz zurück.
- Die Färbung *Linked* (Verbunden) verwendet die Färbung eines anderen Tags zum Formatieren des Tags, das Sie konfigurieren. Damit können Sie Schablonen erstellen und diese dann auf viele Tags in derselben Datenbank anwenden. Hierbei lassen sich Wiederholungen vermeiden, und die Farbeinstellungen können einfach an einem zentralen Ort angepasst werden. Beachten Sie, dass die Färbung „Linked“ (Verbunden) die Daten aus dem ursprünglichen Tag verwendet und die Regeln anwendet, die vom verbundenen Tag definiert wurden. Dies bedeutet, dass das ursprüngliche Tag nicht notwendigerweise dieselbe Farbe hat wie das verbundene Tag und dass seine Farbe von seinem eigenen Datenwert abhängen kann.
- Die Färbung *Fixed* (Fixiert) gibt immer ein festes Farbpaar zurück.
- Die Färbung *Two-State* (Zweifachstatus) nimmt einen numerischen Wert und wählt eines von zwei Farbpaaren aus, je nachdem, ob der Wert Null oder nicht Null ist. Dies ist die permanent definierte Färbung für Markierungs-Tags.
- Die Färbung *Multi-State* (Mehrfachstatus) nimmt einen numerischen Wert und vergleicht ihn mit einer Tabelle, die Datenwerte und Farbpaare enthält. Entweder wird ein Farbpaar ausgewählt, das mit einem übereinstimmenden Datenwert verknüpft ist, oder der Selektor kann so konfiguriert werden, dass das letzte Farbpaar mit einem zugehörigen Datenwert ausgewählt wird, der nicht höher ist als die Daten.

7.2 Färbung Allgemein

Die Färbung „General“ (Allgemein) hat keine Eigenschaften.

7.3 Färbung Verbunden

Die Färbung „Linked“ (Verbunden) hat die folgenden Eigenschaften:

- Mit der Eigenschaft *Color Like* (Farbe wie) können Sie ein Tag auswählen, dem die Färbungsinformationen für dieses Tag entnommen werden sollen. Achten Sie darauf, ein Tag des richtigen Datentyps zu verwenden, d. h. die Färbung eines numerischen Tags muss auf einem anderen numerischen Tag basieren. Andernfalls werden die Standardformatierungsregeln angewendet.

7.4 Färbung Festgelegt

Die Färbung „Fixed“ (Festgelegt) hat die folgenden Eigenschaften:

- Die Eigenschaft *Colors* (Farben) definiert die Farben, die immer verwendet werden sollen.

7.5 Färbung Zweifachstatus

Die Färbung „Two-State“ (Zweifachstatus) hat die folgenden Eigenschaften:

- Die Eigenschaft *ON Colors* (EIN Farben) definiert die Farben, die angezeigt werden sollen, wenn das Tag ungleich Null ist.
- Die Eigenschaft *OFF Colors* (AUS Farben) definiert die Farben, die angezeigt werden sollen, wenn das Tag gleich Null ist.

7.6 Färbung Mehrfachstatus

Die Färbung „Multi-State“ (Mehrfachstatus) hat die folgenden Eigenschaften:

7.6.1 Formatsteuerung

- Die Eigenschaft *States* (Status) definiert, wie viele Status der Mehrfachstatus-Selektor enthält. Maximal 500 Einträge sind möglich. Das Fenster, in dem der Selektor angezeigt wird, wird aktualisiert, um die erforderliche Anzahl von Daten- und Texteingenschaften anzuzeigen.
- Die Eigenschaft *Default Colors* (Standardfarben) definiert die Farben, die verwendet werden sollen, wenn die Daten nicht mit den definierten Status abgeglichen werden können.
- Die Eigenschaft *Match Type* (Übereinstimmungstyp) definiert, wie die Daten mit den verschiedenen Status verglichen werden. Wenn „Discrete“ (Getrennt) ausgewählt ist, müssen die Tag-Daten mit dem Datenwert eines bestimmten Status übereinstimmen, damit dieser Status verwendet werden kann. Wenn „Ranged“ (In Reihe) ausgewählt ist, geht Crimson davon aus, dass die Statusdatenwerte sich in aufsteigender numerischer Reihenfolge befinden. Es verwendet einen Statuswert, wenn die Tag-Daten kleiner oder gleich dem Datenwert dieses Status, aber größer als der Datenwert des vorherigen Status sind.

7.6.2 Formatstatus

- Die Eigenschaften *Data* (Daten) und *Colors* (Farben) definieren die Daten- und Farbwerte für jeden Status.

7.6.3 Farbe Befehle

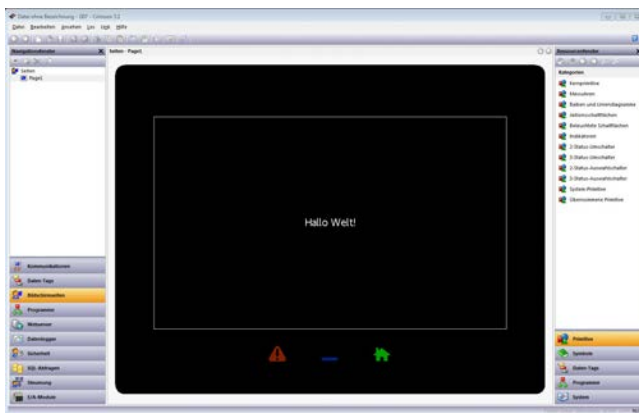
Objekte mit der Färbung „Multi-State“ (Mehrfachstatus) bieten außerdem Optionen, mit denen ihre Status und die zugehörigen Eigenschaften in Unicode-Textdateien exportiert bzw. daraus importiert werden können. Diese Dateien können dann in einer Anwendung wie Microsoft Excel bearbeitet werden. Mit einer zusätzlichen Schaltfläche können die Datenfelder der Färbung mit den Datenfeldern eines Objekts im Format „Multi-State“ (Mehrfachstatus), das für das gleiche Tag konfiguriert ist, synchronisiert werden, damit Sie die gleichen Werte nicht zweimal eingeben müssen.

Kapitel 8 Erstellen von Anzeigeseiten

Über die Kategorie „Display Pages“ (Anzeigeseiten) im Navigationsfenster haben Sie Zugriff auf den neuen Grafikeditor von Crimson® 3.1. Dieser Editor ermöglicht die schnelle und effiziente Erstellung attraktiver Displays und bietet gleichzeitig ein Höchstmaß an Flexibilität.

8.1 Editor-Grundlagen

Nachstehend sehen Sie, wie der Grafikeditor anfänglich aussieht:



Das Bearbeitungsfenster enthält eine Darstellung des Zielgeräts, einschließlich des Display-Bereichs selbst sowie aller Tasten und LED-Symbole. Auf der niedrigsten Vergrößerungseinstellung wird die gesamte Anzeige angezeigt. Dies bedeutet möglicherweise, dass den einzelnen Pixeln auf dem Display des Zielgeräts jeweils weniger als ein Pixel auf Ihrem PC-Display zugewiesen wird. In diesem Fall ist es zwar weiterhin möglich, die meisten Bearbeitungsschritte durchzuführen und Seiten anzuzeigen, doch die Präzision wird ein wenig beeinträchtigt. Daher wird eine entsprechende Warnmeldung angezeigt.

8.1.1 Arbeiten mit Seiten

Die Manipulation von Anzeigeseiten über die Navigationsliste ist intuitiv und funktioniert wie bei jedem anderen Element in einer Crimson-Datenbank. An dieser Stelle soll daran erinnert werden, dass Seiten einfach zwischen Datenbanken kopiert werden können, indem Sie sie im Navigationsfenster der einen Datenbank auswählen und in die entsprechende Kategorie in der Zieldatenbank ziehen. Dadurch ist es sehr einfach, neue Datenbanken durch Kombinieren bereits verwendeter Seitendesigns zu erstellen.

8.1.2 Ändern der Vergrößerungseinstellung

Die Vergrößerungseinstellung lässt sich am einfachsten mit dem Mausrad ändern. Wenn Sie nicht über eine solche Maus verfügen, können Sie den Vergrößerungsmodus des Editors verwenden. Wählen Sie dazu in der Symbolleiste das Vergrößerungsglas aus. In diesem Modus können Sie die Ansicht durch Klicken mit der linken Maustaste vergrößern. Durch Klicken mit der rechten Maustaste bzw. durch Drücken der **STRG**-Taste und Klicken mit der linken Maustaste wird die Ansicht verkleinert. Sie können auch die Größenänderungsbefehle im Menü „View“ (Ansicht) verwenden.

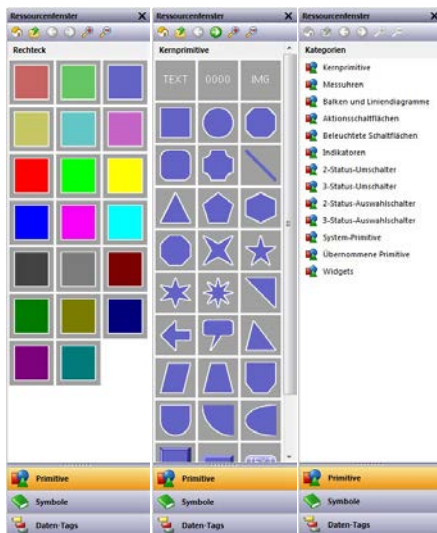
Der erste Schritt bei der Größenänderung besteht darin, von der Vollbildansicht zu einer 1:1-Anzeige zu wechseln, bei der das Display des Zielgeräts im Bearbeitungsfenster zentriert wird. Danach wird die Größe der Ansicht so

verändert, dass die Daten unter dem Mauszeiger weiterhin angezeigt werden. Dadurch können Sie leichter wählen, welchen Bereich des Displays Sie sich genauer ansehen möchten.

8.1.3 Das Ressourcenfenster

Anzeigeseiten werden in der Regel aus Elementen zusammengestellt, die Sie aus dem Ressourcenfenster ziehen. Sie können das Ressourcenfenster einblenden, indem Sie auf die Pfeilleiste rechts im Fenster klicken. Sie können das Ressourcenfenster auch einrasten lassen. Dabei bietet es sich möglicherweise an, das Programmfenster zu maximieren und so den verfügbaren Arbeitsbereich zu vergrößern. Im Ressourcenfenster gibt es drei Kategorien:

8.1.3.1 Primitive



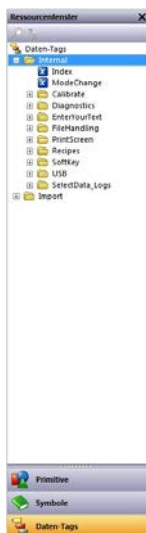
Über die Kategorie „Primitives“ (Primitive) haben Sie Zugriff auf die Hauptbausteine, aus denen Anzeigeseiten erstellt werden. Links in der Abbildung sehen Sie, wie die Kategorie je nach ihrem Status aussehen kann. Wie Sie sehen, enthält die oberste Ebene eine Reihe von Unterkategorien, die jeweils Zugriff auf eine Reihe von Primitiven bieten. Wenn Sie auf ein Symbol klicken, wird eine Unterkategorie mit ihren Primitiven angezeigt. Wenn Sie auf ein bestimmtes Primitiv klicken, werden Varianten dieses Primitivs in vordefinierten Farben angezeigt. Mit den Symbolen in der Symboleiste können Sie zwischen Unterkategorien wechseln, zu einer höheren Ebene wechseln oder die Anzahl der in einer Zeile angezeigten Primitive ändern. Primitive werden im nächsten Kapitel erörtert.

8.1.3.2 Symbolbibliothek



Die Kategorie „Symbol Library“ (Symbolbibliothek) funktioniert ähnlich wie die Kategorie „Primitives“ (Primitive). Sie bietet Zugriff auf eine Reihe von Unterkategorien, die jeweils eine Reihe von vordefinierten Symbolen enthalten. Wenn Sie auf ein bestimmtes Symbol klicken, wird eine Reihe von vorgefärbter Varianten dieses Symbols angezeigt. Diese Funktion wird allerdings seltener verwendet als bei Primitiven. Nehmen Sie sich etwas Zeit, die Symbolbibliothek zu erkunden. Sie enthält Tausende von Bildern, die Datenbanken bei richtiger Verwendung attraktiver und benutzerfreundlicher gestalten können.

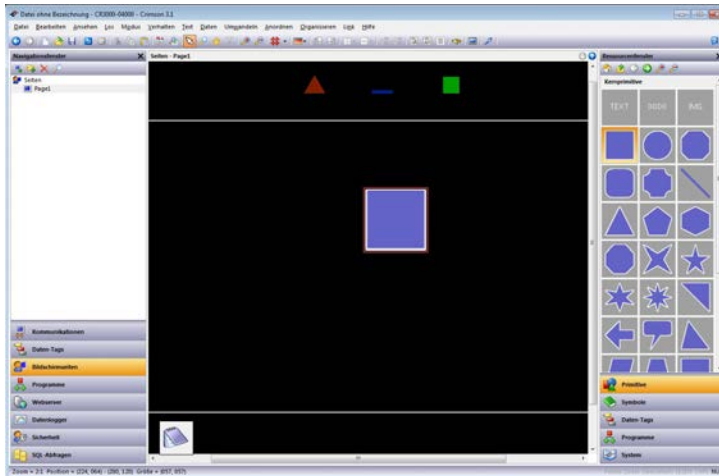
8.1.3.3 Datentags



Die Kategorie „Data Tags“ (Datentags) enthält eine Baumansicht aller Datentags in der aktuellen Datenbank. Sie ermöglicht es, Tags direkt auf eine Anzeigeseite zu ziehen, und bietet bei der Konfiguration der Eigenschaften von Primitiven Zugriff auf Tags. Wenn Sie ein Tag auf eine Seite ziehen, wird ein Datenfeld erstellt, das an dieses Tag gebunden ist. Dabei basieren alle Formatierungseigenschaften auf Eigenschaften, die durch das Tag selbst definiert ist. Sie können auch mehrere Tags auswählen und ziehen. Verwenden Sie dazu wie gewohnt die Tasten UMSCHALT und STRG. Dank dieser Funktion können Sie einer Seite sehr schnell und einfach Daten hinzufügen.

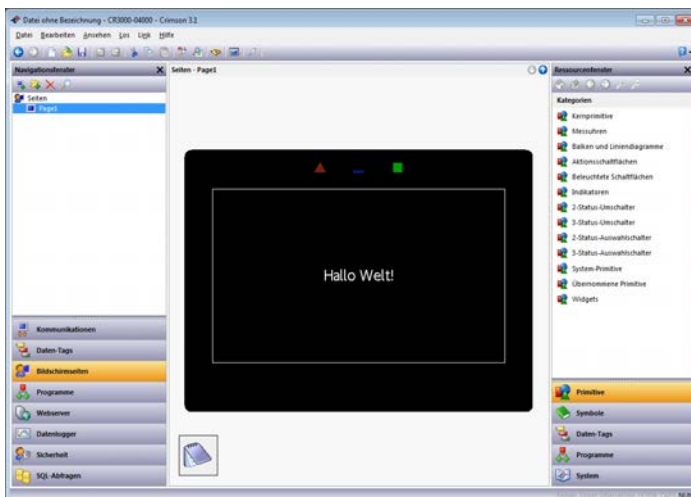
8.1.4 Hinzufügen von Elementen zu einer Seite

Wie oben erwähnt, können die verschiedenen Elemente im Ressourcenfenster auf den Editor gezogen und somit einer Anzeigeseite hinzugefügt werden. Für Tags und Bilder werden geeignete Primitive erstellt. Im folgenden Beispiel sehen Sie, wie nach dem Klicken auf „Core Primitives“ (Kernprimitive) in der Kategorie „Primitives“ (Primitive) ein rechteckiges Primitiv auf die Seite gezogen werden kann:

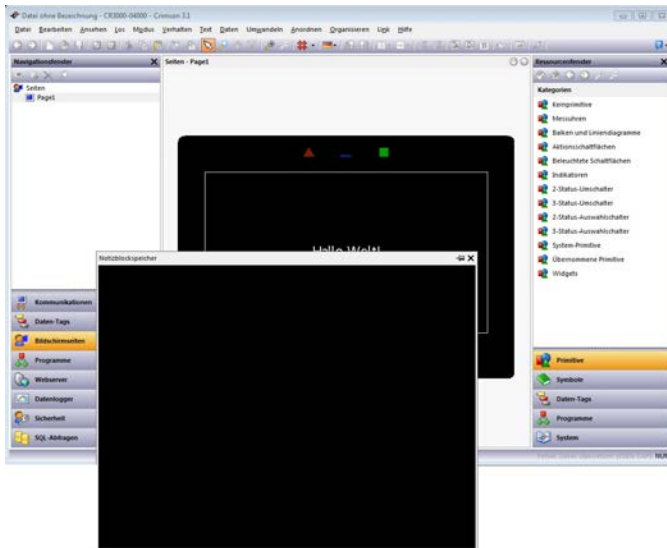


8.1.5 Verwendung des Notizblocks

Der Grafikeditor von Crimson bietet einen Scratchpad-Bereich, der verwendet werden kann, um häufig verwendete Prims zu speichern oder um einen Ort zum vorübergehenden Ablegen von Prims bereitzustellen, während Sie eine Anzeigeseite neu organisieren. Der Inhalt des Scratchpads wird in der Datenbank gespeichert und steht somit von einer Bearbeitungssitzung zur nächsten zur Verfügung. Der Scratchpad kann über die Option „Show Scratchpad“ (Scratchpad anzeigen) im Menü „View“ (Ansicht) aktiviert oder angezeigt werden. Wenn diese Option aktiviert ist, wird sie zunächst als Notizblocksymbol in der linken unteren Ecke des Bearbeitungsfensters angezeigt.



Diese minimierte Form des Scratchpad-Fensters kann an jeden beliebigen Ort verschoben werden, auch auf einen zweiten Monitor. Auf diese Weise können Sie Ihren Arbeitsbereich nach Ihren Wünschen und entsprechend der verfügbaren Monitorfläche organisieren. Um das Scratchpad zu öffnen, klicken Sie entweder auf die minimierte Form oder ziehen Sie ein Prim von der aktuellen Anzeigeseite auf das Notizblocksymbol. Das Scratchpad wird nun geöffnet angezeigt, wie unten dargestellt...



Auf dem Scratchpad platzierte Elemente können wie gewohnt bearbeitet und in den Haupteditor gezogen werden, von wo aus sie zur aktuellen Anzeigeseite hinzugefügt werden können. Standardmäßig erfolgt bei einer solchen Drag-and-Drop-Aktion ein Kopiervorgang, wobei das ursprüngliche Prim auf dem Scratchpad verbleibt. Wenn Sie dies ändern möchten, halten Sie beim Ziehen des Prims die **UMSCHALTTASTE** gedrückt.

Das Scratchpad-Fenster wird automatisch ausgeblendet, wenn Sie versuchen, auf einen Bereich des Bildschirms zuzugreifen, der sonst verdeckt wäre. Wenn Sie beispielsweise im oben gezeigten Layout ein Element aus dem Scratchpad in den Haupteditor ziehen, wird das Scratchpad ausgeblendet, sodass Sie keinen ungehinderten Zugriff mehr auf den Teil der Seite haben, der vom Scratchpad-Fenster abgedeckt wird. Um das automatisch ausgeblendete Scratchpad anzuzeigen, klicken Sie auf das Reißzweckensymbol.

8.2 Arbeiten mit Primitiven

In den folgenden Abschnitten wird beschrieben, wie Sie einfache Vorgänge an Primitiven ausführen.

8.2.1 Auswählen von Primitiven

Zum Auswählen eines Anzeigeprimativs bewegen Sie einfach den Mauszeiger auf das betreffende Primitiv und klicken mit der linken Maustaste. Wenn der Mauszeiger sich über einem Primitiv befindet, wird ein blaues Begrenzungsrechteck eingeblendet, an dem Sie erkennen, was ausgewählt wird. Sobald etwas ausgewählt wurde, wird das Rechteck rot, und es werden Griffe angezeigt, mit denen Sie die Größe des Primitivs bei Bedarf ändern können.

Zum Auswählen mehrerer Primitive ziehen Sie ein Auswahlrechteck um die gewünschten Primitive. Alternativ können Sie jedes Primitiv einzeln auswählen und dabei die **UMSCHALT**TASTE gedrückt halten, um anzugeben, dass jedes Primitiv der Auswahl hinzugefügt werden soll. Wenn mehrere Primitive ausgewählt sind, umgibt das rote Rechteck alle Primitive. Mit den Griffen des Rechtecks können Sie dann die Primitive als Gruppe vergrößern oder verkleinern. Die relative Größe und Position der Primitive wird beibehalten, solange dies in Crimson möglich ist, ohne gegen die Anforderungen bezüglich der Mindestgröße zu verstoßen.

8.2.2 Verborgene Primitive

Wenn Sie feststellen, dass das Primitiv, das Sie auswählen möchten, von einem anderen Primitiv verdeckt ist, drücken Sie die **STRG**-Taste. Dann können Sie die gewünschte Auswahl vornehmen. Alternativ klicken Sie mit der rechten Maustaste, um das Kontextmenü aufzurufen, und wählen das Untermenü „Select“ (Auswählen). Hierdurch wird eine Liste aller Primitive angezeigt, die sich unter dem Mauszeiger befinden, von hinten nach vorn sortiert. Mit jedem Befehl wird das entsprechende Primitiv ausgewählt. Die Auswahl des richtigen Elements gestaltet sich dadurch sehr einfach.

8.2.3 Verwenden der Schnellleiste

Die Schnellleiste ist eine schwebende Symbolleiste, die rechts neben der aktuellen Auswahl angezeigt wird:



Die Leiste ist zunächst transparent und wird immer deutlicher, je näher der Mauszeiger kommt. Wenn Sie den Mauszeiger von der Leiste weg bewegen, wird sie ausgeblendet. Sie erscheint erst wieder, wenn Sie eine weitere Auswahl vornehmen oder das Mauseisrad verwenden. Die Schnellleiste bietet einen einfachen Zugriff auf eine Reihe gängiger Funktionen und reduziert gleichzeitig die Mausbewegung auf ein Minimum. Die Leiste kann mithilfe eines Befehls im Menü „View“ (Ansicht) aktiviert oder deaktiviert werden.

8.2.4 Verschieben von Primitiven zwischen Seiten

Primitive können auf einer Anzeigeseite in der üblichen Art und Weise verschoben werden, aber auch von einer Seite auf eine andere kopiert werden. Wählen Sie hierzu das zu kopierende Primitiv aus, und ziehen Sie es zum Navigationsfenster. Wenn das Fenster ausgeblendet ist, bewegen Sie den Mauszeiger auf die Pfeilleiste. Das Fenster wird dann eingeblendet. Halten Sie die Maus auf die Zielseite. Diese Seite wird daraufhin ausgewählt. Ziehen Sie nun das Primitiv zurück in den Editor und legen Sie es auf der neuen Seite ab. Wenn Sie die **STRG**-Taste gedrückt halten, wird das Primitiv nicht kopiert, sondern verschoben.

8.2.5 Verschieben von Primitiven zwischen Datenbanken

Primitive können auch sehr einfach zwischen Datenbanken gezogen werden. Wählen Sie einfach die Elemente aus, die Sie kopieren möchten. Ziehen Sie sie dann in eine andere Instanz von Crimson, die die neue Datenbank enthält. Dies funktioniert mit ganzen Seiten, Gruppen von Primitiven oder mit einzelnen Elementen.

8.2.6 Ändern der Größe von Primitiven

Die Größenänderung von Primitiven wird intuitiv durchgeführt, indem Sie auf einen der Griffe klicken und in die gewünschte Richtung bewegen. Wenn Sie beim Ziehen die **STRG**-Taste gedrückt halten, bleibt das Verhältnis zwischen Breite und Höhe des Primitivs bei der Größenänderung erhalten. Wenn Sie die **UMSCHALT**TASTE gedrückt halten, erfolgt die Größenänderung vom Mittelpunkt aus, anstatt vom Rand.

8.2.7 Drehen und Spiegeln von Primitiven

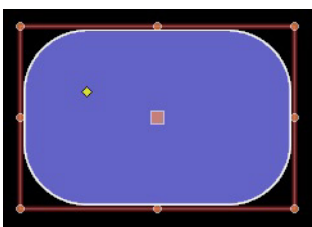
Bestimmte Primitive können in 90-Grad-Schritten gedreht sowie entlang ihrer horizontalen oder vertikalen Achse gespiegelt werden. Die Einstellungen, mit denen die Ausrichtung des Primitivs gesteuert wird, können direkt über das Dialogfeld „Eigenschaften“ des Primitivs festgelegt oder über die Befehle im Menü „Transform“ (Transformieren) geändert werden. Auch Tastaturkürzel können verwendet werden. Mit **STRG+ALT+NACH LINKS** und **STRG+ALT+NACH RECHTS** wird das ausgewählte Primitiv im bzw. gegen den Uhrzeigersinn gedreht. Mit **STRG+ALT+NACH OBEN** und **STRG+ALT+NACH UNTEN** wird das ausgewählte Primitiv entlang seiner vertikalen bzw. horizontalen Achse gespiegelt. Wenn ein geeignetes Primitiv ausgewählt ist, werden auch in der Schnellleiste Symbole zur Ausführung dieser Vorgänge angezeigt.

8.2.8 Verwenden von Layoutgriffen

Einige Primitive haben interne Griffe, die verschoben werden können, um das Layout zu ändern. Zum Beispiel enthält das abgerundete Rechteck in der nachstehenden Abbildung in der oberen linken Ecke einen einzelnen Layoutgriff. Der Griff ist durch eine Raute gekennzeichnet, wenn das Primitiv ausgewählt ist:



Wenn der Griff verschoben wird, ändert sich der Radius der Rechteckwinkel:



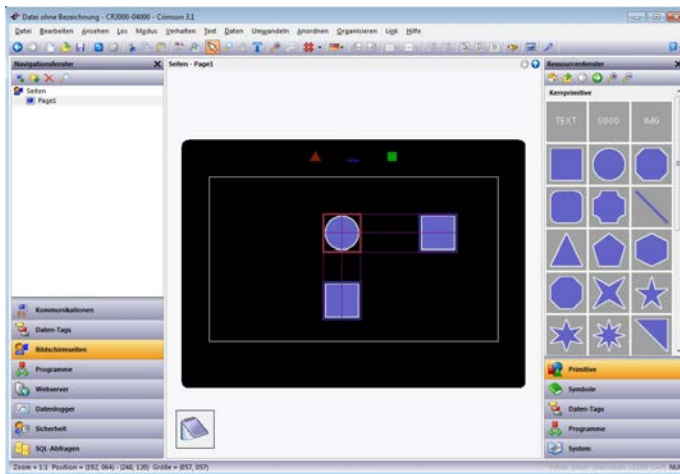
Die Funktion jedes Griffs hängt vom jeweiligen Primitiv ab, ist aber in der Regel intuitiv.

8.2.9 Intelligente Ausrichtung

Wenn Sie im Menü „View“ (Ansicht) die Funktion „Smart Align“ (Intelligent ausrichten) aktiviert haben, werden in Crimson während einer Verschiebung oder einer Größenänderung Führungslinien bereitgestellt. Diese dienen dazu, ein Primitiv an vorhandenen Primitiven oder an der Mitte des Bildschirms auszurichten. Mit etwas Übung

können Sie mit dieser Funktion sehr einfach Primitive gleich bei deren Erstellung ausrichten. Sonst müssten Sie zurückgehen und Ihre Anzeigeseiten ändern, damit die verschiedenen Figuren ausgerichtet sind.

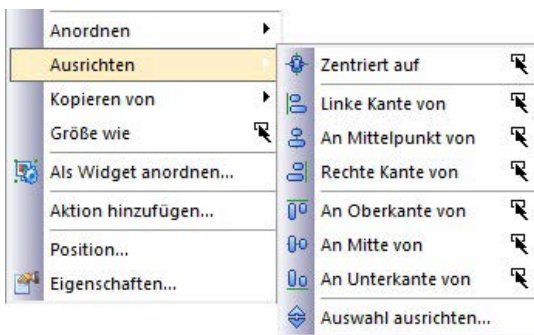
Im nachstehenden Beispiel wird ein Kreis an zwei Quadraten ausgerichtet:



An beiden Kanten der Formen sind Führungslinien vorhanden, was bedeutet, dass sie ausgerichtet sind. Das rote Rechteck hebt das Primitiv hervor, das manipuliert wird. Die beiden blauen Rechtecke heben die Primitive hervor, zu denen die Führungslinien gezeichnet wurden.

8.2.10 Schnellausrichtung

Mit den Schnellausrichtungsfunktionen von Crimson können Primitive an anderen Primitiven ausgerichtet werden, ohne dass ein Dialogfeld geöffnet werden muss. Um diese Funktion zu verwenden, wählen Sie einfach das auszurichtende Primitiv aus und klicken mit der rechten Maustaste, um das Kontextmenü aufzurufen. Wählen Sie das Untermenü „Align“ (Ausrichten) und dann eine der Optionen „With... Of“ (An... von) aus. Diese Optionen sind mit einem Symbol gekennzeichnet, das aus einem Rechteck und Cursor besteht. Der Mauszeiger ändert sich, um darauf hinzuweisen, dass Sie nun auf das Primitiv klicken müssen, an dem ausgerichtet werden soll.



Sobald Sie klicken, wird die Ausrichtung durchgeführt.

8.2.11 Verwenden des Gitters

Über die Schaltfläche „Grid“ (Gitter) in der Symbolleiste können Sie das Verhalten und die Anzeige des Ausrichtungsgitters steuern. Durch Klicken auf den linken Bereich der Schaltfläche wird das Gitter ein- oder ausgeblendet. Wenn Sie auf den Dropdown-Bereich klicken, können Sie die Vorgänge konfigurieren, für die das Raster verwendet werden soll. Sie können das Gitter separat für Erstellungs-, Größenänderungs- und Verschiebungsvorgänge aktivieren oder deaktivieren. Mit den Optionen „All“ (Alle) oder „None“ (Keine) können Sie

das Gitter global aktivieren oder deaktivieren. Außerdem können Sie steuern, ob das Gitter bei Bearbeitungsvorgängen innerhalb von Gruppen verwendet wird.

8.2.12 Ausrichten von Primitiven

Mit den oben erläuterten Optionen „Intelligente Ausrichtung“ und „Schnellausrichtung“ kann eine Vielzahl von Ausrichtungsvorgängen durchgeführt werden, doch gelegentlich möchten Sie sich eines eher traditionellen Ansatzes bedienen. Wählen Sie hierzu einige Primitive aus und wählen Sie im Menü „Arrange“ (Anordnen) den Befehl „Align selection“ (Auswahl ausrichten) aus, um folgendes Dialogfeld aufzurufen:



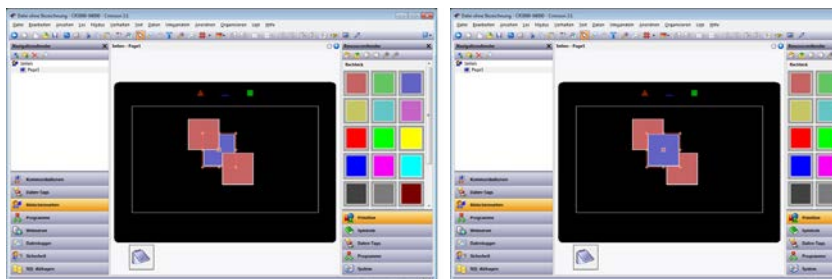
Mit den Einstellungen unter „Horizontal“ und „Vertikal“ (Vertikal) können Sie angeben, welche Art von Ausrichtung durchgeführt werden soll. Mit der Einstellung „Referenz“ (Referenz) geben Sie an, welches Objekt als Referenz für den Ausrichtungsvorgang verwendet wird. Im oben gezeigten Beispiel verwendet der Modus „Auto“ (Automatisch) das ganz links befindliche Primitiv als Referenz, da eine Ausrichtung nach links durchgeführt wird. Andere Ausrichtungsmodi funktionieren ähnlich. Der alternative Modus verwendet das zuerst ausgewählte Element als Referenz. Dieses Element erkennen Sie daran, dass in seiner Mitte ein größeres Quadrat zu sehen ist.

8.2.13 Gleichmäßige Verteilung von Primitiven

Wenn Ihnen mehrere Primitive vorliegen, die Sie gleichmäßig auf der Seite verteilen möchten, können Sie die Befehle „Space Equally Vertical“ (Gleichmäßig vertikal verteilen) und „Space Equally Horizontal“ (Gleichmäßig horizontal verteilen) im Menü „Arrange“ (Anordnen) verwenden. Beide Befehle werden auf die gegenwärtig ausgewählten Primitive angewendet. Sie versuchen, den freien Platz zwischen den Elementen neu zuzuordnen, um eine gleichmäßige Verteilung zu erzielen. Die beiden äußeren Primitive verbleiben an den gegenwärtigen Positionen. Beachten Sie, dass der Befehl möglicherweise fehlschlägt, wenn ungeeignete Primitive ausgewählt werden. Wenn zu wenig freier Platz verfügbar ist, lässt sich möglicherweise keine perfekte Verteilung erzielen.

8.2.14 Ändern der Reihenfolge von Primitiven

Primitive werden in der so genannten Z-Reihenfolge (auch: Z-Ordnung) auf einer Bildschirmseite abgelegt. Gemeint ist damit die Reihenfolge, in der die Primitive gezeichnet werden, und damit auch, ob ein bestimmtes Primitiv vor oder hinter einem anderen steht. Im ersten Beispiel unten befindet sich das blaue Quadrat am Ende der Z-Reihenfolge und wird daher hinter den roten Quadraten angezeigt. Im zweiten Beispiel wurde es an den Anfang der Reihenfolge verschoben und befindet sich jetzt vor den anderen Formen.



Um Elemente in der Z-Reihenfolge zu verschieben, wählen Sie die Elemente aus und verwenden dann die verschiedenen Befehle im Menü „Arrange“ (Anordnen). Mit den Befehlen „Move Forward“ (Nach vorne) und „Move Backward“ (Nach hinten) wird die Auswahl um einen Schritt in die angegebene Richtung verschoben. Mit den Befehlen „Move To Front“ (Nach vorn verschieben) und „Move To Back“ (Nach hinten verschieben) wird die Auswahl an das angegebene Ende der Z-Reihenfolge verschoben. Mit einer Maus, die mit einem Rad ausgestattet ist, können Sie die Auswahl auch verschieben, indem Sie das Rad bei gedrückter **STRG**-Taste betätigen. Beim Scrollen nach oben wird die Auswahl in der Z-Reihenfolge nach hinten verschoben, und beim Scrollen nach unten wird sie nach vorn verschoben.

8.2.15 Duplizieren von Primitiven

Mit der Tastenkombination **STRG+D** oder dem Befehl „Smart Duplicate“ (Smart duplizieren) im Menü „Edit“ (Bearbeiten) können Sie eine Kopie des aktuellen Primitivs erstellen und dessen Eigenschaften so anpassen, dass das Primitiv seine Steuerungsdaten vom nächsten Datenelement erhält. Was unter dem „nächsten“ Datenelement zu verstehen ist, ist vom genauen Typ des Datenelements abhängig. Crimson kann das nächste Register in einem Kommunikationsgerät, das nächste Mitglied eines Arrays oder das nächste Tag in einer Sequenz auswählen. Wenn Sie beispielsweise wiederholt „Smart Duplicate“ (Smart duplizieren) bei einer Schaltfläche verwenden, die `Array[0]` zugeordnet ist, wird eine Reihe von Schaltflächen erzeugt, die `Array[1]`, `Array[2]` usw. zugeordnet sind, bis der gesamte Bildschirm gefüllt ist.

8.2.16 Bearbeiten mehrerer Primitive

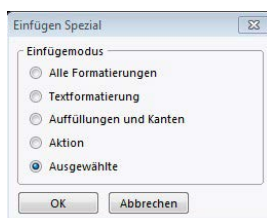
Gelegentlich kann es vorkommen, dass Sie die Eigenschaften mehrerer Primitive bearbeiten möchten. Dies ist in Crimson möglich. Sie bearbeiten ein Primitiv und können dann die Eigenschaften einiger anderer Primitive auf die Eigenschaften des zuerst bearbeiteten Primitivs setzen. Crimson bietet hierfür zwei Methoden, die beide auf demselben Mechanismus beruhen.

8.2.16.1 Verwenden von Kopieren von

Mit dem Befehl „Copy From“ (Kopieren von) können die ausgewählten Eigenschaften eines Primitivs in ein oder mehrere Primitive kopiert werden. Zur Verwendung des Befehls wählen Sie die gewünschten Ziele aus und klicken dann mit der rechten Maustaste, um das zugehörige Kontextmenü aufzurufen. Wählen Sie einen der Befehle „Copy From“ (Kopieren von) aus. Der Cursor ändert sich, damit Sie das Primitiv auswählen können, von dem aus der Kopiervorgang ausgeführt werden soll. Je nach ausgewähltem Befehl werden eine oder mehrere Eigenschaften aus dem Quellprimitiv auf die Zielprimitive angewendet.

8.2.16.2 Verwenden von Einfügen spezial

Mit dem Befehl „Paste Special“ (Einfügen spezial) können Sie dasselbe Ergebnis erzielen. Bei dieser Methode ist es jedoch möglich, auch Eigenschaften zwischen Datenbanken und zwischen mehreren Instanzen von Crimson zu kopieren. Wählen Sie zuerst das Quellprimitiv und dann den Befehl „Copy“ (Kopieren) aus, um das Primitiv in die Zwischenablage zu kopieren. Wählen Sie dann die gewünschten Zielprimitive aus. Klicken Sie mit der rechten Maustaste auf die Auswahl und wählen Sie den Befehl „Paste Special“ (Einfügen spezial) aus. Das folgende Dialogfeld wird angezeigt:



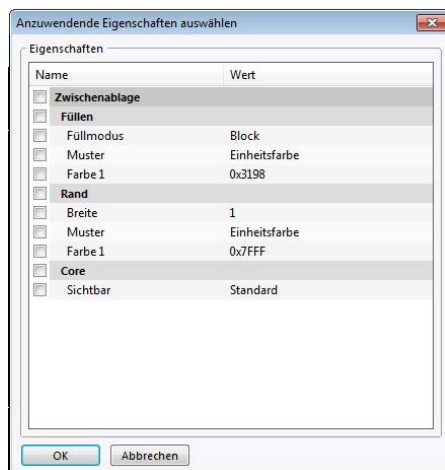
Die ausgewählten Eigenschaften aus dem Quellprimitiv werden auf die Zielprimitive angewendet.

8.2.16.3 Eigenschaftsauswahl

Bei beiden oben beschriebenen Methoden können Sie festlegen, welche Eigenschaften kopiert werden sollen:

- Mit *All Formatting* (Alle Formatierungen) wird mit Ausnahme von Text, Datenelementen oder Aktionen alles kopiert.
- Mit *Text Format* (Textformat) werden die Schriftart, Ausrichtung und Ränder von Text oder Datenelementen kopiert.
- Mit *Fills and Edges* (Flächen und Kanten) werden die Flächen- und Kantenattribute aus der Registerkarte „Figure“ (Bild) kopiert.
- Mit *Action* (Aktion) werden alle Aktionen kopiert, die dem Primitiv zugewiesen sind.

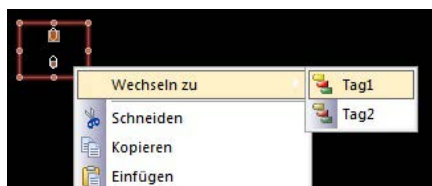
Darüber hinaus kann die Option *Selektive* (Wahlweise) zur Auswahl der zu kopierenden Eigenschaften verwendet werden:



Die Liste enthält eine hierarchische Aufstellung der Eigenschaften, die vom Quellprimitiv definiert werden. Die Eigenschaften sind nach dem Layout angeordnet, das bei der Bearbeitung des Primitivs verwendet wird. Für jede Eigenschaft wird zudem der ihr zugewiesene Wert angezeigt. Jede Eigenschaft oder Gruppe von Eigenschaften kann mithilfe der entsprechenden Kontrollkästchen aktiviert oder deaktiviert werden. Die aktivierten Eigenschaften werden angewendet, sodass Sie im Detail steuern können, was von einem Primitiv zum andern kopiert wird.

8.2.17 Springen zu anderen Elementen

Wenn ein Primitiv auf Tags, Anzeigeseiten oder andere Elemente verweist, wird im Kontextmenü das Untermenü „Jump“ (Springen) angezeigt. Wählen Sie dieses Menü aus, um eine Liste der referenzierten Elemente anzuzeigen. Wählen Sie eines dieser Elemente aus, um direkt zu diesem Abschnitt der Datenbank zu springen. Das folgende Beispiel zeigt ein Primitiv, das auf zwei Tags verweist:



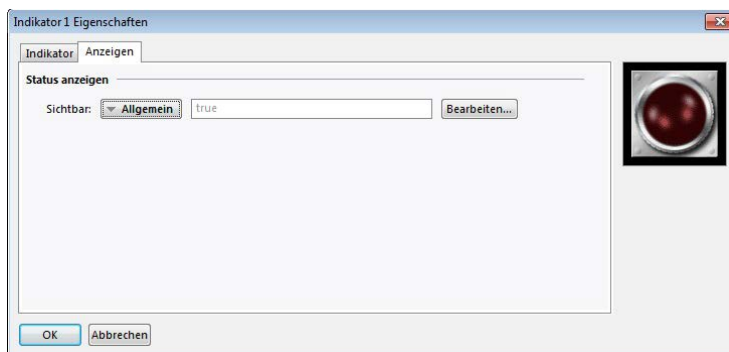
Nachdem Sie die gewünschten Änderungen am Tag vorgenommen haben, können Sie mit der Schaltfläche „Back“ (Zurück) in der Symbolleiste oder mit der Tastenkombination **ALT+NACH LINKS** zu der Anzeigeseite zurückkehren, die Sie gerade bearbeiten. Beachten Sie, dass die Auswahl während der Navigation beibehalten wird. Dadurch ist es einfach, ein referenziertes Objekt anzuzeigen oder zu bearbeiten und dann zur Erstellung der Anzeige zurückzukehren.

8.3 Eigenschaften von Primitiven

Zur Bearbeitung der Eigenschaften eines Primitivs können Sie entweder auf das Primitiv doppelklicken oder im Kontextmenü des Primitivs den Befehl „Properties“ (Eigenschaften) auswählen. Sie können auch das Primitiv auswählen und die Tasten ALT+EINGABE drücken. Das Dialogfeld „Properties“ (Eigenschaften) eines Primitivs enthält verschiedene Registerkarten. Einige Registerkarten werden nur angezeigt, wenn dem Primitiv zusätzliche Elemente wie Text, Daten oder eine Aktion, hinzugefügt wurden. Im Dialogfeld „Properties“ (Eigenschaften) wird eine Live-Vorschau des aktuellen Primitivs angezeigt, sodass Sie die Auswirkungen von Änderungen sehen können, bevor Sie sie übernehmen.

8.3.1 Ein- oder Ausblenden von Primitiven

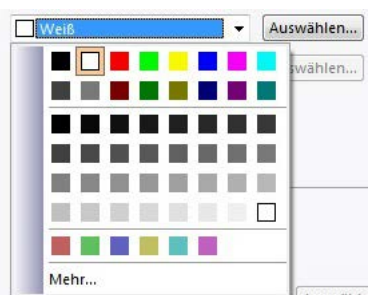
Das Dialogfeld „Properties“ (Eigenschaften) jedes Primitivs enthält die Registerkarte „Show“ (Anzeigen):



Die Eigenschaft *Visible* (Sichtbar) kann auf einen Ganzzahlausdruck gesetzt werden, um das zugehörige Primitiv zur Laufzeit ein- oder auszublenden. Beim Wert Null wird das Primitiv ausgeblendet und bei einem Wert von nicht Null wird es eingeblendet. Alle Primitive sind standardmäßig sichtbar.

8.3.2 Definieren von Farben für Primitive

Die Farben in Primitiven werden mit einem Feld bearbeitet, das in etwa wie in der nachstehenden Abbildung aussieht:



Für die Farbeigenschaft werden eine Pfeilschaltfläche, eine Dropdown-Liste und eine Schaltfläche „Pick“ (Auswählen) angezeigt. Mit der Pfeilschaltfläche wird die Liste eingeblendet, die die verschiedenen Farbanimationsmodi enthält. Sie kann wie gewohnt durch Klicken mit der Maus oder durch Drücken der Leertaste bei ausgewählter Schaltfläche aktiviert werden. Wenn die Schaltfläche ausgewählt ist, können Sie auch den ersten Buchstaben einer der in der Liste enthaltenen Optionen drücken und dadurch die Anzeige der Liste umgehen. Wenn Sie beispielsweise 2 drücken, wird der Modus „2-State“ (Zweifachstatus) ausgewählt.

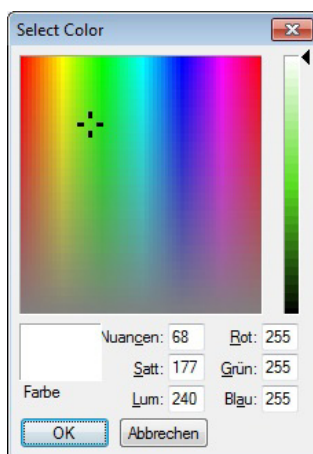
Die folgenden Farbanimationsmodi stehen zur Verfügung:

- Im Modus *Fixed* (Festgelegt) ändert sich die Farbe nicht. Sie wird entweder in der Dropdown-Liste oder im Farbauswahl-Dialogfeld ausgewählt. Das Dialogfeld wird über die Schaltfläche „Pick“ (Auswählen) aufgerufen.
- Im Modus *Tag Text* (Tag-Text) wird die Farbe animiert, um sie an die von einem bestimmten Tag definierte Vordergrundfarbe anzupassen. Das jeweilige Tag kann über die Schaltfläche „Pick“ (Auswählen) ausgewählt werden.
- Im Modus *Tag Back* (Tag-Hintergrund) wird die Farbe animiert, um sie an die von einem bestimmten Tag definierte Hintergrundfarbe anzupassen. Das jeweilige Tag kann über die Schaltfläche „Pick“ (Auswählen) ausgewählt werden.
- Im Modus *Flashing* (Blinkend) wird die Farbe animiert, um mit einer bestimmten Geschwindigkeit zwischen zwei Farben hin und her zu wechseln. Wenn das Blinken deaktiviert ist, wird eine andere Farbe angezeigt.
- Im Modus *2-State* (Zweifachstatus) wird die Farbe animiert, um je nach dem Wert eines Tags oder eines anderen Datenelements zwischen zwei Farben zu wechseln.
- Im Modus *4-State* (Vierfachstatus) wird die Farbe animiert, um je nach dem Wert zweier Tags oder anderer Datenelemente zwischen vier Farben zu wechseln.
- Im Modus *Blended* (Gemischt) wird die Farbe animiert, um gleichmäßig von einer Farbe zu einer anderen überzugehen, ausgehend vom Wert eines Tags oder eines anderen Datenelements im Verhältnis zu angegebenen Mindest- und Höchstwerten.
- Im Modus *Expression* (Ausdruck) kann ein numerischer Ausdruck eingegeben werden, anhand dessen die anzuzeigende Farbe bestimmt wird. Weitere Informationen finden Sie unten.
- Im Modus *Complex* (Komplex) kann ein lokales Programm geschrieben werden, das einen ganzzahligen Wert zurückgibt, um die anzuzeigende Farbe zu definieren. Weitere Informationen finden Sie unten.

Das Dropdown-Menü enthält die folgenden Farben:

- die sechzehn VGA-Standardfarben
- zweiunddreißig Grautöne zwischen Schwarz und Weiß
- alle anderen in der Datenbank verwendeten Farben, bis maximal vierundzwanzig

Mit der Option „More“ (Mehr) am Ende der Liste können Sie das Farbauswahl-Dialogfeld aufrufen:



Dieses Dialogfeld bietet mehrere Möglichkeiten zur Definition einer Farbe. Sie können eine Farbe in der Palette oder im „Regenbogen“-Fenster auswählen oder die expliziten HSL- oder RGB-Parameter eingeben. Wenn die ausgewählte Farbe noch nicht in der Datenbank verwendet wurde und nicht zu den Standardfarben oder Graustufen gehört, wird sie den benutzerdefinierten Farben im Dropdown-Menü hinzugefügt.

8.3.3 Definieren von blinkenden Farben

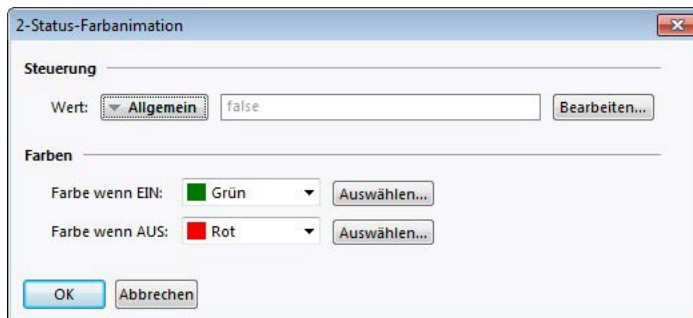
Blinkende Farben werden über das folgende Dialogfeld definiert:



- Die Eigenschaft *Rate* (Frequenz) definiert die Blinkfrequenz. Der Wert 1 ergibt eine Blinkfrequenz von 1 Hz, wobei jede Farbe 500 ms lang angezeigt wird. Von Frequenzen über 4 Hz wird abgeraten, weil die Aktualisierungsrate der Anzeige des Zielgeräts unangenehme „Pulsations“-Effekte erzeugen kann.
- Die Eigenschaft *Enable* (Aktivieren) definiert einen optionalen Ausdruck, der zum Aktivieren oder Deaktivieren des Blinkens verwendet werden kann. Die „Steady Color“ (Feste Farbe) ist die Farbe, die bei deaktiviertem Blinken angezeigt wird.
- Mit den Eigenschaften unter *Colors* (Farben) können Sie die zu verwendenden Farben definieren.

8.3.4 Definieren von Zweifachstatus-Farben

Zweifachstatus-Farben werden über das folgende Dialogfeld definiert:



- Mit der Eigenschaft *Value* (Wert) wird die anzuzeigende Farbe ausgewählt.
- Mit den Eigenschaften unter *Colors* (Farben) können Sie die zu verwendenden Farben definieren.

8.3.5 Definieren von Vierfachstatus-Farben

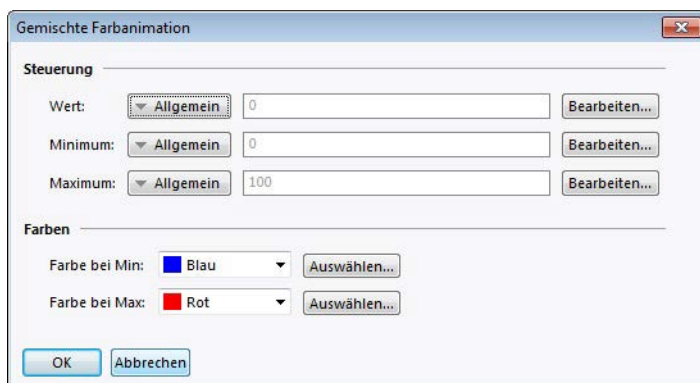
Vierfachstatus-Farben werden über das folgende Dialogfeld definiert:



- Mit den Eigenschaften *Value* (Wert) wird die anzuzeigende Farbe ausgewählt.
- Mit den Eigenschaften unter *Colors* (Farben) können Sie die zu verwendenden Farben definieren.

8.3.6 Definieren von gemischten Farben

Gemischte Farben werden über das folgende Dialogfeld definiert:



- Mit den Eigenschaften *Value* (Wert), *Minimum* und *Maximum* wird die anzuzeigende Farbe ausgewählt. Im gezeigten Beispiel ist die Farbe blau, wenn der Tag-Wert kleiner oder gleich dem Mindestwert ist, und rot, wenn der Tag-Wert größer oder gleich dem Höchstwert ist. Wenn sich der Tag-Wert zwischen den Grenzwerten ändert, wird ein sanfter Übergang von Blau nach Rot erzeugt.
- Mit den Eigenschaften unter *Colors* (Farben) können Sie die zu verwendenden Farben definieren.

8.3.7 Definieren von Behälterfüllungen

Viele geometrische Primitive unterstützen eine sogenannte „Behälterfüllungs“-Option, bei der die Form je nach dem Inhalt eines Tags bis zu einem bestimmten Pegel gefüllt wird. Mit dieser Funktion können einfache Balkendiagramme implementiert oder komplexere Formen gefüllt werden.

Das folgende Beispiel zeigt einen sechszackigen Stern mit einer 60%igen Behälterfüllung von unten nach oben.



Behälterfüllungen werden mithilfe der Einstellungen eines Primitives unter „Fill Behavior“ (Füllverhalten) definiert:

Füllverhalten

Füllmodus:

Wert: Tag1

Minimum: Tag1.Min

Maximum: Tag1.Max

- Die Eigenschaft *Fill Mode* (Füllmodus) legt fest, ob eine Behälterfüllung gezeichnet wird und aus welcher Richtung die Füllung erfolgen soll. Füllungen können an jeder Kante des Primitives beginnen, sodass komplexe Animationen erstellt werden können. Im Modus „Block“ wird die Form mit einem einzelnen Muster gefüllt, und Behälterfüllungen werden deaktiviert.
- Die Eigenschaft *Value* (Wert) gibt den Wert an, der zur Berechnung der Füllhöhe verwendet wird. Wenn ein Tag eingegeben wird, werden die Werte unter „Minimum“ und „Maximum“ automatisch auf die Dateneingabegrenzwerte dieses Tags gesetzt. Zur Angabe dieser Grenzwerte wird die Syntax für Ausdrücke von Tag-Eigenschaften verwendet. Die Eigenschaft „Value“ (Wert) kann eine Ganzzahl oder ein Gleitkommawert sein. Die Berechnung des Füllstands wird immer mit Gleitkommazahlen durchgeführt.
- Die Werte unter *Minimum* und *Maximum* definieren die Grenzwerte, die bei der Skalierung der Eigenschaft „Value“ (Wert) zur Berechnung der Füllhöhe verwendet werden sollen.

8.3.8 Definieren von Füllformaten

Die Füllformateigenschaften eines Primitives geben an, wie der Innenbereich des Primitives gefüllt wird:

Füllformat

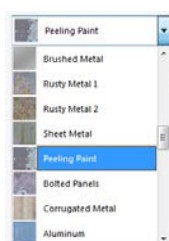
Muster:

Farbe 1: Benutzerdef.03

Farbe 2: Weiß

Farbe 3: Grau12

- Die Eigenschaft *Pattern* (Muster) legt das Füllmuster fest. Die Standardoption ist „Solid Color“ (Einfarbig), doch es stehen auch verschiedene Muster zur Auswahl. Darüber hinaus können Sie eine Textur auswählen, um eine realistische Darstellung eines realen Objekts zu erzeugen. In der folgenden Abbildung sehen Sie einige der verfügbaren Texturen:



- Außerdem sind mehrere abgestufte Füllungen verfügbar:

Muster	Beschreibung
Abgestufte Füllung 1	Farbe 1 oben und unten im Primitiv, vertikaler Wechsel zu Farbe 2 in der Mitte.
Abgestufte Füllung 2	Farbe 1 oben im Primitiv, vertikaler Wechsel zu Farbe 2 unten.
Abgestufte Füllung 3	Farbe 1 links und rechts im Primitiv, horizontaler Wechsel zu Farbe 2 in der Mitte.
Abgestufte Füllung 4	Farbe 1 links im Primitiv, horizontaler Wechsel zu Farbe 2 rechts.
Glänzend und chromfarben	Ein Metallic-Effekt mit den beiden angegebenen Farben.
Horizontaler Zylinder	Ein Effekt, der einen horizontal ausgerichteten Zylinder in Farbe 1 erstellt und mit Farbe 2 akzentuiert.
Vertikaler Zylinder	Ein Effekt, der einen vertikal ausgerichteten Zylinder in Farbe 1 erstellt und mit Farbe 2 akzentuiert.

- Die Eigenschaft *Color 1* (Farbe 1) definiert die erste Farbe, die für die Füllung verwendet werden soll.
- Die Eigenschaft *Color 2* (Farbe 2) definiert eine optionale zweite Farbe für die Füllung.
- Die Eigenschaft *Color 3* (Farbe 3) definiert die Hintergrundfarbe einer Behälterfüllung. Sie ist nicht erforderlich, wenn eine Blockfüllung verwendet wird. Die Eigenschaft ist möglicherweise nicht vorhanden, wenn das aktuelle Primitiv keine Behälterfüllungen unterstützt.

8.3.9 Ältere Füllformate

Für die von Crimson 3.0 unterstützten älteren Primitive wird ein einfacheres Füllformat unterstützt:

Die Eigenschaften stimmen mit den Eigenschaften für das obige komplexere Füllformat überein, doch es stehen weniger Muster zur Auswahl. Insbesondere werden weder Texturen noch metallische und zylindrische Abstufungen unterstützt und es sind weniger Schraffurmuster verfügbar.

8.3.10 Definieren von Randformaten

Die Randformateigenschaften eines Primitivs geben an, wie der Rand des Primitivs gezeichnet wird:

Randformat

Breite: Pixel

Muster:

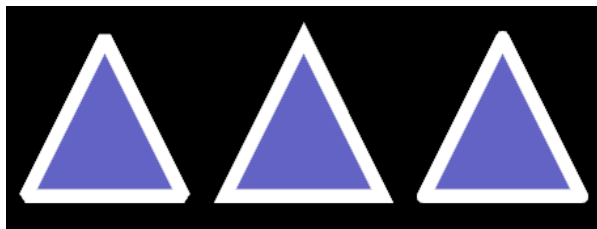
Farbe 1: Weiß

Farbe 2:

Kantenmodus:

Verbindungsmodus:

- Die Eigenschaft *Width* (Breite) definiert, wie breit der Rand sein soll. Mit dem Wert Null wird der Rand vollständig deaktiviert. Bei älteren Primitiven ist eine begrenzte Anzahl an Randgrößen verfügbar. Neuere Primitive unterstützen beliebige Breiten bis 80 Pixel.
- Die Eigenschaft *Pattern* (Muster) definiert, wie der Rand gefüllt wird. Bei älteren Primitiven musste der Rand einfarbig sein. Bei neueren Primitiven sind Ränder mit abgestuften Füllungen, Mustern oder Texturen möglich. Die Einstellungen sind analog zu den oben definierten Einstellungen für das Füllformat eines Primitivs.
- Die Eigenschaft *Color 1* (Farbe 1) definiert die erste Farbe, die für den Rand verwendet werden soll.
- Die Eigenschaft *Color 2* (Farbe 2) definiert eine optionale zweite Farbe für den Rand.
- Die Eigenschaft *Edge Mode* (Randmodus) definiert, wo der Rand relativ zur Begrenzung des gefüllten Teils der Form platziert wird. Die Ränder können innerhalb oder außerhalb einer Form platziert oder so gezeichnet werden, dass sie auf der Begrenzung der Form zentriert sind. Diese Einstellung hat einen subtilen Einfluss darauf, wie Behälterfüllungen verarbeitet werden, sowie auf die Position des Randes im Verhältnis zum Begrenzungsrahmen des Primitivs. Sie ist nur für Randbreiten von mindestens drei Pixeln verfügbar. Schmalere Ränder werden automatisch positioniert.
- Die Eigenschaft *Join Mode* (Verbindungsmodus) definiert, wie der Rand an der Außenseite der einzelnen Ecken einer Form gezeichnet wird. Im Modus „Bevel“ (Abschrägung) werden die Ecken abgeflacht. Im Modus „Miter“ (Gehrung) wird jeder Rand zu einer Spitze verlängert. Gehrungen, die sich zu weit erstrecken, werden jedoch abgeschnitten, um unattraktive Ergebnisse zu vermeiden. Im Modus „Rounded“ (Gerundet) werden die Ecken gerundet, wobei der Radius der Randbreite entspricht. In der folgenden Abbildung sehen Sie, wie ein Dreieck mit einem 10 Pixel breiten Rand in jedem der drei Modi aussieht.



8.3.11 Definieren des Randbeschnitts

Mit den Randbeschnitteigenschaften eines Primitivs kann auf beiden Seiten des Randes eine optionale Linie gezeichnet werden:

Kantenbeschnidung

Breite der Beschnidung: Pixel

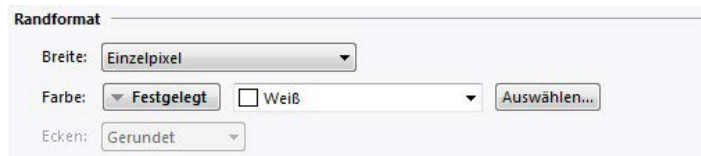
Kanten beschniden:

Farbe der Beschnidung:

- Die Eigenschaft *Trim Width* (Breite des Beschnitts) gibt an, wie breit der Beschnitt ist. Mit dem Standardwert Null wird der Beschnitt deaktiviert. Beschnitte werden häufig bei Rändern mit komplexeren Stilen wie beispielsweise Texturen angewendet. Eine Beschnittlinie an der Außenseite des Randes begrenzt den Rand deutlicher, damit er sich besser vom Hintergrund abhebt.
- Die Eigenschaft *Trim Edges* (Beschnittkanten) gibt an, auf welcher Seite des Randes der Beschnitt angewendet werden soll. Sie kann auf die innere Kante, die äußere Kante oder beide Kanten angewendet werden.
- Die Eigenschaft *Trim Color* (Farbe des Beschnitts) gibt die Farbe des Beschnitts an.

8.3.12 Ältere Randformate

Bei älteren Primitiven wird ein einfacheres Randformat unterstützt:



Randformat

Breite: Einzelpixel

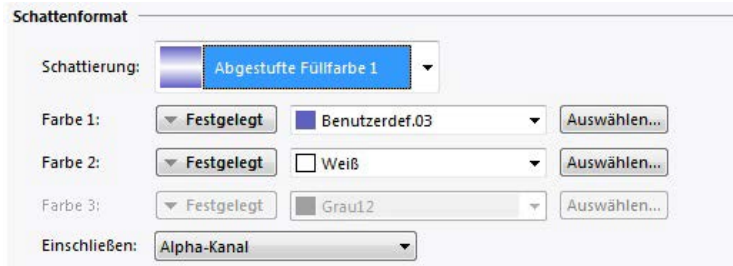
Farbe: Festgelegt Weiß

Ecken: Gerundet

- Die Eigenschaft *Width* (Breite) gibt an, wie breit der Rand ist. Der Rand kann angezeigt werden, indem Sie den Wert „None“ (Kein) wählen. Bei älteren Primitiven unterstützt Crimson nur ungerade Randgrößen mit einer Breite von bis zu 9 Pixel.
- Die Eigenschaft *Color* (Farbe) gibt die Farbe des Randes an.
- Die Eigenschaft *Corners* (Ecken) ist nur für Rechtecke verfügbar und definiert, ob beim Zeichnen des Randes gerundete oder spitzen Ecken verwendet werden sollen. Bei allen anderen Primitiven verwenden standardmäßig abgerundete Ecken verwendet.

8.4 Zuweisen neuer Farben zu Symbolen

Wenn Sie einer Seite ein Symbol hinzufügen, indem Sie es aus dem Bereich „Symbole“ des Ressourcenfensters ziehen, wird ein spezielles Primitiv erstellt, welches das resultierende Bild aufnimmt. Dieses einfache Bildprimitiv unterstützt alle Bildanpassungsfunktionen, die im nächsten Kapitel beschrieben werden. Es unterstützt auch die Zuweisung neuer Farben zum Bild. Dies bedeutet, dass die im Symbol vorhandenen Farben durch eine einzelne Farbe oder eine abgestufte Füllung ersetzt werden können. Die Zuweisung neuer Farben wird über die Eigenschaften „Shade Format“ (Farbtonformat) auf der Registerkarte „Shading“ (Schattierung) des einfachen Bildprimitivs konfiguriert.



- Die Eigenschaften *Shading* (Schattierung), *Color 1* (Farbe 1), *Color 2* (Farbe 2) und *Color 3* (Farbe 3) sind wie oben beschrieben.
- Die Eigenschaft *Include* (Einschließen) definiert, wie die neue Farbe von Crimson angewendet werden soll. Wenn „Alpha Channel“ (Alphakanal) ausgewählt ist, wird die neue Farbe auf alle Pixel angewendet, die nicht vollständig transparent sind. Dadurch werden alle internen Details des Symbols entfernt. Diese Option wird meist für Symbole der Kategorie „Basic Shapes“ (Einfache Formen) verwendet. Bei den anderen Einstellungen wird mindestens einer der roten, grünen und blauen Kanäle ausgewählt, die die Farben der einzelnen Pixel codieren, aus denen das Symbol besteht. Anhand der Sättigung der ausgewählten Kanäle wird die Sättigung der neuen Farbe für jedes Pixel definiert. Hierdurch bleiben die internen Details des Symbols erhalten, aber die Farbe wird dennoch geändert. Dieses Verhalten ist schwer zu erklären. Daher sollten Sie mit dieser Einstellung experimentieren, bis Sie mit dem Ergebnis zufrieden sind.

Eine Behälterfüllungsoption wird ebenfalls unterstützt und wie oben beschrieben konfiguriert. Wenn Sie diese Option mit Symbolen aus der Kategorie „Tank Cutaways“ (Behälter-Schnittmodelle) kombinieren, können Sie eine attraktive Anzeige des Füllstands eines Behälters erstellen.

8.5 Verwenden von Gruppen

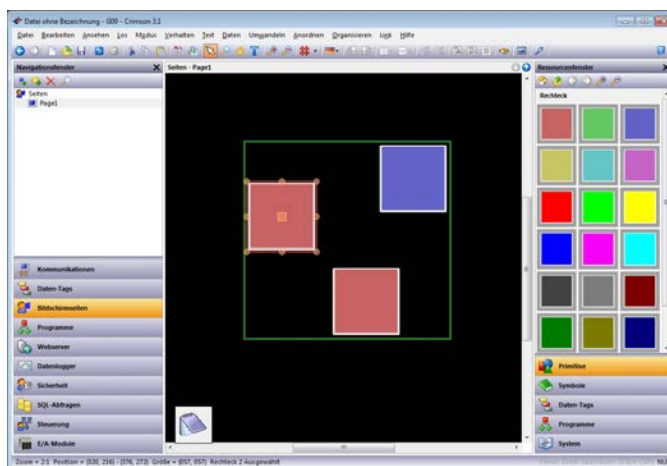
Eine Gruppe ist eine Sammlung von Primitiven, die als einzelnes Objekt behandelt werden.

8.5.1 Erstellen und Auflösen von Gruppen

Wenn Sie mehrere Primitive zu einer Gruppe zusammenfassen möchten, können Sie sie wie oben beschrieben auswählen und dann im Menü „Organize“ (Organisieren) den Befehl „Group“ (Gruppieren) wählen. Alternativ können Sie auch die Tastenkombination **STRG+G** verwenden. Sobald eine Gruppe erstellt wurde, können Sie sie ganz so wie ein einzelnes Objekt verschieben, kopieren und die Größe ändern. Eine Gruppe kann in ihre Bestandteile aufgelöst werden. Wählen Sie dazu zunächst die Gruppe aus. Wählen Sie dann den Befehl „Ungroup“ (Gruppierung aufheben) aus oder verwenden Sie die Tastenkombination **STRG+U**. Beachten Sie, dass Gruppen sowohl Primitive als auch weitere Gruppen umfassen können und dass Gruppen innerhalb vernünftiger Grenzen verschachtelt werden können.

8.5.2 Bearbeiten von Gruppen

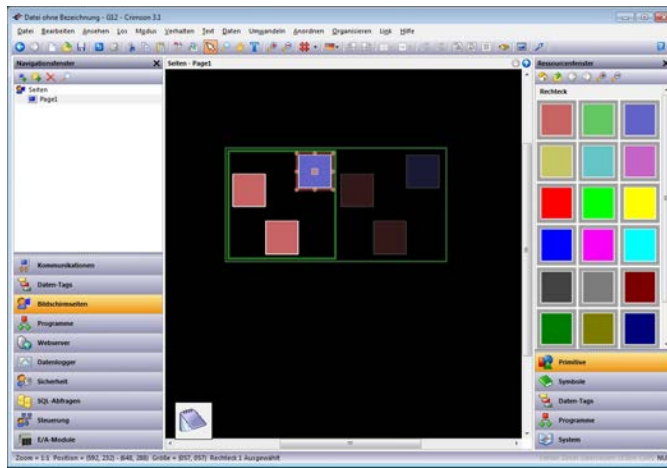
Sobald eine Gruppe erstellt wurde, möchten Sie gegebenenfalls deren Inhalt bearbeiten, ohne sie zuvor aufzulösen. Dies ist besonders nützlich, wenn Sie verschachtelte Gruppen erstellt haben, denn die erneute Verschachtelung kann sich sehr kompliziert gestalten. Um eine Gruppe zu bearbeiten, wählen Sie zunächst diese Gruppe aus und klicken dann auf eines ihrer Mitglieder. (Klicken Sie nicht auf den Griff in der Mitte des Gruppenobjekts. Dieser Griff dient dazu, die ganze Gruppe zu verschieben oder auszuwählen.) Sobald das Gruppenmitglied ausgewählt wurde, wechselt Crimson in den Gruppenbearbeitungsmodus, wie in der nachstehenden Abbildung gezeigt:



Die Gruppe, die bearbeitet wird, ist von einem grünen Rechteck umgeben. Die Bearbeitung einer Gruppe funktioniert im Wesentlichen wie die Bearbeitung einer Seite. Allerdings ist es nicht möglich, Elemente über die Gruppenbegrenzungen hinaus zu verschieben. Elemente können kopiert, eingefügt, in der Größe verändert und gelöscht werden. Tatsächlich können alle üblichen Vorgänge durchgeführt werden. Sie können sogar neue Elemente aus dem Ressourcenfenster ziehen und in einer Gruppe ablegen. Wenn Sie den Gruppenmodus beenden möchten, klicken Sie außerhalb der Gruppe oder drücken die Taste **Esc**.

8.5.3 Bearbeiten verschachtelter Gruppen

In Crimson ist auch die Bearbeitung von Gruppen möglich, die sich selbst innerhalb von Gruppen befinden:



Zum Aktivieren dieser Funktion beginnen Sie zuerst mit der Bearbeitung der äußeren Gruppe. Wählen Sie dann die innere Gruppe aus und klicken Sie auf ein Mitglied dieser inneren Gruppe. Im oben gezeigten Beispiel sehen Sie, wie die Gruppenhierarchie anhand einer Reihe von verblassten Rechtecken dargestellt wird. Beachten Sie außerdem, dass Elemente außerhalb der aktuellen Gruppen in verblassten Farben angezeigt werden, um zu verdeutlichen, wo die Gruppe endet. Wenn Sie die Bearbeitung verschachtelter Gruppen mit der Taste Esc beenden, gelangen Sie bei jeder Betätigung der Taste eine Ebene weiter nach oben.

8.5.4 Erweitern von Gruppen

Wie oben erwähnt ist die Verschiebung von Primitiven während der Gruppenbearbeitung nur begrenzt möglich, d. h. ein Primitiv kann nicht aus der Gruppe verschoben werden, der es angehört. Wenn Sie Primitive am Rand einer Gruppe anpassen möchten, können Sie im Kontextmenü der Gruppe den Befehl „Expand and Edit“ (Erweitern und bearbeiten) auswählen. Dadurch werden die Gruppenbegrenzungen von den Primitiven weg nach außen verschoben, damit Sie die gewünschten Anpassungen vornehmen können. Wenn der Gruppenbearbeitungsmodus abgebrochen wird, wird die Gruppenbegrenzung nach innen verschoben, damit sie wieder an den Inhalten anliegt.

8.6 Hinzufügen von Bewegungen zu Primitiven

Jedes Primitiv kann so animiert werden, dass es sich innerhalb eines von Ihnen definierten Begrenzungsrechtecks dynamisch bewegt. Primitive können horizontal, vertikal oder in beiden Dimensionen bewegt werden. Sie können auch anhand von Polarkoordinaten so bewegt werden, dass sie in einem variablen Abstand um einen Punkt kreisen. Alle Dimensionen werden anhand eines Steuerungswerts und zweier Grenzwerte definiert.

Um eine Bewegung anzuwenden, wählen Sie ein oder mehrere Primitive aus und wählen Sie anschließend im Menü „Behavior“ (Verhalten) einen der Befehle für „Add Movement“ (Bewegung hinzufügen). Um die Primitive erscheint ein rotes Rechteck, das die Bewegungsgruppe darstellt, in der die Animation stattfindet. Die Größe der Gruppe kann geändert werden, um die Ausdehnung der Animation zu ändern. Anders als bei einer normalen Gruppe hat die Änderung der Größe einer Bewegungsgruppe keine Auswirkungen auf die Größe der Primitive selbst. Der Inhalt der Gruppe kann genau wie bei Standardgruppen mit den bereits beschriebenen Verfahren bearbeitet werden. Bei der Bearbeitung der Polarbewegung stellt eine Ellipse den Pfad dar, dem die Primitive folgen, wenn der Radius auf 100 % eingestellt ist. Beachten Sie, dass diese Ellipse grundsätzlich kleiner ist als die Gruppe selbst. Dies liegt daran, dass sie die Position der Mitte der animierten Elemente kennzeichnet und dass genügend Platz übrig bleiben muss, um sicherzustellen, dass die Elemente nicht über die Gruppenbegrenzung hinausgehen.

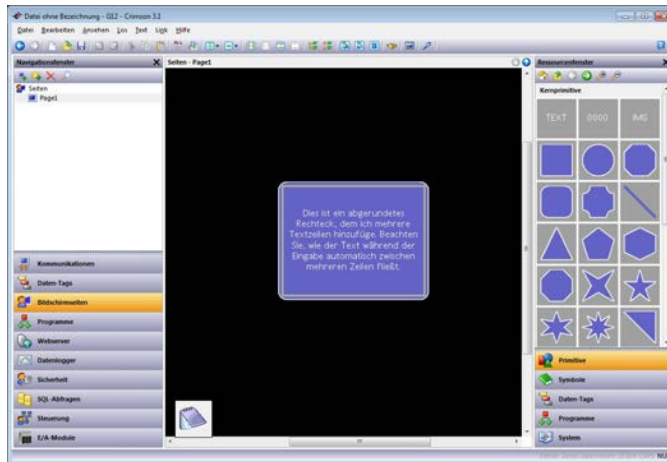
Um festzulegen, wie die Bewegung gesteuert wird, öffnen Sie die Eigenschaften der Bewegungsgruppe:



Im oben gezeigten Beispiel geht es um die Konfiguration der 2D-Bewegung. Die Polarbewegung wird auf ähnliche Weise konfiguriert. Für jede Bewegungsdimension definiert der Wert *Position*, wo der Inhalt der Gruppe im Verhältnis zu ihrer Kontur platziert wird. Die Werte *Minimum* und *Maximum* stellen die Grenzwerte der Steuerungswerte dar. Bei 2D-Bewegungen führen die minimalen Einstellungen dazu, dass der Gruppeninhalt in der linken oberen Ecke angezeigt wird.

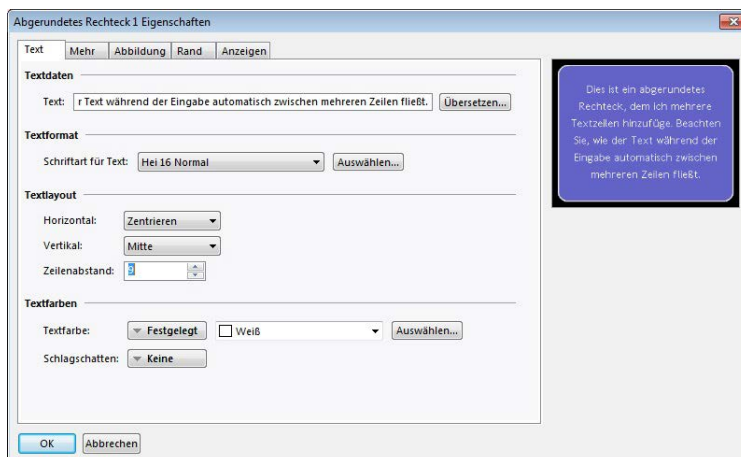
8.7 Hinzufügen von Text zu Primitiven

Die meisten Primitive von Crimson können Text aufnehmen. Um einem Primitiv Text hinzuzufügen, wählen Sie einfach das Primitiv aus, drücken die Taste F2 und beginnen mit der Eingabe. Alternativ können Sie mit der rechten Maustaste auf das Primitiv klicken und im daraufhin angezeigten Menü den Befehl „Add Text“ (Text hinzufügen) auswählen. In der nachstehenden Abbildung wird Text in ein abgerundetes Rechteck eingegeben:



Beachten Sie, dass das Begrenzungsrechteck des Primitivs in Gelb angezeigt wird und dass alle anderen Primitive auf der Seite verblasst sind. Beachten Sie auch, dass der Texteditor einen automatischen Zeilenumbruch durchführt. Versuchen Sie einmal selbst, die Größe eines Primitivs mit Text zu ändern. Sie werden feststellen, dass Crimson den Text automatisch so angepasst, dass er in die neue Form passt.

Während der Textbearbeitung ändert sich die Symbolleiste. Sie enthält dann Befehle zum Ändern der Textausrichtung und zum Vergrößern oder Verkleinern des Zeilenabstands. Die etwas anspruchsvolleren Texteingenschaften können durch Auswählen von „Text Properties“ (Texteigenschaften) im Kontextmenü eines Primitivs oder durch Drücken der Tasten ALT+EINGABE im Textbearbeitungsmodus aufgerufen werden:



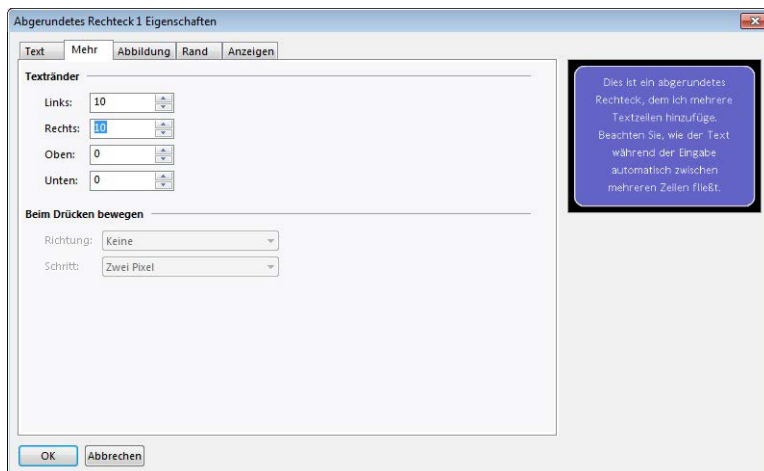
8.7.1 Texteigenschaften

- Die Eigenschaft *Text* enthält den anzuzeigenden Text. Zur Codierung harter Zeilenumbrüche werden Senkrechtstriche verwendet. Da dieses Feld eine übersetzbare Zeichenfolge ist, können mehrsprachige Versionen bearbeitet werden. Dies bedeutet auch, dass die Eigenschaft auf einen Ausdruck gesetzt und der Inhalt somit dynamisch geändert werden kann. Crimson unterstützt einen

vollständig dynamischen Rückfluss und ermöglicht dadurch komplexe und attraktive Präsentationsoptionen.

- Mit der Eigenschaft *Text Font* (Schriftart) kann die gewünschte Schrift ausgewählt werden. Die neue Standardschriftart von Crimson ist Hei, eine Unicode-Schriftart, die vereinfachtes Chinesisch und die meisten anderen Sprachen unterstützt. Über die Schaltfläche „Pick“ (Auswählen) können Sie das Schriftauswahl-Dialogfeld aufrufen. Jede auf Ihrem System installierte Schriftart kann in einer Form wiedergegeben werden, die vom Zielgerät verwendet werden kann. Es liegt in Ihrer Verantwortung, sich zu vergewissern, dass Sie für diese Art der Schriftartnutzung ausreichend lizenziert sind.
- Die Eigenschaft *Horizontal* (Horizontal) definiert die horizontale Ausrichtung des Textes.
- Die Eigenschaft *Vertical* (Vertikal) definiert die vertikale Ausrichtung des Textes.
- Die Eigenschaft *Line Spacing* (Zeilenabstand) definiert zusätzlichen Zeilenabstand in Pixeln.
- Die Eigenschaft *Text Color* (Textfarbe) dient zur Auswahl der Textfarbe.
- Mit der Eigenschaft *Drop Shadow* (Schlagschatten) wird ein optionaler Schatten an der rechten und unteren Kante des Textes aktiviert. Dieser Effekt ist nützlich, wenn Sie möchten, dass sich der Text von seinem Hintergrund abhebt, insbesondere wenn der Hintergrund ein Bild mit vielen Farben ist.

8.7.2 Weitere Eigenschaften

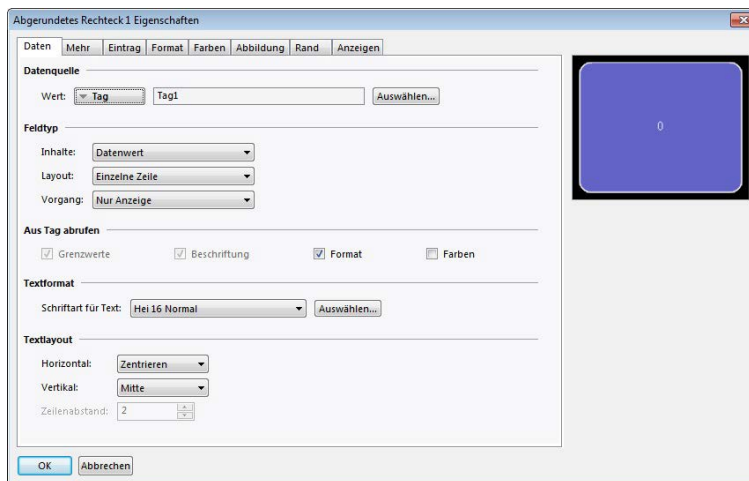


- Mit den Eigenschaften unter *Text Margins* (Textträger) können Sie den leeren Bereich um den Text im Verhältnis zu dem vom Primitiv bereitgestellten Textbegrenzungsrahmen steuern. Dies kann nützlich sein, wenn Sie den Text innerhalb des Primitivs neu positionieren möchten oder eine bessere visuelle Zentrierung bei der Arbeit mit Schriften wünschen, die über oder unter ihren Zeichen viel Platz für diakritische Zeichen oder Unterlängen bieten.
- Die Eigenschaft *Direction* (Richtung) definiert die Richtung, in die der Text bei einem Klick auf das zugehörige Primitiv verschoben wird. Sie ist nur aktiviert, wenn dem Primitiv eine Aktion zugewiesen ist oder wenn es sich bei dem Primitiv um so etwas wie eine Schaltfläche handelt, der eine inhärente Aktion zugeordnet ist. Diese Option ist beim Erstellen benutzerdefinierter Schaltflächen nützlich, die bei einer Berührung Feedback geben sollen.
- Die Eigenschaft *Step* (Schritt) gibt an, wie weit der Text bei einem Klick auf das Primitiv verschoben werden soll. Je nach gewünschtem Effekt können ein bis drei Pixel ausgewählt werden.

8.8 Hinzufügen von Daten zu Primitiven

Primitive, die das Hinzufügen von Text unterstützen, unterstützen auch die Anzeige von Live-Daten und können optional für die Dateneingabe konfiguriert werden. Um einem Primitiv Daten hinzuzufügen, klicken Sie mit der rechten Maustaste auf das Primitiv und wählen im daraufhin angezeigten Menü den Befehl „Add Data“ (Daten hinzufügen) aus. Alternativ können Sie das Primitiv auswählen und die Tastenkombination **STRG+F2** drücken. Das Dialogfeld „Eigenschaften“ des Primitivs wird angezeigt. Es enthält eine Reihe zusätzlicher Registerkarten, mit denen Sie das Datenelement und sein Verhalten definieren können.

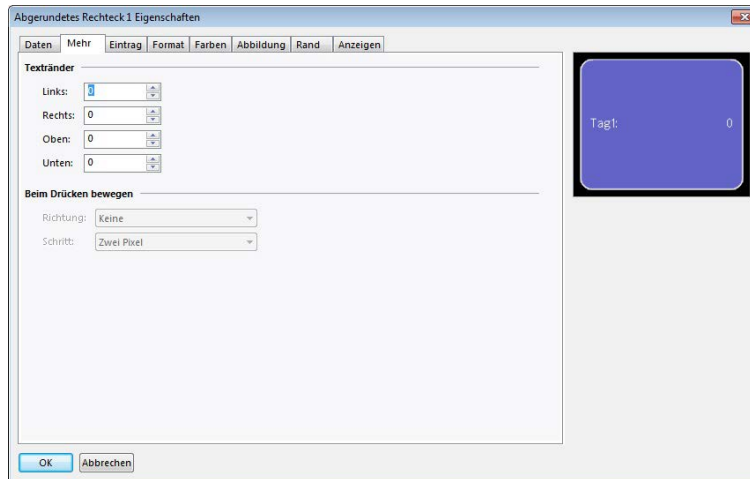
8.8.1 Dateneigenschaften



- Die Eigenschaft *Value* (Wert) definiert den anzuzeigenden Datenwert.
- Die Eigenschaft *Contents* (Inhalte) legt fest, ob das Feld den Datenwert, den Datenwert nebst zugehöriger Bezeichnung oder nur die Bezeichnung anzeigen soll.
- Die Eigenschaft *Layout* definiert, wie die ausgewählten Daten formatiert werden sollen. Bei der Einstellung „Single Line“ (Einzelne Zeile) wird alles in einer einzigen Zeile platziert. Wenn der Text zu groß ist, fließt er über den verfügbaren Platz hinaus. Bei der Einstellung „Multiple Lines“ (Mehrere Zeilen) wird der Text auf mehrere Zeilen verteilt, sofern dies notwendig ist, um ihn im Primitiv unterzubringen. In diesem Modus wird die Bezeichnung immer in einer eigenen Zeile platziert, getrennt von den zugehörigen Daten.
- Die Eigenschaft *Operation* (Vorgang) definiert, ob das Feld nur den Wert anzeigen oder auch die Dateneingabefunktionalität bereitstellen soll. Die Dateneingabe ist nur verfügbar, wenn der ausgewählte Datenwert schreibbar ist.
- Die Eigenschaften unter *Get from Tag* (Aus Tag abrufen) definieren, ob bestimmte Eigenschaften des Datenfelds lokal definiert oder mit den Eigenschaften des angezeigten Tags verknüpft sind. Die Optionen sind nur verfügbar, wenn unter „Value“ (Wert) ein Tag angegeben ist.
- Mit der Eigenschaft *Text Font* (Schriftart) kann die gewünschte Schrift ausgewählt werden. Die neue Standardschriftart von Crimson ist Hei. Dies ist eine Unicode-Schriftart, die vereinfachtes Chinesisch und die meisten anderen Sprachen unterstützt. Über die Schaltfläche „Pick“ (Auswählen) können Sie das Schriftauswahl-Dialogfeld aufrufen. Jede auf Ihrem System installierte Schriftart kann in einer Form wiedergegeben werden, die vom Zielgerät verwendet werden kann. Es liegt in Ihrer Verantwortung, sich zu vergewissern, dass Sie für diese Art der Schriftartnutzung ausreichend lizenziert sind.

- Die Eigenschaft *Horizontal* (Horizontal) definiert die horizontale Ausrichtung des Textes.
- Die Eigenschaft *Vertical* (Vertikal) definiert die vertikale Ausrichtung des Textes.
- Die Eigenschaft *Line Spacing* (Zeilenabstand) definiert, wie viele zusätzliche Pixel bei der Arbeit im Layoutmodus „Multiple Lines“ (Mehrere Zeilen) zwischen Zeilen eingefügt werden sollen.

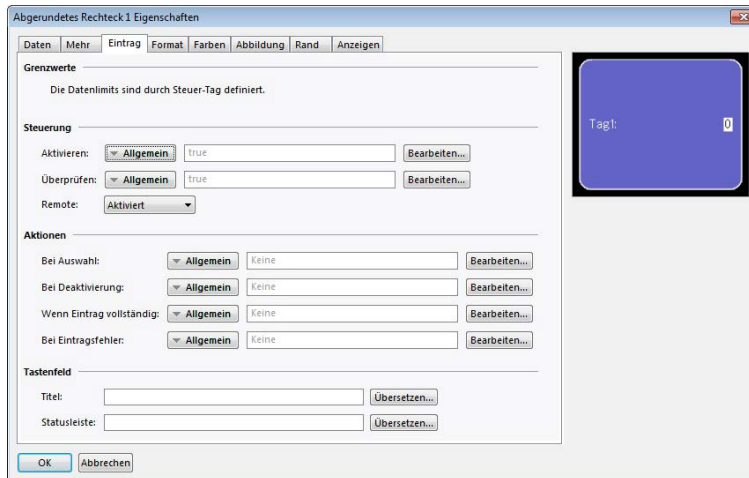
8.8.2 Weitere Eigenschaften



- Mit den Eigenschaften unter *Text Margins* (Textränder) können Sie den leeren Bereich um den Text im Verhältnis zu dem vom Primitiv bereitgestellten Textbegrenzungsrahmen steuern. Dies kann nützlich sein, wenn Sie den Text innerhalb des Primitivs neu positionieren möchten oder eine bessere visuelle Zentrierung bei der Arbeit mit Schriften wünschen, die über oder unter ihren Zeichen viel Platz für diakritische Zeichen oder Unterlängen bieten.
- Die Eigenschaft *Direction* (Richtung) definiert die Richtung, in die der Text bei einem Klick auf das zugehörige Primitiv verschoben wird. Sie ist nur aktiviert, wenn dem Primitiv eine Aktion zugewiesen ist oder wenn es sich bei dem Primitiv um so etwas wie eine Schaltfläche handelt, der eine inhärente Aktion zugeordnet ist. Diese Option ist beim Erstellen benutzerdefinierter Schaltflächen nützlich, die bei einer Berührung Feedback geben sollen.
- Die Eigenschaft *Step* (Schritt) gibt an, wie weit der Text bei einem Klick auf das Primitiv verschoben werden soll. Je nach gewünschtem Effekt können ein bis drei Pixel ausgewählt werden.

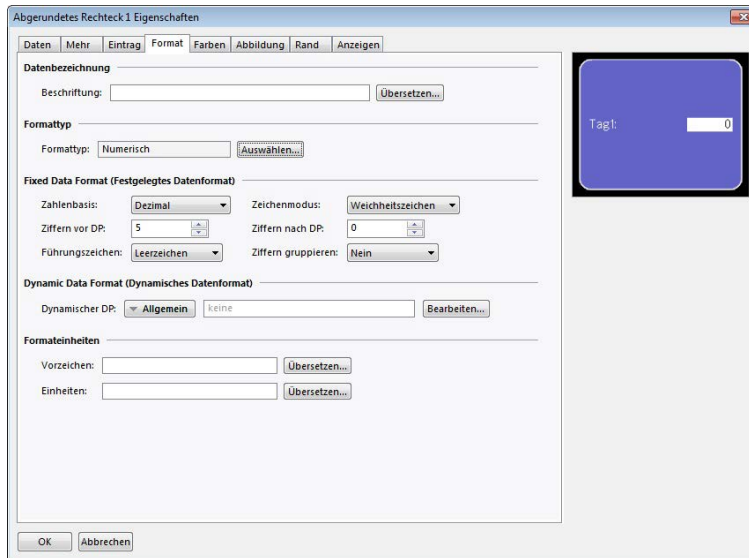
8.8.3 Eingabeeigenschaften

Diese Eigenschaften sind nur verfügbar, wenn die Dateneingabe aktiviert ist:



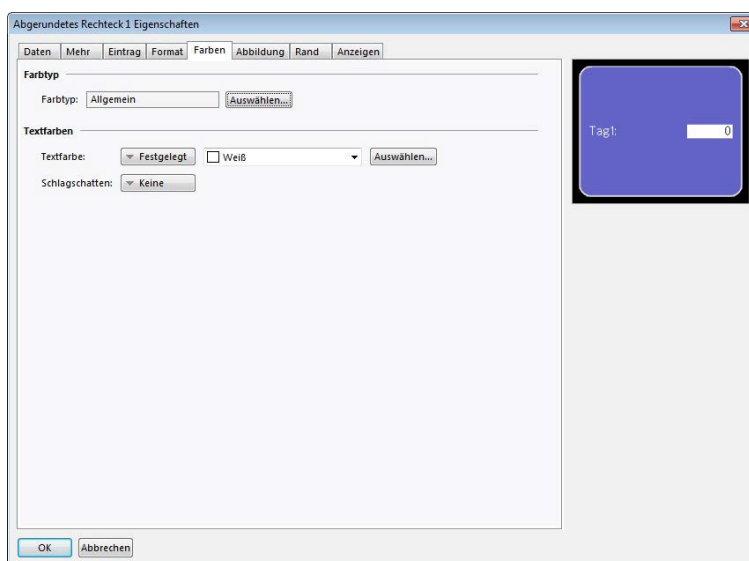
- Über die Eigenschaft *Enable* (Aktivieren) kann ein Ausdruck zum Aktivieren oder Deaktivieren der Dateneingabe bereitgestellt werden. Deaktivierte Dateneingabefelder verhalten sich genau wie schreibgeschützte Felder.
- Mit der Eigenschaft *Validate* (Überprüfen) können Sie einen Ausdruck definieren, der zur Prüfung von eingegebenen Werten verwendet wird. Der Ausdruck muss als nicht Null ausgewertet werden, um die Eingabe zu ermöglichen, oder als Null, um die Eingabe zu blockieren. Wenn Sie beispielsweise `Data%25==0` eingeben, können nur Vielfache von 25 eingegeben werden, weil dies die einzigen Werte sind, bei denen die Verringerung auf den Betrag 25 einen Nullwert ergibt. Während der Ausführung dieses Ausdrucks behält die spezielle Systemvariable `Data` den neu eingegebenen Wert bei.
- Die Eigenschaft *Remote* aktiviert oder deaktiviert die Fähigkeit, die Dateneingabe per Fernzugriff über den Webserver zu steuern.
- Die Eigenschaft *On Selected* (Bei Auswahl) gibt eine Aktion an, die ausgeführt werden soll, wenn der Benutzer kurz vor der Dateneingabe auf das Dateneingabefeld klickt.
- Die Eigenschaft *On Deselected* (Bei Aufheben der Auswahl) gibt eine Aktion an, die ausgeführt werden soll, wenn die Dateneingabe beendet wird, entweder weil ein Wert geschrieben, eine Seite geändert oder der Eingabeprozess vom Benutzer durch Klicken auf eine Schaltfläche abgebrochen wurde.
- Die Eigenschaft *On Entry Complete* (Bei Eingabeende) gibt eine Aktion an, die ausgeführt werden soll, wenn die Dateneingabe erfolgreich abgeschlossen wurde.
- Die Eigenschaft *On Entry Error* (Bei Eingabefehler) gibt eine Aktion an, die ausgeführt werden soll, wenn der Benutzer einen ungültigen Wert eingibt.
- Im Eigenschaftsfeld *Title* (Titel) können Sie einen Titel eingeben, der oben im Tastenfeld angezeigt wird, wenn dieses aktiviert ist.
- Im Eigenschaftsfeld *Status Bar* (Statusleiste) können Sie Anmerkungen oder Informationen eingeben, die unten im Tastenfeld angezeigt werden, wenn dieses aktiviert ist.

8.8.4 Formateigenschaften



- Die Eigenschaft *Label* (Bezeichnung) bestimmt die Bezeichnung, die auf dieses Feld angewendet werden soll. Diese Eigenschaft ist möglicherweise nicht verfügbar, wenn die Bezeichnung nicht angezeigt wird oder wenn das Feld so konfiguriert ist, dass seine Bezeichnung vom Steuerungstag abgerufen wird.
- Das Feld *Format Type* (Formattyp) gibt den Formattyp an, der beim Anzeigen und optionalen Bearbeiten des Datenwerts verwendet werden soll. Auch dieses Feld ist möglicherweise nicht verfügbar, wenn das Format vom Steuerungstag abgerufen wird.
- Weitere Eigenschaften richten sich nach dem ausgewählten Datenformat. Weitere Informationen zu den Eigenschaften der einzelnen Formate finden Sie im Kapitel „Verwenden von Formaten“.

8.8.5 Farbeigenschaften

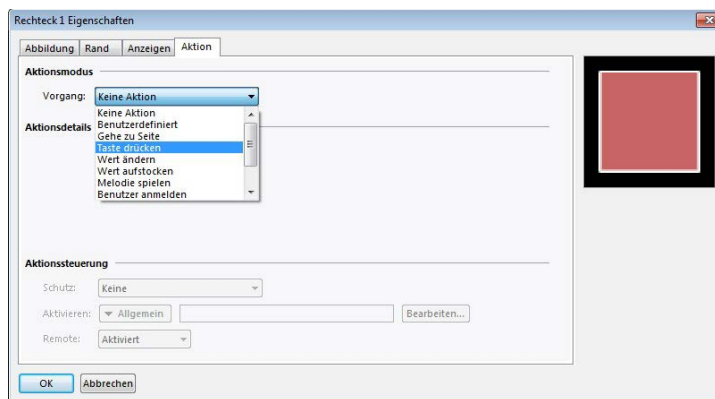


- Das Feld *Color Type* (Farbtyp) gibt die Färbung an, die bei der Anzeige des Datenwerts verwendet werden soll. Dieses Feld ist möglicherweise nicht verfügbar, wenn die Färbung vom Steuerungstag abgerufen wird.
- Mit der Eigenschaft *Text Color* (Textfarbe) wird die Farbe des Textes außer Kraft gesetzt, wenn die Färbung „General“ (Allgemein) verwendet wird.
- Mit der Eigenschaft *Drop Shadow* (Schlagschatten) wird ein optionaler Schatten an der rechten und unteren Kante des Textes aktiviert. Dieser Effekt ist nützlich, wenn Sie möchten, dass sich der Text von seinem Hintergrund abhebt, insbesondere wenn der Hintergrund ein Bild mit vielen Farben ist. Er ist nur bei der Färbung „General“ (Allgemein) verfügbar.
- Weitere Eigenschaften richten sich nach der ausgewählten Färbung. Weitere Informationen zu den Eigenschaften der einzelnen Färbungen finden Sie im Kapitel „Verwenden von Färbungen“.

8.9 Hinzufügen von Aktionen zu Primitiven

Primitive, die keine eigene implizite Aktion ausführen, unterstützen das Hinzufügen von benutzerdefinierten Aktionen, die ausgeführt werden, wenn der Bediener den Touchscreen berührt oder loslässt. Zum Hinzufügen einer Aktion wählen Sie entweder im Kontextmenü des Primitivs den Befehl „Add Action“ (Aktion hinzufügen) aus oder Sie wählen das Primitiv aus und drücken die Tastenkombination **STRG+I**.

Dem Dialogfeld „Eigenschaften“ des Primitivs wird die Registerkarte „Action“ (Aktion) hinzugefügt:



8.9.1 Schützen von Aktionen

Mit der Eigenschaft „Protection“ (Schutz) einer Aktion kann verhindert werden, dass eine Aktion versehentlich aufgerufen wird. Diese Funktion wird zusätzlich zu der vom Sicherheitssystem bereitgestellten Schutzfunktion ausgeführt und wird aufgerufen, bevor mit den damit verbundenen Aktionen begonnen wird. Die folgenden Schutzmodi stehen zur Verfügung:

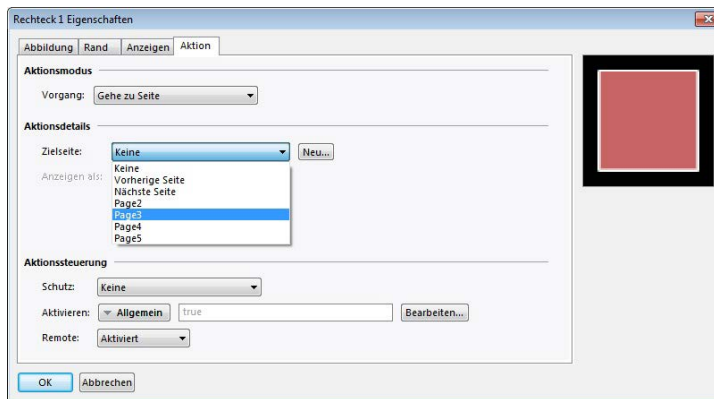
- Im Modus *Confirmed* (Bestätigt) wird ein Popup-Fenster mit der Aufforderung angezeigt, die Aktion zu bestätigen. Anschließend wird die Aktion sofort ausgeführt, sofern der Benutzer der Aktion zustimmt.
- Im Modus *Locked* (Gesperrt) wird ein Popup-Fenster mit dem Hinweis angezeigt, dass die Aktion gesperrt ist. Wenn der Benutzer der Aktion zustimmt, wird diese entsperrt. Der Benutzer muss die Aktion dann erneut aktivieren, damit sie tatsächlich stattfinden kann. Wenn eine andere Aktion ausgewählt wird, wird die vorherige Aktion gesperrt. Die Aktion wird auch bei einer Überschreitung der globalen Zeitüberschreitung gesperrt.
- Der Modus *Hard Locked* (Vollständig gesperrt) funktioniert ähnlich wie der Modus „Locked“ (Gesperrt). Allerdings wird die Aktion gleich nach ihrer Ausführung wieder gesperrt und muss jedes Mal entsperrt werden.

8.9.2 Aktivieren von Aktionen

Wenn Sie eine bestimmte Aktion vom Zustand „true“ einer Bedingung abhängig machen möchten, geben Sie im Feld *Enable* (Aktivieren) einen Ausdruck für diese Bedingung ein. Dieser Ausdruck kann direkt auf ein Markierungs-Tag verweisen oder einen der im Abschnitt „Schreiben von Ausdrücken“ definierten Vergleichs- oder logischen Operatoren verwenden. Wenn Sie eine komplexere Logik benötigen, bei der etwa eine von mehreren Aktionen auf der Basis von komplexeren Entscheidungen ausgeführt wird, konfigurieren Sie den Schlüssel für den Modus „User Defined“ (Benutzerdefiniert). Damit rufen Sie ein Programm auf, das die erforderliche Logik implementiert. Mit der Eigenschaft *Remote* können Sie zudem den Zugriff auf diese Aktion vom Webserver aus blockieren.

8.9.3 Die Aktion Gehe zu Seite

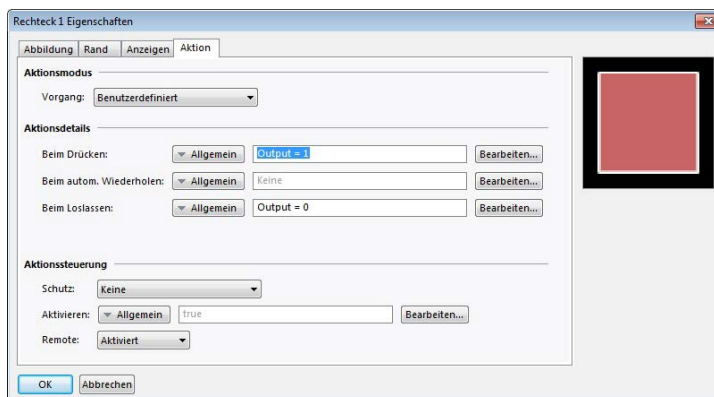
Mit dieser Aktion wird das Zielgerät angewiesen, eine neue Seite anzuzeigen:



- Mit der Eigenschaft *Target Page* (Zielseite) können Sie angeben, welche Seite angezeigt werden soll. Zusätzlich zu den Seiten in der Datenbank können Sie „Previous Page“ (Vorherige Seite) oder „Next Page“ (Nächste Seite) auswählen, um innerhalb der Seitenverlaufsliste zu navigieren. Über die Schaltfläche „New“ (Neu) kann eine neue Seite erstellt werden, ohne das Dialogfeld zu verlassen.
- Mit der Eigenschaft *Show As* (Anzeigen als) können Sie angeben, wie die Seite angezeigt werden soll. Bei Auswahl von „Normal Page“ (Normale Seite) wird die Seite auf die übliche Weise ausgewählt. Bei Auswahl von „Simple Popup“ (Einfaches Popup) werden die Primitive auf der neuen Seite über der aktuellen Seite in einem rechteckigen Popup angezeigt, das alle vorhandene Popups ersetzt. Bei der Einstellung „Nested Popup“ (Verschachteltes Popup) wird die Seite ebenfalls als Popup angezeigt, doch die neue Seite wird über einem etwaig vorhandenen Popup angezeigt. „Popup Menü“ (Popup-Menü) ist eine ältere Einstellung für Geräte mit Softkeys auf der linken Seite des Bildschirms. Derartige Geräte werden von Crimson 3.1 gegenwärtig nicht unterstützt. Ein Popup kann durch Ausführen der Funktion `HidePopup()` oder durch Auswahl von „Hide Popup“ (Popup ausblenden) oder „Hide All Popups“ (Alle Popups ausblenden) für die Eigenschaft „Operation“ (Vorgang) einer Aktion geschlossen werden.

8.9.4 Die benutzerdefinierte Aktion

Mit dieser Aktion werden eine oder mehrere benutzerdefinierte Aktionen ausgeführt:



- Die Eigenschaft *On Pressed* (Bei Drücken) definiert die Aktion, die beim Drücken des Primitivs ausgeführt werden soll. Diese Aktion kann beliebige Funktionen in der Funktionsreferenz oder die im Kapitel

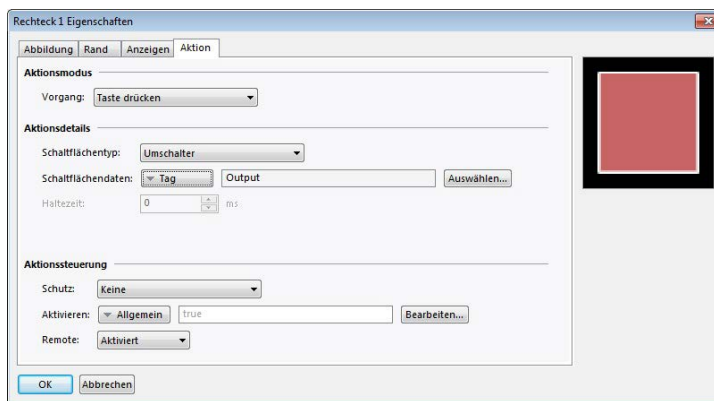
„Schreiben von Aktionen“ beschriebenen Datenänderungsoperatoren aufrufen. Sie kann auch ein Programm ausführen, um eine komplexere Aktion durchzuführen.

- Die Eigenschaft *On Auto-Repeat* (Bei automatischer Wiederholung) definiert die Aktion, die ausgeführt werden soll, wenn das Primitiv gedrückt und festgehalten wird. Die Aktion tritt sowohl beim ersten Drücken als auch bei nachfolgenden automatischen Wiederholungen auf. Es ist daher nicht notwendig, sowohl diese Eigenschaft als auch „On Pressed“ (Bei Drücken) zu definieren. Diese Aktion kann beliebige Funktionen aus der Funktionsreferenz oder die unter „Schreiben von Aktionen“ beschriebenen Datenänderungsoperatoren aufrufen. Sie kann außerdem ein Programm ausführen.
- Die Eigenschaft *On Released* (Bei Loslassen) definiert die Aktion, die beim Loslassen des Primitivs ausgeführt werden soll. Diese Aktion kann beliebige Funktionen aus der Funktionsreferenz oder die unter „Schreiben von Aktionen“ beschriebenen Datenänderungsoperatoren aufrufen. Sie kann außerdem ein Programm ausführen.

Im oben gezeigten Beispiel wird eine benutzerdefinierte Aktion für das Implementieren eines Druckknopfschalters verwendet.

8.9.5 Die Aktion Drucktaste

Diese Aktion wird für das Emulieren einer Drucktaste verwendet:



- Die Eigenschaft *Button Type* (Schaltflächentyp) wählt das gewünschte Verhalten aus:

Schaltflächentyp	Verhalten des Primitivs
Umschalten	Änderung des Datenstatus, wenn das Primitiv betätigt wird.
NEIN Vorübergehend	Daten werden auf 1 gesetzt, wenn das Primitiv betätigt wird. Daten werden auf 0 gesetzt, wenn das Primitiv losgelassen wird.
NC Vorübergehend	Daten werden auf 0 gesetzt, wenn das Primitiv betätigt wird. Daten werden auf 1 gesetzt, wenn das Primitiv losgelassen wird.
Einschalten	Daten werden auf 1 gesetzt, wenn das Primitiv betätigt wird.
Ausschalten	Daten werden auf 0 gesetzt, wenn das Primitiv betätigt wird.

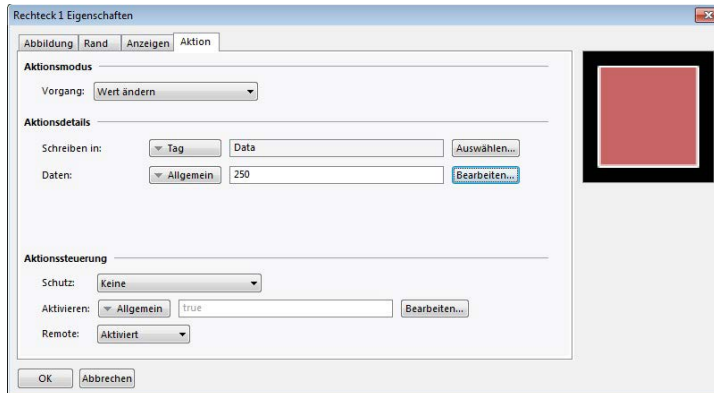
- Die Eigenschaft *Button Data* (Schaltflächendaten) definiert die Daten, die durch die Drucktaste geändert werden sollen.
- Mit der Eigenschaft *Hold Time* (Haltezeit) wird bei Drucktastern sichergestellt, dass die mit dem gedrückten Zustand assoziierten Daten zumindest eine bestimmte Zeit lang geschrieben werden, selbst wenn die Taste gedrückt und sofort losgelassen wird. Mit ihrer Hilfe kann sichergestellt werden, dass das externe Gerät die Tastenaktivierung unter Verhältnissen erkennt, in denen die Transaktions-schreibvorgänge von Crimson unzureichend sind oder deaktiviert wurden. Es wird davon abgeraten,

sich auf diese Eigenschaft zu verlassen, da bei ihrer Justierung die Kommunikationsverzögerungen im System berücksichtigt werden müssen.

Im oben gezeigten Beispiel wird der Wert des Tags `Output` beim Berühren des Primitivs umgeschaltet.

8.9.6 Die Aktion Wert ändern

Diese Aktion dient zum Schreiben eines numerischen Werts in ein Datenelement:

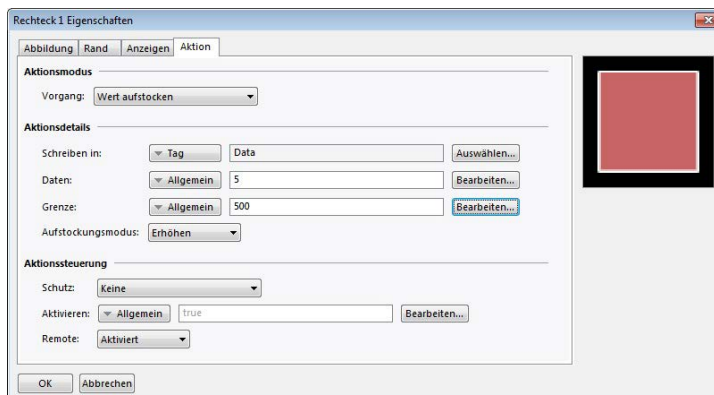


- Die Eigenschaft *Write To* (Schreiben in) definiert das zu ändernde Datenelement.
- Die Eigenschaft *Data* (Daten) definiert die zu schreibenden Daten.

Im oben gezeigten Beispiel wird das Tag `Data` beim Berühren des Primitivs auf 250 gesetzt. Beachten Sie, dass diese Aktion entweder Gleitkomma- oder ganzzahlige Werte unterstützt. Die Eigenschaft „Data“ (Daten) muss von einem Typ sein, der für das durch die Eigenschaft „Write To“ (Schreiben in) definierte Datenelement geeignet ist.

8.9.7 Die Aktion Wert steigern

Diese Aktion dient zum Erhöhen oder Reduzieren eines Datenelements: In der nachstehenden Abbildung werden die verfügbaren Optionen angezeigt:



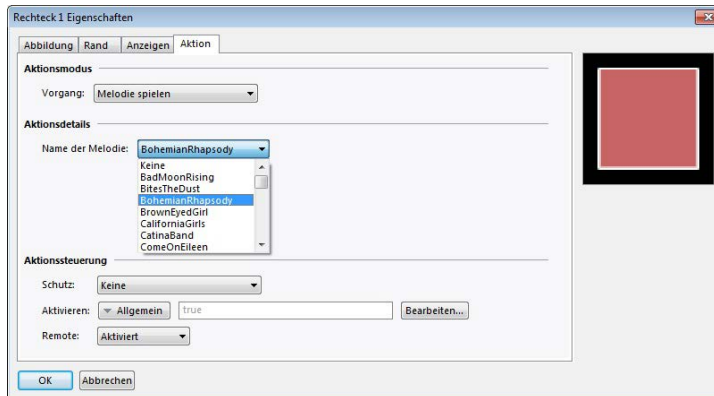
- Die Eigenschaft *Write To* (Schreiben in) definiert das zu ändernde Datenelement.
- Die Eigenschaft *Data* (Daten) gibt den Schritt an, um den das Element erhöht oder reduziert wird.
- Die Eigenschaft *Limit* (Grenze) definiert den minimalen oder maximalen Datenwert.
- Die Eigenschaft *Ramp Mode* (Steigerungsmodus) bestimmt, ob das Element erhöht oder reduziert werden soll.

Im oben gezeigten Beispiel wird das Tag `Data` beim Drücken und Festhalten des Primitivs wiederholt um 5 erhöht, bis 500 erreicht ist. Beachten Sie, dass diese Aktion entweder Gleitkomma- oder ganzzahlige Werte unterstützt.

Die Eigenschaften „Data“ (Daten) und „Limit“ (Grenze) müssen einem Typ entsprechen, der für das durch die Eigenschaft „Write To“ (Schreiben in) definierte Datenelement geeignet ist.

8.9.8 Die Aktion Melodie abspielen

Diese Aktion spielt über den internen Signalgeber des Zielgeräts eine ausgewählte Melodie ab.



- Mit *Tune Name* (Name der Melodie) wählen Sie die zu spielende Melodie aus.

Mit der Funktion `PlayRTTTL()` können eigene Melodien abgespielt werden.

8.9.9 Die Aktion Benutzer anmelden

Diese Aktion aktiviert den Anmeldebildschirm auf dem Zielgerät. Diese Aktion bietet keine Auswahlmöglichkeiten.

8.9.10 Die Aktion Benutzer abmelden

Diese Aktion meldet den aktuellen Benutzer des Zielgeräts ab. Diese Aktion bietet keine Auswahlmöglichkeiten.

8.9.11 Die Aktion Popup ausblenden

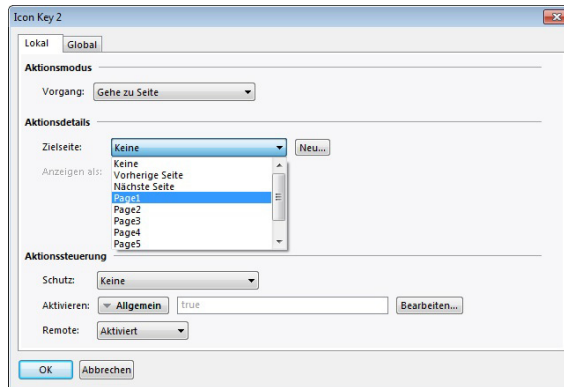
Diese Aktion blendet eine Seite aus, die als Popup angezeigt wurde, damit die darunter liegende Seite bzw. – bei einem verschachtelten Popup – das zuvor angezeigte Popup wieder angezeigt wird. Wenn kein Popup angezeigt wird, ist die Aktion wirkungslos.

8.9.12 Die Aktion Alle Popups ausblenden

Diese Aktion blendet alle Seiten aus, die als Popups oder verschachtelte Popups angezeigt wurden, damit die darunter liegende Seite wieder angezeigt wird. Wenn nur ein einzelnes Popup vorhanden ist, hat die Aktion die gleiche Wirkung wie die Aktion „Hide Popup“ (Popup ausblenden). Wenn kein Popup angezeigt wird, ist die Aktion wirkungslos.

8.10 Hinzufügen von Aktionen zu Symbolen

Bestimmte Geräte unterstützen das Hinzufügen von Aktionen zu den LED-Symbolen, die über oder unter dem Display des Geräts angezeigt werden. Verkleinern Sie die Ansicht, bis Sie die Symbole sehen. Doppelklicken Sie dann darauf, um die Eigenschaften anzuzeigen:



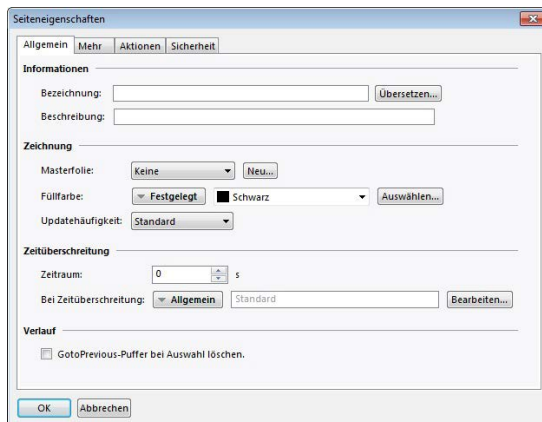
Wie Sie sehen, enthält dieses Dialogfeld zwei Registerkarten, die beide eine Aktion definieren. Die erste Registerkarte definiert die Aktion, die von diesem Symbol ausgeführt wird, wenn die aktuelle Seite angezeigt wird. Die zweite Registerkarte definiert eine Aktion, die auf jeder Seite ausgeführt werden soll. Diese Aktionen werden als lokale bzw. globale Aktionen bezeichnet. Nachdem Sie eine Aktion definiert haben, können Sie mit der rechten Maustaste auf das Symbol klicken und im daraufhin eingeblendeten Menü entweder „Make Global“ (Global einstellen) oder „Make Local“ (Lokal einstellen) wählen, um den Aktionstyp zu ändern. Diese Optionen sind nicht verfügbar, wenn beide Aktionen bereits definiert wurden.

Wie Sie die zugehörigen LEDs steuern, wird in einem späteren Abschnitt dieses Kapitels erörtert.

8.11 Bearbeiten von Seiteneigenschaften

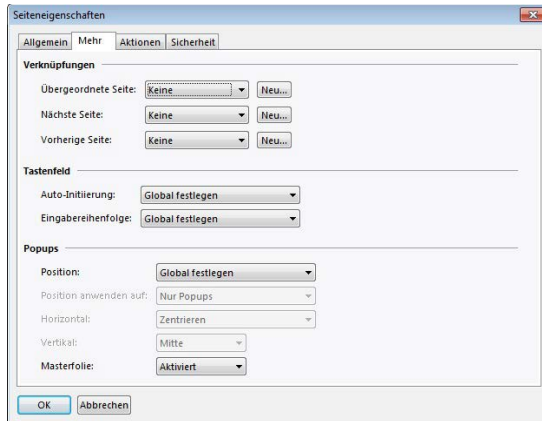
Wenn Sie im Bearbeitungsfenster mit der rechten Maustaste an eine Stelle klicken, an der sich keine Primitive befinden, wird das Kontextmenü aktiviert. In diesem Menü können Sie die Befehle für Eigenschaften auswählen, um die Eigenschaften einer Anzeigeseite zu bearbeiten:

8.11.1 Allgemeine Eigenschaften



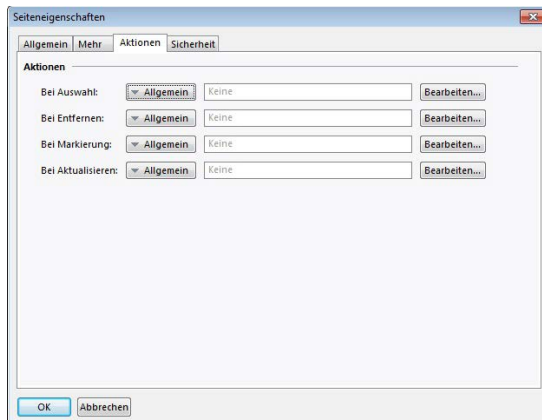
- Die Eigenschaften *Label* (Bezeichnung) und *Description* (Beschreibung) definieren allgemeine übersetzbare Zeichenfolgen, auf die mithilfe der Crimson-Eigenschaft „Extrahierungssyntax“ von anderer Stelle aus zugegriffen werden kann. Weitere Informationen finden Sie in dem Kapitel „Schreiben von Ausdrücken“.
- Die Eigenschaft *Master Slide* (Masterfolie) ermöglicht die Auswahl einer anderen Seite, die als Hintergrund für die aktuelle Seite verwendet wird. Auf diese Weise können gängige Benutzeroberflächenelemente wie Uhren, Alarmstatusanzeigen usw. auf einer einzigen Seite gezeichnet und dann in mehrere andere Seiten einbezogen werden.
- Die Eigenschaft *Fill Color* (Füllfarbe) definiert die Hintergrundfarbe der Seite, sofern keine Masterfolie verwendet wurde. Von der Animation der Hintergrundfarbe wird abgeraten. Die Hardware muss bei Änderungen alle Elemente auf der Seite neu zeichnen, wodurch die Leistung beeinträchtigt werden kann.
- Die Eigenschaft *Update Rate* (Aktualisierungsrate) bestimmt, wie häufig die Seite aktualisiert wird. Die Einstellung „Overdrive“ sollte unter normalen Umständen nicht verwendet werden. Die vorgegebene Einstellung entspricht gegenwärtig der Standardeinstellung.
- Die Eigenschaften unter *Timeout* definieren das Verhalten bei einer Zeitüberschreitung. Wenn der unter „Period“ (Zeitraum) angegebene Zeitraum ohne Benutzeraktivität verstreicht, wird die Aktion „On Timeout“ (Bei Timeout) ausgeführt. Weitere Informationen zu den möglichen Aktionen finden Sie im Kapitel „Schreiben von Aktionen“.
- Die Eigenschaft *Clear GotoPrevious Buffer on Selection* (GotoPrevious-Puffer bei Auswahl löschen) gibt an, dass der von `GotoPrevious()` und `GotoNext()` verwaltete Verlaufspuffer bei der Auswahl dieser Seite gelöscht werden soll. In der Regel legen Sie diese Eigenschaft auf der Hauptmenüseite Ihrer Datenbank fest und entfernen so die Möglichkeit, über diesen Punkt hinaus zurückzugehen.

8.11.2 Weitere Eigenschaften



- Die Eigenschaftengruppe „Links“ (Verknüpfungen) gibt die Seiten an, die von verschiedenen Standardaktionen auf einer Anzeigeseite ausgewählt werden sollen. Die Eigenschaft *Parent Page* (Übergeordnete Seite) definiert eine Seite, die ausgewählt werden soll, wenn eine Zeitüberschreitung stattfindet und keine Aktion definiert ist. Die Eigenschaft *Next Page* (Nächste Seite) definiert eine Seite, die ausgewählt werden soll, wenn die Eingabnavigation aktiviert ist und der Fokus über das letzte Feld auf der Seite hinaus verschoben wird. Die Eigenschaft *Previous Page* (Vorherige Seite) definiert eine Seite, die ausgewählt werden soll, wenn der Fokus auf ähnliche Weise über das erste Feld hinaus verschoben wird.
- Mit der Eigenschaft *Auto Initiate* (Auto-Initiierung) wird angegeben, ob das Tastenfeld für die Dateneingabe automatisch angezeigt werden soll, sobald ein alphanumerisches Zeichen von einer externen Quelle (z. B. Tastatur, USB-Scanner) empfangen wird und sofern aktivierte Eingabefelder vorhanden sind. Wenn dieser Modus deaktiviert ist, muss ein aktives Dateneingabefeld berührt werden, um das Tastenfeld anzuzeigen.
- Mit der Eigenschaft *Entry Order* (Eingabereihenfolge) wird festgelegt, wie mit den Schaltflächen „Weiter“ und „Vorherige“ des Tastenfelds zwischen den Feldern navigiert werden soll. Felder können in Zeilen und dann in Spalten oder aber in Spalten und dann in Zeilen ausgewählt werden.
- Mit der Eigenschaft *Position* kann die global definierte Position von Popup-Fenstern für diese Seite überschrieben werden. Wenn lokale Einstellungen aktiviert sind, wird die Position anhand der Einstellungen *Horizontal* (Horizontal) und *Vertical* (Vertikal) angegeben.
- Mit der Eigenschaft *Master Slide* (Masterfolie) wird angegeben, ob die Masterfolie während der Anzeige eines Popups aktiv bleiben soll. Bei der Standardeinstellung „Enabled“ (Aktiviert) funktionieren Schaltflächen auf der Masterfolie, auch wenn Schaltflächen auf der eigentlichen Seite deaktiviert sind, während ein Popup vorhanden ist. Dies kann nützlich sein, wenn Sie möchten, dass globale Navigationsoptionen auf der Masterfolie immer verfügbar sein sollen.

8.11.3 Aktionseigenschaften



- Die Eigenschaft *On Select* (Bei Auswahl) definiert eine Aktion, die ausgeführt werden soll, wenn die Seite angezeigt wird.
- Die Eigenschaft *On Remove* (Bei Entfernen) definiert eine Aktion, die ausgeführt werden soll, wenn die Seite deaktiviert wird.
- Die Eigenschaft *On Tick* (Bei Ticken) definiert eine Aktion, die einmal pro Sekunde ausgeführt werden soll.
- Die Eigenschaft *On Update* (Bei Aktualisieren) definiert eine Aktion, die bei jeder Anzeigeaktualisierung ausgeführt werden soll.

8.11.4 Sicherheitseigenschaften

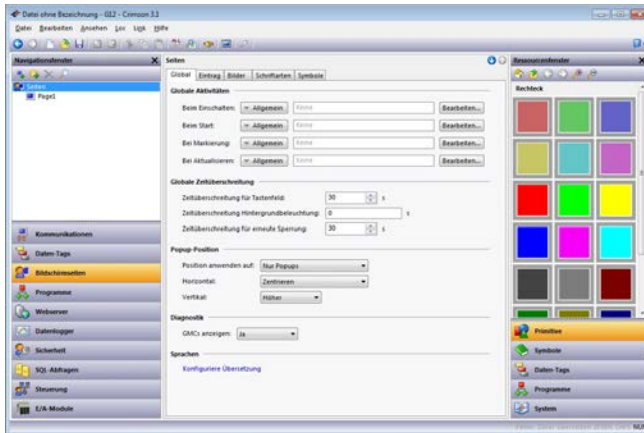
Informationen zu Sicherheitsbeschreibungen finden Sie im Kapitel „Verwenden von Sicherheit“.

8.12 Einstellungen der Benutzeroberfläche

Wenn Sie in der Navigationsliste das Stammelement auswählen, erhalten Sie Zugriff auf die Einstellungen der Benutzeroberfläche.

8.12.1 Globale Eigenschaften

Die Registerkarte „Global“ enthält verschiedene allgemeine Einstellungen, die für die gesamte Datenbank gelten:



8.12.1.1 Globale Aktivitäten

- Die Eigenschaft *On Power Up* (Bei Start) definiert eine Aktion, die ausgeführt werden soll, wenn das System gestartet wird.
- Die Eigenschaft *On Startup* (Bei Initialisierung) definiert eine Aktion, die kurz danach ausgeführt werden soll.¹
- Die Eigenschaft *On Tick* (Bei Ticken) definiert eine Aktion, die einmal pro Sekunde ausgeführt werden soll.
- Die Eigenschaft *On Update* (Bei Aktualisieren) definiert eine Aktion, die bei jeder Anzeigeaktualisierung ausgeführt werden soll.

Beachten Sie, dass diese Aktionen im Kontext der Benutzeroberflächentask ausgeführt werden. Dies bedeutet, dass Benutzerreaktionen beeinträchtigt werden, wenn die Aktionen zu lange dauern, und umgekehrt bestimmte Benutzeroberflächenaktivitäten, insbesondere die mit dem Sicherheitssystem verbundenen, zu einer Verzögerung der Aktionen führen können. Diese Aktionen sollten nicht für zeitkritische Kontrollaktivitäten verwendet werden. Informationen zu geeigneteren Alternativen finden Sie im Abschnitt zur IEC-61131-Steuerung.

8.12.1.2 Globale Timeouts

- Die Eigenschaft *Keypad Timeout* (Tastenfeld Timeout) definiert eine Zeitspanne ohne Benutzeraktivität, nach deren Ablauf alle Dateneingabevorgänge abgebrochen werden und das zugehörige Popup-Tastenfeld aus der Anzeige entfernt wird.
- Die Eigenschaft *Backlight Timeout* (Hintergrundbeleuchtung Timeout) definiert eine Zeitspanne ohne Benutzeraktivität, nach deren Ablauf die Anzeigebeleuchtung ausgeschaltet wird, um Energie zu sparen und die Anzeige zu schonen. Mit dem Standardwert Null wird diese Funktion deaktiviert.

1. Der Unterschied zwischen diesen beiden Eigenschaften ist gering und für die meisten Benutzer unwichtig.

- Die Eigenschaft *Relock Timeout* (Timeout erneut sperren) definiert die Zeitspann, nach deren Ablauf alle Aktionen, die durch die Methoden „Locked“ (Gesperrt) und „Hard Locked“ (Vollständig gesperrt) geschützt sind, automatisch wieder gesperrt werden. Die Aktionen können dann nur genutzt werden, wenn der Benutzer sie erneut entsperrt.

8.12.1.3 Popup-Position

- Die Eigenschaften *Horizontal* (Horizontal) und *Vertical* (Vertikal) definieren die Standardposition für Popup-Anzeigeseiten und Popup-Ziffernblöcke. Diese können bei Bedarf auf Seitenebene außer Kraft gesetzt werden, indem Sie in den Eigenschaften der betreffenden Seite neue Werte eingeben.

8.12.1.4 Diagnostik

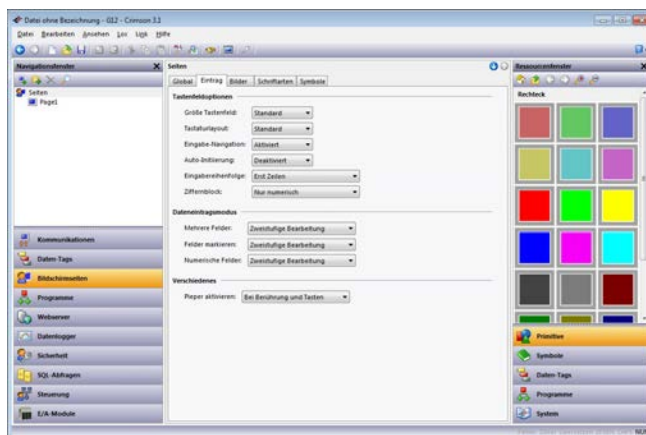
- Die Eigenschaft *Show GMCs* (GMCs anzeigen) dient zum Aktivieren oder Deaktivieren der Anzeige bestimmter Diagnoseinformationen nach einem Systemfehler zur Laufzeit. Die Informationen erleichtern zwar die Behebung von Softwareproblemen, können von Benutzern jedoch als störend empfunden werden.

8.12.1.5 Sprachen

- Mit der Schaltfläche *Configure Translation* (Übersetzung konfigurieren) können Sie die Sprachen konfigurieren, die im System verwendet werden sollen. Weitere Informationen finden Sie in dem Kapitel zum Thema „Lokalisierung“.

8.12.2 Eingabeeigenschaften

Die Registerkarte „Entry“ (Eingabe) enthält globale Einstellungen, die für die Dateneingabe gelten:



8.12.2.1 Ziffernblockoptionen

- Mit der Eigenschaft *Keypad Size* (Tastefeldgröße) können Sie die Größe des Tastenfelds für die Dateneingabe auswählen. Die Größe des Tastenfelds nimmt mit den verschiedenen Einstellungen stetig zu. Bei der Einstellung „Maximum“ nimmt das Tastenfeld fast den gesamten Bildschirm ein. Dies ist nützlich, wenn die Bediener beispielsweise unhandliche Handschuhe tragen.
- Mit der Eigenschaft *Keypad Layout* (Tastefeldlayout) wird zwischen dem Standardtastaturlayout und speziellen Layouts für verschiedene Sprachen gewählt. Diese Version von Crimson unterstützt spezielle Tastaturen für Französisch und Hebräisch.
- Mit der Eigenschaft *Input Navigation* (Eingabenavigation) werden die Tasten **WEITER** und **VORHERIGE** in den verschiedenen Popup-Tastefeldern ein- bzw. ausgeblendet. Mit diesen Tasten können Sie durch Eingabefelder navigieren, ohne zuerst das Tastenfeld zu deaktivieren.
- Mit der Eigenschaft *Auto Initiate* (Auto-Initiierung) wird angegeben, ob das Tastenfeld für die Dateneingabe automatisch angezeigt werden soll, sobald die Seite ausgewählt wurde und sofern aktivierte Eingabefelder vorhanden sind. Wenn dieser Modus deaktiviert ist, muss ein aktives Dateneingabefeld berührt werden, um das Tastenfeld anzuzeigen.
- Mit der Eigenschaft *Entry Order* (Eingabereihenfolge) wird festgelegt, wie mit den Schaltflächen „Weiter“ und „Vorherige“ des Tastenfelds zwischen den Feldern navigiert werden soll. Felder können in Zeilen und dann in Spalten oder aber in Spalten und dann in Zeilen ausgewählt werden.
- Mit der Option *Numeric Keypad* (Ziffernblock) wird das Erscheinungsbild des Tastenfelds gesteuert, das für numerische Einträge verwendet wird. Insbesondere wird festgelegt, ob Änderungen zusätzlich zur oder anstatt der herkömmlichen Eingabe aktiviert sind.

8.12.2.2 Dateneingabemodus

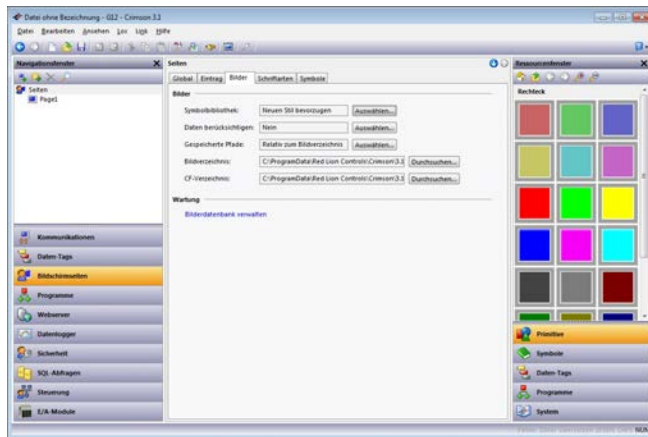
- Mit der Eigenschaft *Multi Data Entry* (Mehrfachdateneingabe) wird der Dateneingabemodus für Objekte im Format Mehrfachstatus gesteuert. Bei Auswahl von „Two-Stage Editing“ (Zweistufige Bearbeitung) muss die **EINGABETASTE** gedrückt werden, um die Änderungen zu bestätigen. Bei Auswahl der einstufigen Bearbeitung werden die neuen Daten in das zugehörige Datenelement geschrieben, sobald mit **ERHÖHEN** oder **SENKEN** eine Änderung vorgenommen wird. Die einstufige Eingabe ist schneller, kann jedoch bei der Änderung einer Mehrfachstatus-Einstellung dazu führen, dass Zwischenwerte geschrieben werden.
- Mit der Eigenschaft *Flag Data Entry* (Dateneingabe markieren) wird der Dateneingabemodus für Objekte im Format Zweifachstatus gesteuert. Dies funktioniert genauso wie die oben genannte Eigenschaft.

8.12.2.3 Sonstige

- Mit der Eigenschaft *Activate Beeper* (Pieper aktivieren) wird der Signalgeber des Zielgeräts nach Bedarf ein- oder ausgeschaltet. Der Pieper gibt Feedback zur Tastatur- und Touchscreen-Aktivierung, kann jedoch während des Entwicklungsprozesses als störend empfunden werden.

8.12.3 Eigenschaften von Bildern

Die Registerkarte „Images“ (Bilder) dient zum Verwalten von Bildern in der Datenbank:



8.12.3.1 Bilder

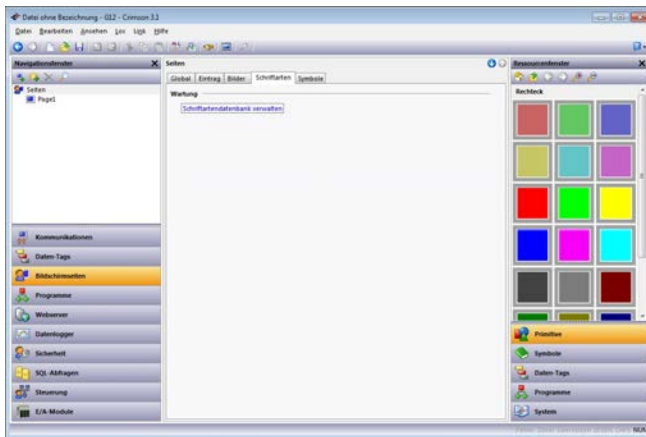
- Die Eigenschaft *Symbol Library* (Symbolbibliothek) gibt an, ob Crimson geglättete Symbole im neuen Stil oder die von Crimson 2.0 und Crimson 3.0 verwendeten raueren Symbole im alten Stil bevorzugen soll. Bei Datenbanken, die aus Crimson 3.0 importiert wurden, werden Symbole im alten Stil bevorzugt, bis diese Einstellung geändert wird. Dadurch wird sichergestellt, dass die visuelle Konsistenz gewahrt bleibt, bis Sie prüfen können, welche Auswirkungen der Wechsel zu den neuen Symbolen hat. Für neue, in Crimson 3.1 erstellte Datenbanken werden standardmäßig die glatteren, hoch auflösenden Symbole verwendet.
- Die Eigenschaft *Include Data* (Daten berücksichtigen) gibt an, ob externe Bilder, die in eine Anzeigeseite gezogen wurden, als Zeiger auf den Quellspeicherort gespeichert werden oder ob die eigentlichen Bilddaten in die Datenbankdatei aufgenommen werden sollen. Durch die Berücksichtigung von Bilddaten wird die Datenbank in der Regel so umfangreich, dass es unmöglich wird, die Funktion „Support Upload“ (Upload unterstützen) zu verwenden, ohne den Speicher des Zielgeräts zu füllen.
- Die Eigenschaft *Stored Paths* (Gespeicherte Pfade) definiert, wie Bildverknüpfungen gespeichert werden. Im Modus „Absolute“ (Absolut) wird der vollständige Pfad gespeichert, einschließlich des Laufwerksbuchstabens. Die beiden relativen Modi speichern und interpretieren Bildpfade entweder relativ zur Datenbank oder relativ zum Crimson-Bildverzeichnis. Datenbank- und Bilddateien können somit zwischen Computern verschoben werden, ohne dass Sie sich um die absoluten Speicherpfade kümmern müssen.
- Die Eigenschaft *Image Directory* (Bildverzeichnis) definiert den oben referenzierten Bildpfad.
- Mit der Eigenschaft *CF-Directory* (CF-Verzeichnis) können Sie festlegen, wo Crimson Kopien der Bilder findet, die auf der Speicherkarte eines Zielgeräts gespeichert werden. Das Bild-Primitiv der Speicherkarte verwendet diesen Pfad bei der Bildvorschau.

8.12.3.2 Wartung

- Mit der Schaltfläche *Manage Image Database* (Bilderdatenbank verwalten) können Sie den Image Manager (Bildmanager) aufrufen, um die in der Datenbank verwendeten Bilder anzuzeigen und zu bearbeiten. Weitere Informationen hierzu finden Sie im folgenden Abschnitt.

8.12.4 Schrifteigenschaften

Die Registerkarte „Fonts“ (Schriftarten) dient zum Verwalten von Schriftarten in der Datenbank:

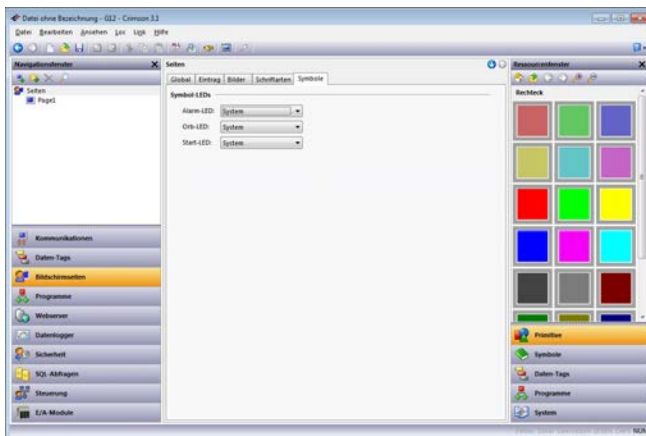


8.12.4.1 Wartung

- Mit der Verknüpfung *Manage Font Database* (Schriftartendatenbank verwalten) können Sie den Font Manager (Schriftartmanager) aufrufen, um die in der Datenbank verwendeten Schriftarten anzuzeigen und zu bearbeiten. Weitere Informationen hierzu finden Sie im folgenden Abschnitt.

8.12.5 Symboleigenschaften

Auf der Registerkarte „Icons“ (Symbole) werden die Eigenschaften der Symbol-LEDs auf den HMIs festgelegt, die sie bereitstellen:



- Mit der Eigenschaft *Alarm LED* (Alarm-LED) legen Sie fest, ob die linke oder Alarm-LED vom System gesteuert wird oder vom Benutzer gesteuert werden kann. Standardmäßig weist eine rot blinkende LED auf einen aktiven und nicht angenommenen Alarm hin und eine rot leuchtende LED auf einen angenommenen Alarm.
- Mit der Eigenschaft *Orb LED* (Orb-LED) legen Sie fest, ob die mittlere oder Orb-LED vom System gesteuert wird oder vom Benutzer gesteuert werden kann. Standardmäßig weist eine blau blinkende LED darauf hin, dass Bootcode in ein Gerät geladen wird. Beim Zugriff auf eine Speicherkarte flackert die LED.

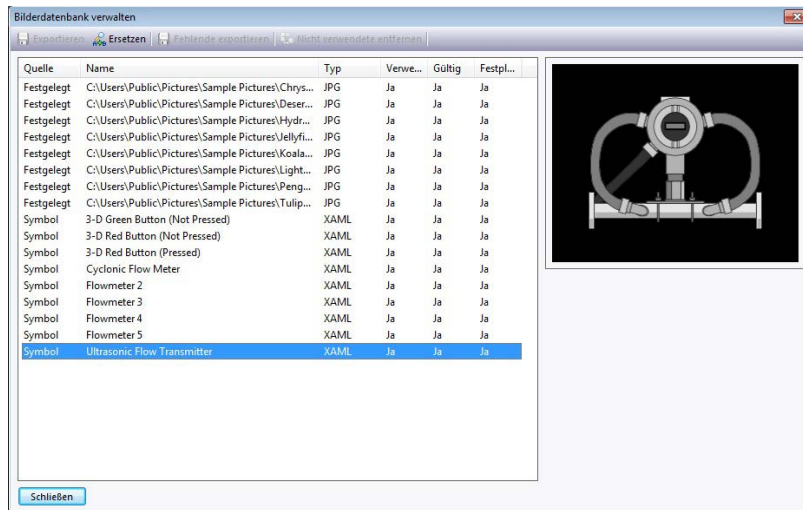
- Mit der Eigenschaft *Home LED* (Start -LED) legen Sie fest, ob die rechte oder Home-LED vom System gesteuert wird oder vom Benutzer gesteuert werden kann. Standardmäßig weist eine grün leuchtende LED darauf hin, dass das Gerät eingeschaltet ist.

Im Abschnitt „Hinzufügen von Aktionen zu Symbolen“ weiter oben in diesem Kapitel finden Sie Informationen zum Hinzufügen von Aktionen zu LEDs auf Geräten, die berührungsempfindliche Signalgeber unterstützen.

8.13 Verwalten von Bildern

Der Image Manager (Bildmanager) wird über die Registerkarte „Images“ (Bilder) in den Einstellungen für die Benutzeroberfläche aufgerufen. Er enthält eine Liste der in der Datenbank referenzierten Bilder und deren zugehörigen Eigenschaften. Damit können Sie die Bilder anzeigen und bestimmte Änderungen an der Speicherung und Verwendung der Bilder vornehmen.

Das Beispiel unten zeigt den Image Manager (Bildmanager) einer komplexen Datenbank:



In der Hauptlistenansicht werden die Eigenschaften der verschiedenen Bilder angezeigt:

- Die Spalte *Source* (Quelle) gibt an, ob das Bild über einen festen oder relativen Pfad aus einer Datei abgerufen wird, aus der Symbolbibliothek oder aus internen Daten, die gespeichert werden, wenn ein Bild aus einer anderen Quelle eingefügt oder gezogen wurde.
- In der Spalte *Name* wird der Dateiname für Bilder angezeigt, die in Dateien gespeichert sind, sowie die relevanten Symbolinformationen für Bilder, die aus der Symbolbibliothek stammen.
- Die Spalte *Type* (Typ) gibt den Dateityp der Bilddaten an.
- Die Spalte *Used* (Verwendet) gibt an, ob das Bild in der Datenbank verwendet wird.
- Die Spalte *Valid* (Gültig) gibt an, ob gültige Bilddaten verfügbar sind. Diese Spalte kann auf „Nein“ gesetzt werden, wenn ein Bild aus einer Datenträgerdatei bezogen wurde, die nicht mehr verfügbar ist, und wenn die Datenbank nicht mit der oben beschriebenen Eigenschaft „Include Data“ (Daten berücksichtigen) darauf konfiguriert ist, eigene Bilddaten zu speichern.
- Die Spalte *Disk* (Festplatte) gibt an, ob das Bild auf der Festplatte vorhanden ist. Bilder, die direkt in den Editor eingefügt oder gezogen wurden, waren möglicherweise nie auf der Festplatte vorhanden, und Bilder, die aus Dateien stammen, aber auch in der Datenbank gespeichert wurden, können jetzt fehlen, wenn die Datei nicht mehr vorhanden ist.

Mithilfe der Symbolleiste am oberen Fensterrand können verschiedene Befehle ausgeführt werden:

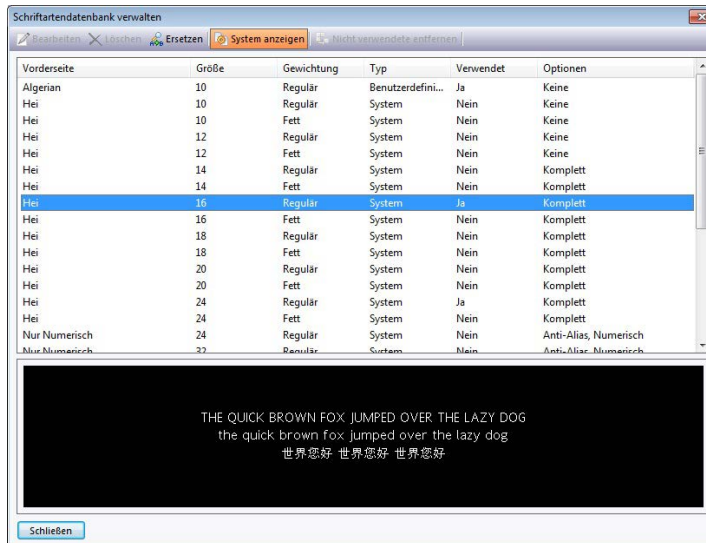
- Der Befehl *Export* (Exportieren) speichert ein Bild, das zwar verfügbar, aber nicht in einer Datei auf der Festplatte gespeichert ist. Wenn für das ausgewählte Bild bereits ein Dateiname definiert wurde, wird dieser Name verwendet. In anderen Fällen werden Sie aufgefordert, einen Dateinamen auszuwählen.
- Mit dem Befehl *Replace* (Ersetzen) können Sie ein bestimmtes Bild durch ein anderes ersetzen. Alle Verweise auf das Bild in der Datenbank werden aktualisiert, um die Änderung widerzuspiegeln.

- Mit dem Befehl *Export All* (Alle exportieren) werden alle Bilder gespeichert, die zwar verfügbar, aber nicht auf der Festplatte gespeichert sind, und für die Dateinamen festgelegt wurden. Mit diesem Befehl können Sie sicherstellen, dass alle Bilder in externen Dateien gespeichert werden, bevor Sie „Include Data“ (Daten berücksichtigen) deaktivieren.
- Mit dem Befehl *Purge Unused* (Unbenutzte entfernen) werden alle Bilder entfernt, die nicht in der Datenbank verwendet werden. Dadurch wird beim Speichern der Datenbank auf der Festplatte Speicherplatz gespart. Die Verwendung dieses Befehls kann auch die Speicherauslastung im Zielgerät verringern.

8.14 Verwalten von Schriftarten

Der Font Manager (Schriftartmanager) wird über die Registerkarte „Fonts“ (Schriftarten) in den Einstellungen für die Benutzeroberfläche aufgerufen. Er enthält eine Liste der in der Datenbank referenzierten Schriftarten und deren zugehörigen Eigenschaften. Damit können Sie die Schriftarten anzeigen und bestimmte Änderungen an der Speicherung und Verwendung der Schriftarten vornehmen.

Im unten stehenden Beispiel ist der Font Manager (Schriftartmanager) einer komplexen Datenbank abgebildet:



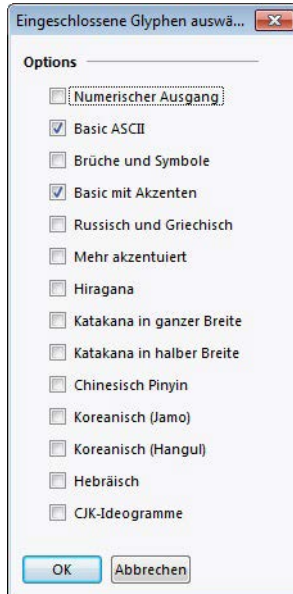
In der Hauptlistenansicht werden die Eigenschaften der verschiedenen Schriftarten angezeigt:

- Die Eigenschaft *Face* (Schrift) zeigt den Namen der Schriftart an.
- Die Eigenschaft *Size* (Größe) zeigt die Höhe der Schriftart in Pixel an.
- Die Eigenschaft *Weight* (Gewicht) gibt an, ob die Schriftart fett angezeigt wird oder nicht.
- Die Eigenschaft *Type* (Typ) gibt an, ob es sich um eine System- oder benutzerdefinierte Schriftart handelt.
- Die Eigenschaft *Used* (Verwendet) gibt an, ob die Schriftart in der Datenbank verwendet wird.
- Die Eigenschaft *Options* (Optionen) listet die für die Schriftart ausgewählten Optionen auf.

Mithilfe der Symbolleiste am oberen Fensterrand können verschiedene Befehle ausgeführt werden:

- Mit der Schaltfläche *Edit* (Bearbeiten) können Sie die Eigenschaften von benutzerdefinierten Schriftarten bearbeiten.
- Mit der Schaltfläche *Delete* (Löschen) kann eine nicht verwendete Schriftart gelöscht werden. Nachdem eine Schriftart gelöscht wurde, wird sie nicht mehr in der Dropdown-Liste für die Schriftartauswahl angezeigt, kann aber mit der zugehörigen Auswahl Schaltfläche neu erstellt werden.
- Mit der Schaltfläche *Replace* (Ersetzen) kann eine Schriftart durch eine andere ersetzt werden. Alle Verweise auf die Schriftart in der Datenbank werden aktualisiert, um die Änderung widerzuspiegeln.
- Mit der Schaltfläche *Show System* (System anzeigen) wird gesteuert, ob Systemschriftarten in der Liste angezeigt werden.
- Mit der Schaltfläche *Purge Unused* (Unbenutzte entfernen) werden alle nicht benutzten Schriftarten aus der Datenbank entfernt. Dadurch wird beim Speichern der Datenbank auf der Festplatte Speicherplatz gespart. Eine entfernte Schriftart wird genau wie eine gelöschte Schriftart nicht mehr in der Dropdown-Liste für die Schriftartauswahl angezeigt, kann aber mit der zugehörigen Auswahl Schaltfläche neu erstellt werden.

Wenn Sie die Eigenschaften einer benutzerdefinierten Schriftart bearbeiten, wird folgendes Dialogfeld angezeigt:



Mit den verschiedenen Optionen können bestimmte Zeichensätze in das Schriftartbild aufgenommen werden, das erstellt und auf das Zielgerät heruntergeladen wird. Durch die Beschränkung der Zeichen auf die für Ihre Anwendung erforderlichen Zeichen sparen Sie insbesondere bei größeren Schriftarten Speicherplatz. Beachten Sie, dass die Option „Numeric Output“ (Numerische Ausgabe) allein verwendet werden kann, um die Schriftart auf Ziffern, Dezimalstellen und andere Zeichen zu beschränken, die für die Wiedergabe von konventionellen, wissenschaftlichen oder Hexadezimalzahlen verwendet werden.

8.15 Weiterführende Themen

In diesem Abschnitt werden fortgeschrittene Techniken beschrieben, die von der Mehrheit der Benutzer normalerweise nicht benötigt werden.

8.15.1 Definieren von Farbausdrücken

Wie bereits erwähnt, können Farbeigenschaften über Ganzzahlausdrücke definiert werden bzw. über lokale Programme, die ganzzahlige Werte zurückgeben. Diese Mechanismen sind für Situationen vorgesehen, in denen normale Farbanimationsmethoden nicht ausreichen.

Crimson arbeitet mit 15-Bit-Farbkonstanten, wobei die fünf niedrigsten Bits Rot, die nächsten fünf Bits Grün und die oberen fünf Bits Blau darstellen. (Beachten Sie, dass dies auch für Geräte mit umfangreicheren Farbsätzen gilt. Die zusätzlichen Farben werden für abgestufte Füllungen, Symbole und Bilder verwendet, können jedoch nicht für Farbeinstellungen und Konstanten ausgewählt werden.) Sie können Farbwerte genau wie andere ganzzahlige Werte bearbeiten.

8.15.1.1 Erstellen von Farben

Mit der Funktion `ColGetRGB(r,g,b)` können Sie einen Farbwert aus dessen roten, grünen und blauen Komponenten erstellen. Obwohl Crimson 15-Bit-Farbwerte verwendet, die drei 5-Bit-Werte enthalten, werden die an diese Funktion übergebenen Argumente intern um den Faktor 8 reduziert und sollten daher im Bereich 0 bis 255 liegen. `ColGetRGB(128, 0, 64)` gibt daher eine violette Farbe zurück, die den Rot-Wert 16, keine Grün-Komponente und den Blau-Wert 8 hat.

8.15.1.2 Aufteilen von Farben

Die Funktionen `ColGetRed(rgb)`, `ColGetGreen(rgb)` und `ColGetBlue(rgb)` können verwendet werden, um auf die einzelnen Farbkomponenten eines Farbwerts zuzugreifen. Im Einklang mit der von `ColGetRGB()` verwendeten Konvention werden die von diesen Funktionen zurückgegebenen Werte auf 0 bis 255 skaliert.

8.15.1.3 Auswählen von Farben

Die Funktion `ColPick2()` kann verwendet werden, um ausgehend vom Wert eines Ausdrucks zwischen zwei Farben zu wählen. Der Ausdruck `ColPick2(Flag1, Col1, Col2)` gibt beispielsweise `Col1` zurück, wenn `Flag1` nicht Null ist, bzw. `Col2`, wenn `Flag1` Null ist. Das erste und zweite Farbagument können bei Bedarf durch Aufrufe an die Funktion `ColGetRGB()` ersetzt werden.

8.15.1.4 Mischen von Farben

Die Funktion `ColBlend()` kann verwendet werden, um eine Farbe zu erzeugen, die eine benutzerdefinierte Mischung aus zwei anderen Farben darstellt. Der Ausdruck `ColBlend(Data, 0, 100, Col1, Col2)` gibt beispielsweise `Col1` zurück, wenn `Data` 0 ist, und `Col2`, wenn „Data“ gleich 100 ist. Bei den Zwischenwerten handelt es sich um entsprechende Mischungen der beiden Farben, wodurch ein weicher Übergang von einer Farbe zur anderen möglich ist. Die Farbagumente können durch Aufrufe an die Funktion `ColGetRGB()` ersetzt werden.

8.15.1.5 Reagieren auf Berührungen

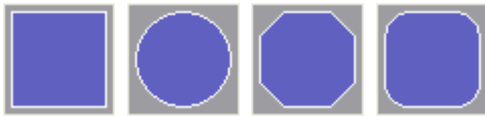
Die Systemvariable `IsPressed` ist „true“, wenn das aktuelle Primitiv berührt wurde, andernfalls „false“. `SiZeichnungs-Nr. LP1044-Ge` kann zusammen mit den Farbauswahlfunktionen verwendet werden, um ein Primitiv je nach Berührungsstatus zu animieren. Beachten Sie, dass Primitive nur für Berührungen aktiviert werden, wenn eine Aktion dafür definiert ist oder wenn sie eine inhärente Aktion unterstützen.

Kapitel 9 Primitiv-Arten

In diesem Kapitel werden alle von Crimson bereitgestellten Primitive einzeln beschrieben.

9.1 Kernprimitive

9.1.1 Geometrische Primitive



Die geometrischen Primitive stellen einfache Formen dar. Dazu gehören Rechtecke, vollständige und partielle Ellipsen, Rechtecke mit abgerundeten oder beschnittenen Ecken, Polygone, Sterne, Pfeile, Parallelogramme und Trapeze. Alle diese Primitive unterstützen Behälterfüllungen und können daher zur Implementierung von Effekten wie Balkendiagrammen verwendet werden. Sie unterstützen außerdem das Hinzufügen von Text oder Daten und können daher zum Erstellen von Text- oder Datenanzeigen bzw. zur Bereitstellung von Dateneingaben verwendet werden. Ferner unterstützen sie das Hinzufügen von Aktionen und können daher zur Implementierung interaktiver Anzeigeelemente verwendet werden.

9.1.1.1 Grundeigenschaften

Alle geometrischen Primitive verfügen über Eigenschaften, die ihr grundlegendes Erscheinungsbild in Bezug auf Füllungen, Ränder und Kantenbeschneidung steuern. Mit diesen Eigenschaften kann eine Vielzahl von Effekten erstellt werden. Experimentieren Sie mit den verschiedenen Einstellungen, um sich mit den Möglichkeiten vertraut zu machen. Diese Standardeinstellungen finden Sie im vorherigen Kapitel. Die geometrischen Primitive sind zwar sehr einfach, doch die Unterstützung von Behälterfüllungen, Daten, Text und Aktionen bedeutet, dass ein großer Teil der meisten Datenbanken allein mit Rechtecken oder abgerundeten Rechtecken erstellt werden kann.

9.1.1.2 Weitere Eigenschaften

Einige geometrische Primitive bieten zusätzliche Konfigurationsoptionen:

- Rechtecke mit abgerundeten oder beschnittenen Ecken verfügen über die Eigenschaft *Skip Corners* (Ecken überspringen), mit der der Eckeneffekt bei bestimmten Ecken ausgelassen werden kann. Dies ersetzt die verschiedenen halb beschnittenen Formen in Crimson 3.0 und sorgt gleichzeitig für mehr Flexibilität.
- Polygone verfügen über die Eigenschaft *Rotation* (Drehung), mit der die Ausrichtung des Polygons innerhalb des Begrenzungsrechtecks angepasst werden kann. Beachten Sie, dass das Polygon immer so skaliert ist, dass das Rechteck gefüllt wird, und dass die Drehung entsprechend angewendet wird.
- Auch Sterne verfügen über die Eigenschaft *Rotation* (Drehung). Zusätzlich verfügen über sie die Eigenschaft *Inner Radius* (Innenradius), mit der die Spitzigkeit der Sternspitzen angepasst werden kann. Dieses Verhalten lässt sich schwer erklären, wird beim Experimentieren mit der Einstellung jedoch schnell klar.
- Pfeile, Dreiecke, Parallelogramme und Trapeze verfügen über die Eigenschaften *Direction* (Richtung) und *Reflection* (Spiegelung), mit denen die Formen entweder anhand dieser Eigenschaften oder über das Menü „Transform“ (Transformieren) im Grafikeditor gedreht und gespiegelt werden können.

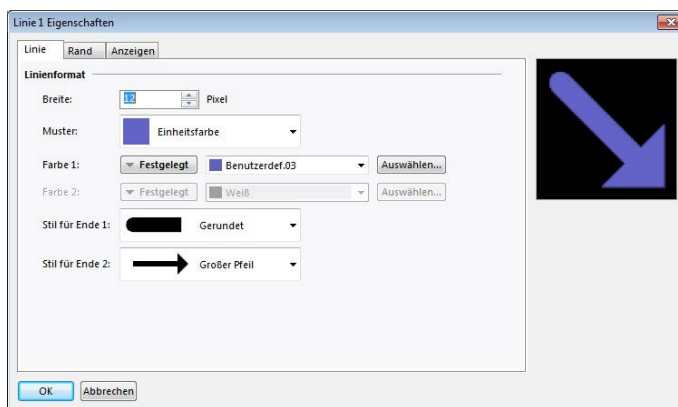
- Partielle Ellipsen verfügen ebenfalls über Transformationseigenschaften. Zusätzlich verfügen sie über die Eigenschaft *Show Edges* (Kanten anzeigen), die definiert, ob die Kante nur um den Bogen herum oder um den Bogen und die geraden Kanten herum gezeichnet wird.

9.1.2 Das Linien-Primitiv



Das Linien-Primitiv wird verwendet, um eine Linie auf der Seite zu platzieren. Das Linien-Primitiv von Crimson 3.1 unterstützt eine Vielzahl von Einstellungen, darunter strukturierte und abgestufte Füllungen, Linienränder und Kantenbeschneidung sowie wählbare Endformate zum Erstellen von Pfeilen und anderen Formen.

9.1.2.1 Linieneigenschaften



- Die Eigenschaft *Width* (Breite) gibt die Breite der Linie von 1 bis 80 Pixel an.
- Die Eigenschaft *Pattern* (Muster) gibt das Muster an, mit dem die Linie gefüllt werden soll. Die verfügbaren Optionen sind identisch mit denen für Füllungen und Ränder. Informationen zu den verschiedenen Einstellungen finden Sie im vorherigen Kapitel.
- Die Eigenschaften *Color 1* (Farbe 1) und *Color 2* (Farbe 2) dienen zur Auswahl der Farben für das ausgewählte Muster.
- Die Eigenschaften *End 1 Style* (Stil für Ende 1) und *End 2 Style* (Stil für Ende 2) definieren die Effekte, die auf die Linienenden angewendet werden sollen. Eine flache Endkappe endet genau am Ende der Linie. Eine quadratische Endkappe ragt um einen Betrag über das Ende der Linie hinaus, der der Breite der Linie entspricht. Die abgerundeten und spitzen Endkappen dürften genau wie die verschiedenen Pfeilspitzen und anderen Formen selbsterklärend sein.

9.1.2.2 Kanteneigenschaften

Informationen zu Rändern und zur Kantenbeschneidung finden Sie im vorherigen Kapitel.

9.1.3 Text- und Daten-Primitive



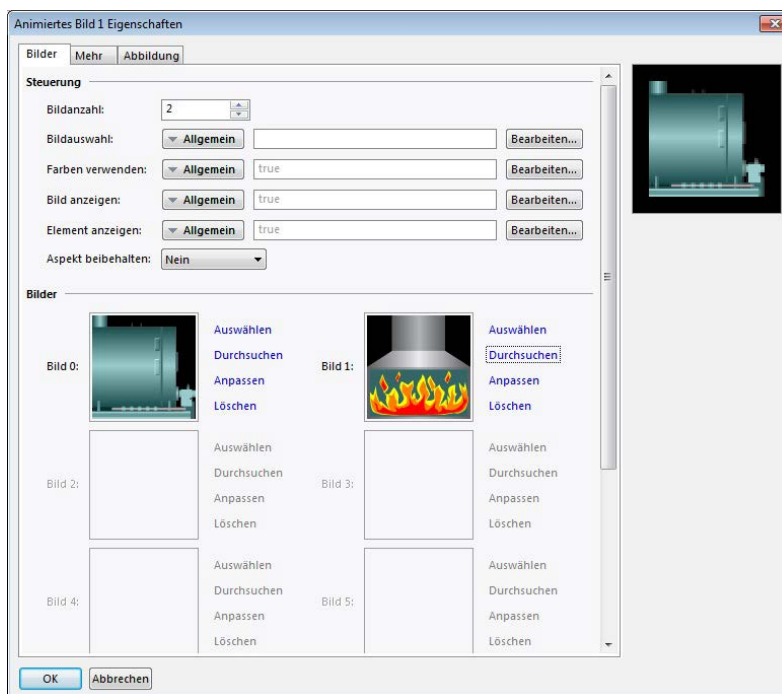
Die Textfeld- und Datenfeld-Primitive sind Rechtecke mit vordefinierten Daten- und Textelementen, für die keine Füll- oder Randfarben definiert sind. Sie sind darauf ausgelegt, das Hinzufügen von Daten- und Textelementen einfacher zu gestalten. Außerdem sollen sie jenen Benutzern die Arbeit erleichtern, die nicht damit vertraut sind, aus einfachen geometrischen Primitiven komplette Datenbanken aufzubauen. Sie können auch verwendet werden, um einem Primitiv ein zweites Daten- oder Textelement hinzuzufügen, und helfen beim Erstellen von Gruppen. Details zu den Standardeinstellungen finden Sie im vorherigen Kapitel.

9.1.4 Das Bild-Primitiv



Das Bild-Primitiv dient zum Anzeigen eines Bildes, das möglicherweise anhand eines numerischen Werts unter mehreren Bildern ausgewählt wurde. Das Primitiv unterstützt die Anzeige von Bitmaps, JPEGs, Metadateien, Bitmaps und vielen anderen Bildtypen. Es kann einen transparenten oder gefüllten Hintergrund verwenden und optional einen Rand definieren, der das Bild umgibt. Es unterstützt außerdem das Hinzufügen von Daten, Text oder Aktionen und ermöglicht so das Erstellen komplexerer Elemente.

Unten ist die Registerkarte „Image“ (Bild) für ein animiertes Bild-Primitiv abgebildet:



- Die Eigenschaft *Image Count* (Bildanzahl) definiert die Anzahl der Bildplätze dieses Primitivs. Eines der Bilder wird dann auf Basis des Werts der Eigenschaft „Image Select“ (Bildauswahl) für die Anzeige zu einer beliebigen Zeit ausgewählt.
- Die Eigenschaft *Image Select* (Bildauswahl) wählt das gewünschte Bild aus. Es wird als Wert mit der Basis Null betrachtet und entsprechend der Bildanzahl reduziert. Wenn also vier Bilder definiert sind, wird bei den Werten 0, 4, 8 usw. das erste Bild angezeigt und bei den Werten 1, 5, 9 usw. das zweite Bild und so fort.
- Die Eigenschaft *Use Colors* (Farben verwenden) wird dazu verwendet, ein Bild auf Schwarzweiß zu reduzieren oder seine Farben zu erhalten. Ein Ausdruck, der als nicht Null ausgewertet wird, oder ein leerer Ausdruck ergeben ein Farbbild. Bei einem Nullwert wird das Bild mithilfe der normalen

R-G-B-Helligkeitsgewichtungen auf Graustufen reduziert. Diese Option ist nützlich, wenn auf einer Schaltfläche der deaktivierte Status eines Bildes angezeigt wird.

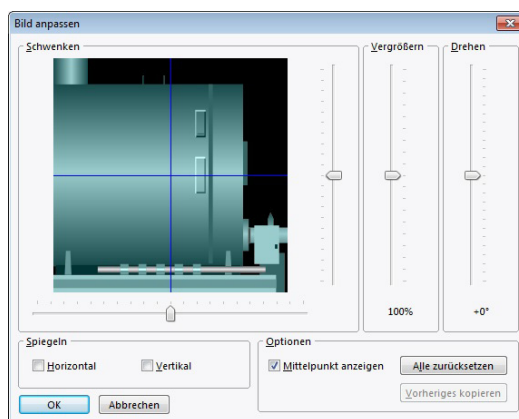
- Mit der Eigenschaft *Show Image* (Bild zeigen) wird das Bild ein- oder ausgeblendet. Wenn für das Primitiv weder ein Rand noch eine Füllung definiert ist, entspricht es aus funktionaler Sicht der Eigenschaft *Show Item* (Element anzeigen), zeigt aber ansonsten weiterhin den Rand oder Hintergrund gemäß Konfiguration an.
- Mit der Eigenschaft *Show Item* (Element anzeigen) wird das gesamte Primitiv ein- bzw. ausgeblendet.

9.1.4.1 Definieren von Bildern

Im Bereich *Images* (Bilder) des Dialogfelds werden die Bilder für jeden Bildplatz definiert. Über die Schaltfläche „Pick“ (Auswählen) neben den einzelnen Bildern wird ein Dialogfeld angezeigt, das Sie daran erinnert, dass Sie ein Bild einfach auf das Feld ziehen können. Dieses Bild kann aus der Kategorie „Symbolbibliothek“ des Ressourcenfensters, aus einem Ordner in Windows Explorer oder aus einer anderen Drag-and-Drop-fähigen Anwendung gezogen werden. Mit der Schaltfläche „Browse“ (Durchsuchen) können Sie eine Datei öffnen, die ein geeignetes Bildformat enthält, und in diesen Bildplatz laden. Wie bereits erwähnt, werden JPEGs, Metadateien, Bitmaps und viele andere Grafikdateiformate unterstützt.

9.1.4.2 Anpassen von Bildern

Über die Schaltfläche „Adjust“ (Anpassen) neben einem Bild können Sie das Bild ändern:

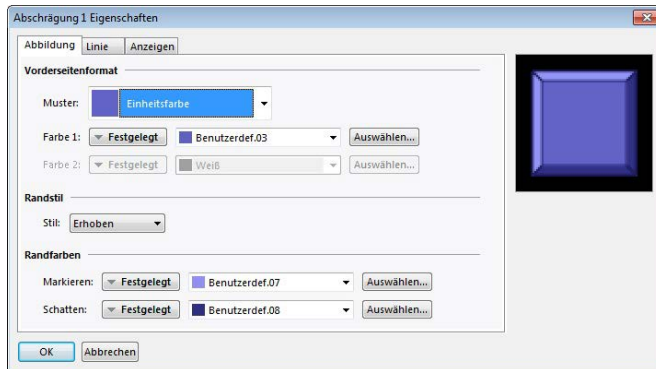


Die verschiedenen Schieberegler können zum Schwenken, Zoomen und Drehen des Bildes verwendet werden. Mit den Kontrollkästchen kann das Bild horizontal oder vertikal gespiegelt werden. Das Kontrollkästchen „Show Center“ (Mittelpunkt anzeigen) blendet die blauen Linien ein und aus, die die Bildmitte markieren. Mit der Schaltfläche „Reset“ (Zurücksetzen) kann das Bild in seinen ursprünglichen Zustand zurückversetzt werden. Gelegentlich wird ein Bild mithilfe der Bearbeitungsoptionen verändert, um verschiedene Zustände zu erstellen, die bei der Animation eingesetzt werden können.

9.1.5 Das abgeschrägte Primitiv

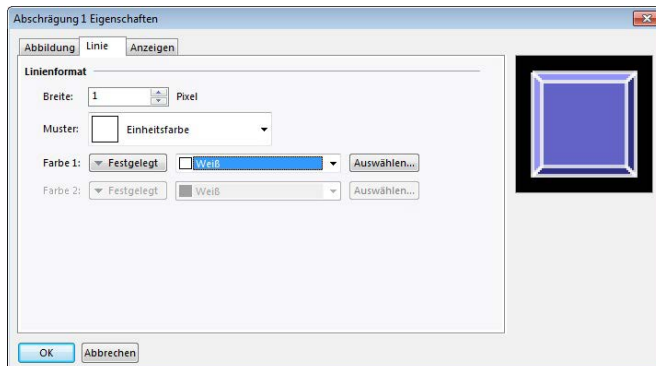
Das abgeschrägte Primitiv zeigt ein Rechteck mit einem erhöhten oder abgesenkten Rahmen an. Die Flächenfarbe sowie die Farben für die hervorgehobenen und schattierten Anteile können geändert werden. Sie können auch die Farbe und Breite der Linien im Primitiv ändern oder sie vollständig entfernen. Außerdem kann ein Layoutgriff verwendet werden, um die Breite der abgeschrägten Teile anzupassen.

9.1.5.1 Figureigenschaften



- Die Eigenschaften *Pattern* (Muster), *Color 1* (Farbe 1) und *Color 2* (Farbe 2) definieren die Farbe der Fläche des Primitivs auf die übliche Weise. Weitere Details finden Sie im vorherigen Kapitel.
- Die Eigenschaft *Style* (Stil) steuert, ob die Abschrägung angehoben oder abgesenkt wird. Dieser Effekt wird einfach durch Vertauschen der Licht- und Schattenfarben erreicht.
- Mit den Eigenschaften *Highlight* (Licht) und *Shadow* (Schatten) werden die zwei Farben definiert, mit denen der 3D-Effekt für die Außenseite des Primitivs erstellt wird.

9.1.5.2 Linieneigenschaften



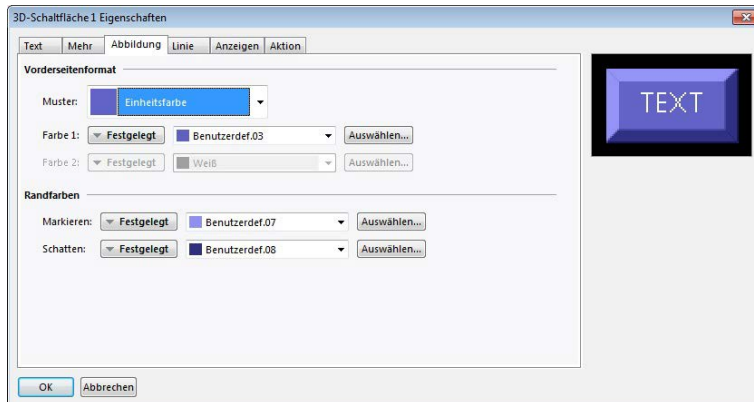
- Die Linieneigenschaften sind konventionell. Es ist lediglich zu beachten, dass bei der Einstellung der *Width* (Breite) auf Null alle Trennlinien entfernt werden. Details hierzu finden Sie im vorherigen Kapitel.

9.1.6 Die Schaltflächen-Primitive



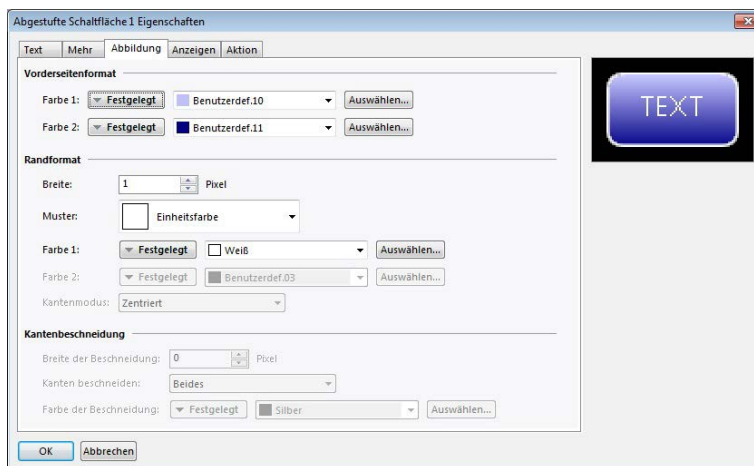
Die Schaltflächen-Primitive implementieren abgeschrägte oder abgestufte Schaltflächen. Der Text wurde vorkonfiguriert, damit die Schaltfläche beschriftet werden kann. Er kann jedoch entfernt werden, damit Live-Daten hinzugefügt werden können. Eine Registerkarte „Action“ (Aktion) wird ebenfalls standardmäßig bereitgestellt. Sie wird jedoch deaktiviert, wenn Dateneingabe konfiguriert ist, denn beim Betätigen von Schaltflächen mit Dateneingabefeldern wird die Bearbeitung aktiviert.

Die primitivenspezifische Eigenschafts-Registerkarte für eine abgeschrägte Schaltfläche sehen Sie unten:



Details zu den Standardeinstellungen finden Sie im vorherigen Kapitel.

Die primitivenspezifische Eigenschafts-Registerkarte für eine abgestufte Schaltfläche sehen Sie unten:



Details zu den Standardeinstellungen finden Sie im vorherigen Kapitel.

9.2 Messgerät-Primitive

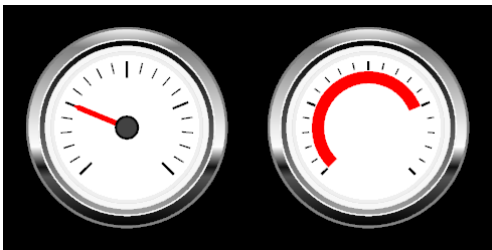
Crimson 3.1 unterstützt eine Vielzahl von linearen und radialen Messgeräten, die zum Anzeigen von Prozessvariablen in einem intuitiven und leicht lesbaren Format verwendet werden können. Alle Messgeräte haben ähnliche Verhaltensweisen und ähnliche Konfigurationseigenschaften. Messgeräte zeigen an sich keine Daten in Textform an, doch die Anzeige kann mit Text- und Datenfeldern erweitert werden.

9.2.1 Messgerät-Konzepte

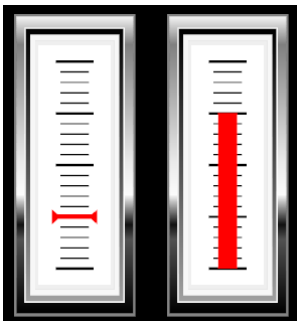
In den folgenden Abschnitten werden verschiedene Konzepte beschrieben, die für alle Messgeräte gelten.

9.2.1.1 Typen

Radiale Messgeräte zeigen einen Wert mittels eines sich drehenden Zeigers oder eines sich radial ausdehnenden Bandes an:

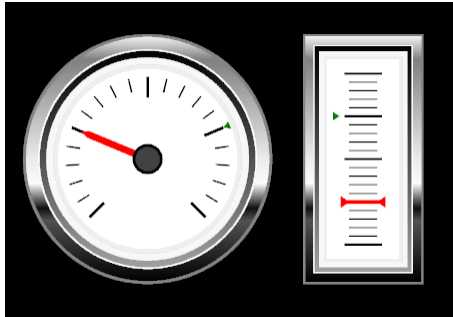


Lineare Messgeräte zeigen einen Wert mithilfe einer beweglichen Linie oder eines farbigen Balkens an:



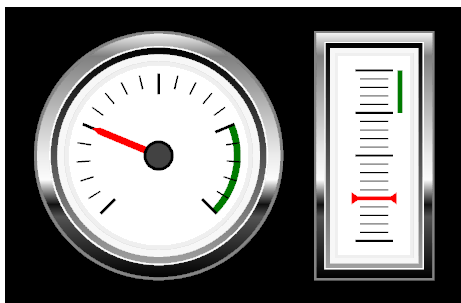
9.2.1.2 Fehler

Einstellmarken („Bugs“) sind dreieckige Markierungen außen an der Messgeräteskala. Sie werden zur Kennzeichnung von Sollwerten oder anderen wichtigen Werten verwendet. Jedes Messgerät kann bis zu zwei Einstellmarken unterstützen, wobei Farbe und Position dynamisch über Tags einstellbar sind. In der nachstehenden Abbildung sehen Sie eine einzelne grüne Einstellmarke auf einer radialen und einer linearen Anzeige:



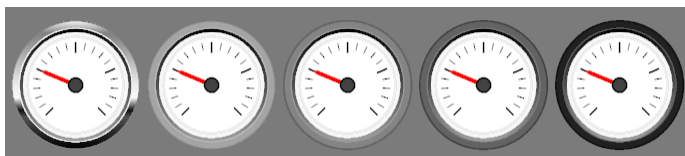
9.2.1.3 Bänder

Bänder sind radiale oder lineare Bereiche, die außerhalb der Messgeräteskala gezeichnet sind. Bänder dienen zur Kennzeichnung wichtiger Betriebsbereiche. Ein grünes Band könnte beispielsweise den Zielbereich für eine Prozessvariable angeben und ein rotes Band einen Bereich, der beim Betrieb vermieden werden soll. Jedes Messgerät kann bis zu zwei Bänder unterstützen, wobei Farbe und Position dynamisch über Tags einstellbar sind. In der nachstehenden Abbildung sehen Sie ein grünes Band auf einem radialen und einem linearen Messgerät:



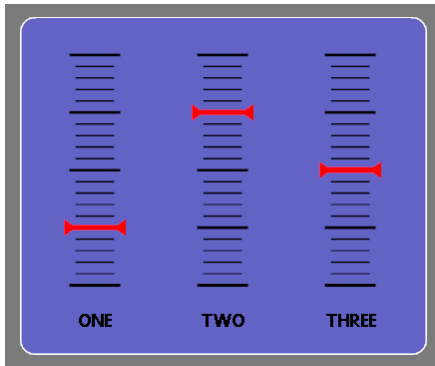
9.2.1.4 Stile

Messgeräte können in verschiedenen Stilen erstellt werden, wobei jeder Stil vordefinierte Farben auf die Elemente des Messgeräts anwendet. Mit einem benutzerdefinierten Stil kann jede Farbe manuell festgelegt werden, sodass eine Vielzahl von visuellen Designs erstellt werden kann. Nachstehend ist ein radiales Messgerät in den fünf vordefinierten Stilen abgebildet:



9.2.1.5 Nackte Messgeräte

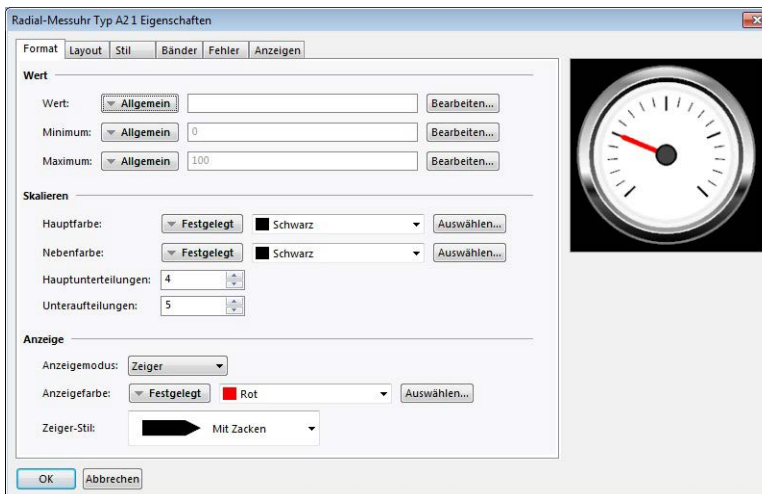
Ein nacktes Messgerät ist eines, das nur den Zeiger, die Skala, die Einstellmarken und Bänder anzeigt. Es zeigt weder die Einfassung an noch den Hintergrund, vor dem sich der Zeiger bewegt. Zum Erstellen nackter Messgeräte wählen Sie auf der Seite „Style“ (Stil) die Option „Naked“ (Nackt) aus. Mit dieser Option können Sie Messgeräte mit benutzerdefinierten Einfassungen erstellen oder mehrere Messgeräte innerhalb einer einzigen Figur kombinieren. Nachstehend sind drei nackte lineare Messgeräte in einem abgerundeten Rechteck abgebildet:



9.2.2 Eigenschaften von Messgeräten

Alle Messgeräte haben ähnliche Eigenschaften.

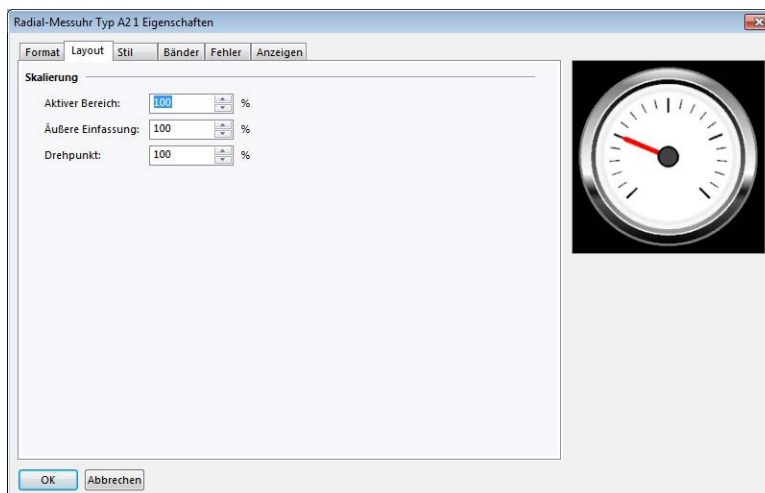
9.2.2.1 Formateigenschaften



- Die Eigenschaft *Value* (Wert) gibt den Wert an, der vom Messgerät angezeigt werden soll. Die Eigenschaften *Minimum* und *Maximum* geben den Bereich an, den die Skala des Messgeräts abdecken soll. Der Bereich wird nicht nur für die Skalierung der Eigenschaft „Value“ (Wert) verwendet, sondern auch zur Bestimmung der Position von Einstellmarken und Bändern.
- Die Eigenschaften *Major Color* (Hauptfarbe) und *Minor Color* (Nebenfarbe) geben die Farben an, die für die Haupt- und Unterabschnitte der Skala verwendet werden sollen. Diese Eigenschaften können animiert werden, damit die Skala wichtige Änderungen im Prozess anzeigt.

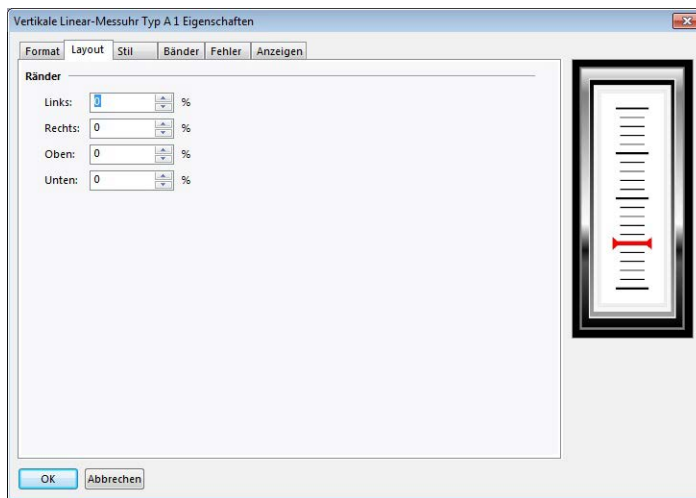
- Die Eigenschaften *Major Divisions* (Hauptunterteilungen) und *Minor Divisions* (Nebenunterteilungen) steuern, wie die Skala aufgeteilt wird. Beachten Sie, dass die Unterteilungen ohne Bezug auf die Mindest- und Höchstwerte angewendet werden. Wenn Sie die *Minor Divisions* (Nebenunterteilungen) auf 1 setzen, wird nur die Hauptunterteilung angezeigt. Die Sub-Pixel-Grafikmaschine von Crimson sorgt dafür, dass selbst umfangreiche Unterteilungen deutlich angezeigt werden können.
- Mit der Eigenschaft *Display Mode* (Anzeigemodus) geben Sie an, ob Sie einen Zeiger oder ein sich ausdehnendes Band verwenden möchten. Am Anfang dieses Abschnitts sind Beispiele für die einzelnen Modi von linearen und radialen Messgeräten abgebildet.
- Die Eigenschaft *Display Color* (Anzeigefarbe) steuert die Farbe des Zeigers oder Bandes. Beachten Sie, dass diese Eigenschaft animiert werden kann, um die Aufmerksamkeit auf wichtige Änderungen im Prozess zu lenken.
- Die Eigenschaft *Pointer Style* (Zeiger-Stil) definiert, wie das äußere Ende eines radialen Zeigers oder beide Enden eines linearen Zeigers gezeichnet werden. Die Eigenschaft wird bei einem Band nicht verwendet.

9.2.2.2 Layouteigenschaften für radiale Messgeräte



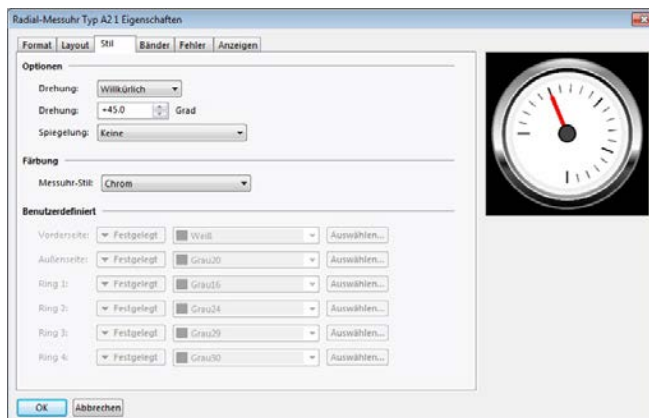
- Mit den Eigenschaften unter „Scaling“ (Skalierung) kann die radiale Größe der einzelnen Elemente der Messuhren angepasst werden, um ein anderes visuelles Ergebnis zu erzielen oder um Platz für zusätzliche Text- oder Datenfelder zu schaffen. Die Auswirkungen der einzelnen Werte lassen sich am besten nachprüfen, indem Sie Änderungen vornehmen und sich die Ergebnisse im Dialogfeld „Eigenschaften“ des Primitivs ansehen.

9.2.2.3 Layouteigenschaften für lineare Messgeräte



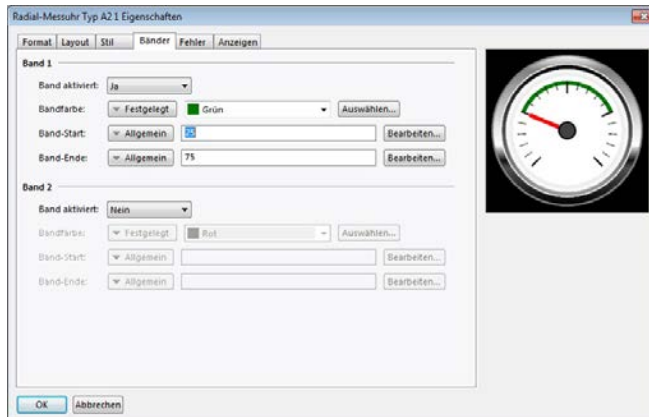
- Mit den Eigenschaften unter „Margins“ (Ränder) kann der Rand um den aktiven Bereich des Messgeräts angepasst werden, um Platz für zusätzliche Text- oder Datenfelder zu schaffen. Wenn Sie beispielsweise den unteren Rand vergrößern, kann unterhalb der Skala eine Beschriftung platziert werden.

9.2.2.4 Stileigenschaften



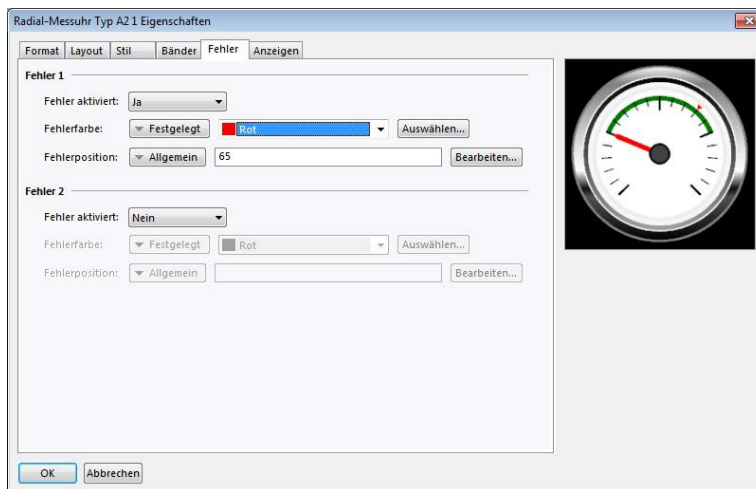
- Mit den Eigenschaften *Rotation* (Drehung) und *Reflexion* (Spiegelung) kann das Messgerät transformiert werden, um unterschiedliche Ausrichtungen und unterschiedliche Richtungen der Zeigerbewegung zu erreichen. Die Eigenschaften können auch über das Menü „Transform“ (Transformieren) oder über die Symbole in der Schnellleiste des Primitivs angezeigt werden. Mit der Einstellung „Arbitrary rotation“ (Beliebige Rotation) kann für die zweite Eigenschaft *Rotation* ein anderer Wert als ein Vielfaches von 90° ausgewählt werden. Um einen beliebigen Winkel gedrehte Messinstrumente sind insofern eingeschränkt, dass sie immer in ihr Begrenzungsrechteck passen müssen. Dies kann zu nicht intuitiven Ergebnissen führen, mit denen Sie sich durch Experimentieren aber vertraut machen können.
- Die Eigenschaft *Gauge Style* (Messuhr-Stil) ermöglicht die Auswahl eines vordefinierten Stils oder benutzerdefinierter Farben. Die Eigenschaft ermöglicht auch die Auswahl des Modus „Naked“ (Nackt), die weiter oben in diesem Abschnitt beschrieben ist.
- Mit den Eigenschaften unter „Custom“ (Benutzerdefiniert) können die Farben geändert werden, die den verschiedenen Elementen der Messgeräte-Einfassung zugewiesen sind. Durch die Anpassung der verschiedenen Einstellungen lassen sich interessante Ergebnisse erzielen.

9.2.2.5 Bandeigenschaften



- Mit der Eigenschaft *Band Enabled* (Band aktiviert) wird das zugehörige Band aktiviert bzw. deaktiviert.
- Mit der Eigenschaft *Band Color* (Bandfarbe) wird die Farbe des Bands festgelegt. Beachten Sie, dass diese Farbe animiert werden kann, um die Aufmerksamkeit auf Änderungen im Prozess zu lenken.
- Mit den Eigenschaften *Band Start* (Band-Beginn) und *Band End* (Band-Ende) wird der Bereich festgelegt, über den sich das Band erstrecken soll. Die Werte beziehen sich auf die Eigenschaften *Minimum* und *Maximum*, die auf der Registerkarte „Format“ des Messgeräts festgelegt werden.

9.2.2.6 Eigenschaften von Einstellmarken



- Mit der Eigenschaft *Bug Enabled* (Marke aktiviert) wird die zugehörige Einstellmarke aktiviert bzw. deaktiviert.
- Mit der Eigenschaft *Bug Color* (Markenfarbe) wird die Farbe der Einstellmarke festgelegt. Beachten Sie, dass diese Farbe animiert werden kann, um die Aufmerksamkeit auf Änderungen im Prozess zu lenken.
- Die Eigenschaft *Bug Position* (Markenposition) definiert die Position der Einstellmarke. Der Wert bezieht sich auf die Eigenschaften *Minimum* und *Maximum*, die auf der Registerkarte „Format“ des Messgeräts festgelegt werden.

9.3 Balken- und Liniendiagramme

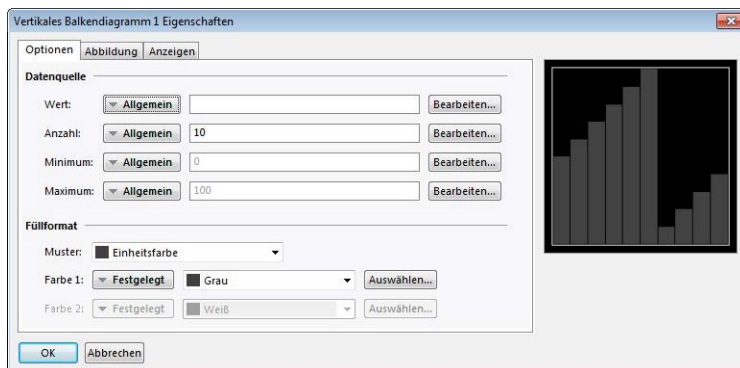


Balkendiagramme zeigen einen oder mehrere Werte aus einem Array als eine Reihe von horizontalen Balken an. Mit Streudiagrammen können bis zu vier verknüpfte Sätze von x- und y-Werten dargestellt und durch optionale Datenmarkierungen sowie eine Verbindungslinie ergänzt werden. Sie bieten außerdem die Möglichkeit, eine Regressionslinie hinzuzufügen, die die passendste lineare Beziehung anzeigt, die den einzelnen Datensätze zugeordnet ist.

9.3.1 Die Balkendiagramm-Primitive

Balkendiagramme gibt es in horizontaler und vertikaler Form.

9.3.1.1 Optionseigenschaften



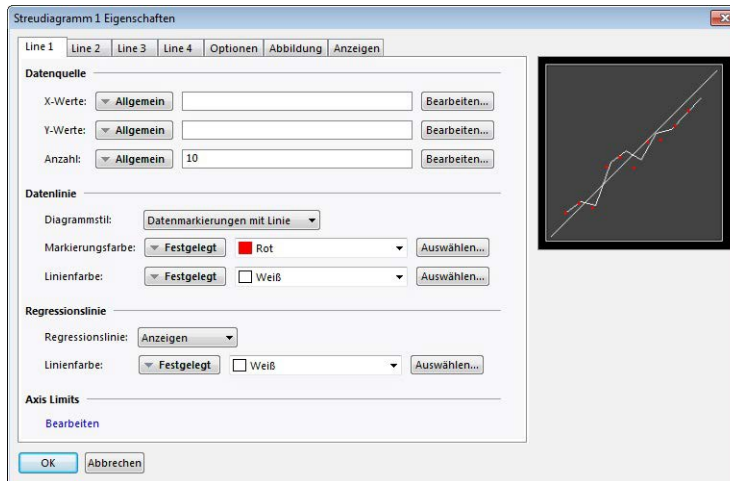
- Die Eigenschaft *Value* (Wert) definiert ein Array-Element als Quelle der anzuzeigenden Daten. Das Element wird für den ersten Balken verwendet. Insgesamt werden *Count* (Anzahl) Elemente angezeigt, wobei die jeweils nachfolgenden Array-Elemente verwendet werden.
- Mit den Eigenschaften *Minimum* und *Maximum* werden die unteren und oberen Grenzwerte des Anzeigebereichs definiert. Balken, die außerhalb dieses Bereichs liegen, werden abgeschnitten.
- Die Eigenschaftsgruppe „Fill Format“ (Füllformat) definiert die Farbe, die für die Balken verwendet werden soll.

9.3.1.2 Figureigenschaften

Die Eigenschaften auf der Registerkarte „Figure“ (Figur) steuern den Hintergrund des Diagramms und sind konventionell.

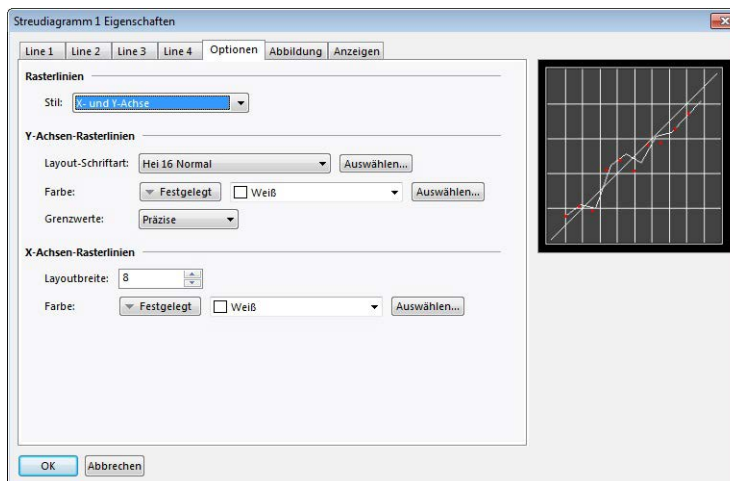
9.3.2 Eigenschaften von Streudiagrammen

9.3.2.1 Linieneigenschaften



- Die Eigenschaften *X Values* (X-Werte) und *Y Values* (Y-Werte) definieren Array-Elemente als Quelle des ersten zu zeichnenden Datenpunkts. Insgesamt werden *Count* (Anzahl) Elemente angezeigt, wobei die jeweils nachfolgenden Array-Elemente verwendet werden. Wenn *Count* (Anzahl) auf Null gesetzt ist, wird der Datensatz nicht angezeigt.
- Die Eigenschaft *Graph Style* (Diagrammstil) definiert, ob Datenmarkierungen gezeichnet werden und ob die Datenpunkte mit Linien verbunden werden.
- Die Eigenschaften *Marker Color* (Markierungsfarbe) und *Line Color* (Linienfarbe) definieren die Farben, die zur Darstellung der jeweiligen Elemente verwendet werden sollen.
- Die Eigenschaft *Regression Line* (Regressionslinie) legt fest, ob Crimson die lineare Beziehung, die am besten für die Gruppe von Punkten geeignet ist, berechnen und dann anhand einer Linie darstellen soll. Mit der Eigenschaft *Line Color* (Linienfarbe) kann die Farbe ausgewählt werden.

9.3.2.2 Optionseigenschaften



- Die Eigenschaft *Style* (Stil) steuert, welcher Satz bzw. welche Sätze von Rasterlinien aktiviert werden.

- Die Eigenschaftengruppe „Y Axis Grid Lines“ (Y-Achsen-Rasterlinien) steuert, wie die horizontalen Rasterlinien gezeichnet und angeordnet werden. Das System bestimmt anhand eines automatischen Algorithmus die beste Möglichkeit, die Linien auf Basis der Mindest- und Höchstwerte der Linie 1 zu platzieren. Die Eigenschaften *Layout Font* (Layout-Schriftart) und *Limit Values* (Grenzwerte) entsprechen denen des älteren Primitivs für Skalen, d. h. bei Übereinstimmung mit diesen Einstellungen kann eine Skala neben dem Streudiagramm platziert und an den zugehörigen Rasterlinien ausgerichtet werden.
- Die Eigenschaftengruppe „X Axis Grid Lines“ (X-Achsen-Rasterlinien) steuert, wie die vertikalen Rasterlinien gezeichnet und angeordnet werden. Das System bestimmt anhand eines automatischen Algorithmus die beste Möglichkeit, die Linien auf Basis der Anzahl der Punkte für Linie 1 und dem durch die Eigenschaft *Layout Width* (Layoutbreite) angegebenen Mindestabstand zu platzieren.

9.3.2.3 Figureigenschaften

Die Eigenschaften auf der Registerkarte „Figure“ (Figur) steuern den Hintergrund des Diagramms und sind konventionell.

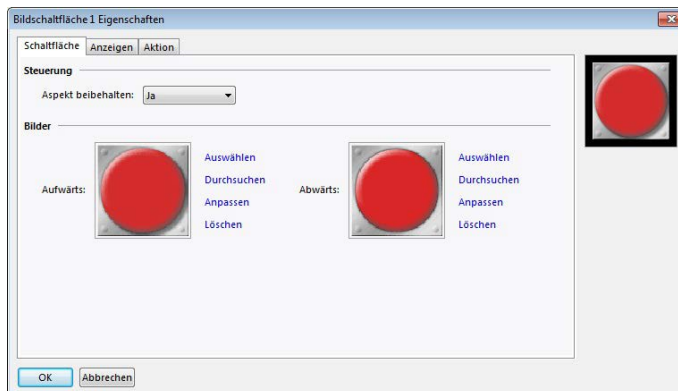
9.4 Aktionsschaltflächen



Aktionsschaltflächen verwenden vorausgewählte Bilder aus der Symbolbibliothek, um eine Schaltfläche zu erstellen, die eine bestimmte Aktion ausführt, wenn sie betätigt wird. Es werden mehr Versionen bereitgestellt als in der obigen Abbildung gezeigt. Durch Klicken auf eine Schaltfläche im Ressourcenfenster werden die verschiedenen Farbvarianten angezeigt, die verfügbar sind. Die quadratische Schaltfläche ist beispielsweise in rot, grün oder schwarz verfügbar:



Wenn Sie eine Aktionsschaltfläche verwenden, verwenden Sie im Dialogfeld „Eigenschaften“ in erster Linie die Registerkarte „Actions“ (Aktionen), um gemäß der Beschreibung im vorherigen Kapitel eine Aktion zu definieren. Über die Registerkarte „Button“ (Schaltfläche) können Sie auch die Schaltflächenbilder anpassen oder eigene Versionen definieren:



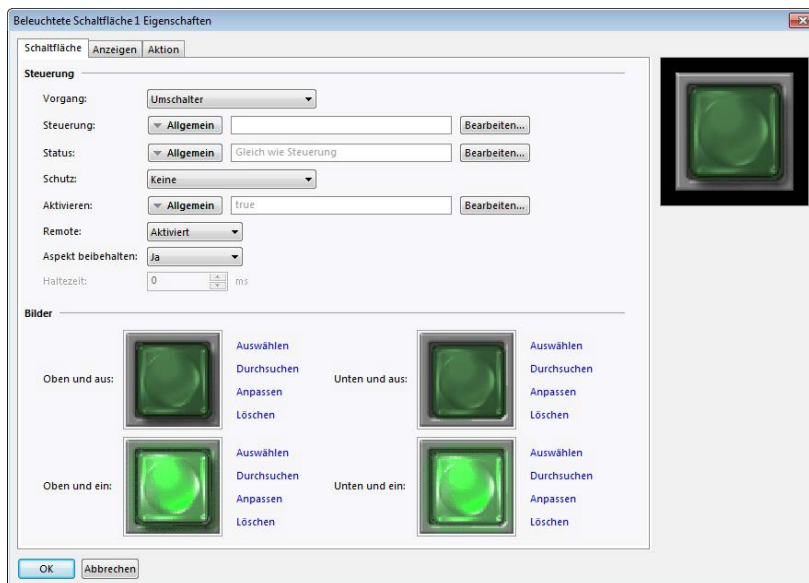
9.5 Beleuchtete Schaltflächen



Beleuchtete Schaltflächen verwenden vorausgewählte Bilder aus der Symbolbibliothek, um eine Schaltfläche zu erstellen, die ein Tag steuert. Sie leuchten ausgehend vom Status dieses Tags oder dem Status eines anderen Ausdrucks auf. Es werden mehr Versionen bereitgestellt als in der obigen Abbildung gezeigt. Durch Klicken auf eine Schaltfläche im Ressourcenfenster werden die verschiedenen Farbvarianten angezeigt, die verfügbar sind. Die unten abgebildete Candy-Schaltfläche ist beispielsweise in den Farben Rot, Grün, Gelb, Blau oder Grau verfügbar:



Die primitivenspezifische Eigenschafts-Registrierkarte für diese Primitiven sieht so aus:



- Die Eigenschaft *Operation* (Vorgang) bestimmt das gewünschte Verhalten:

Vorgang	Schaltflächenverhalten
Umschalten	Änderung des Datenstatus, wenn das Primitiv gedrückt wird
Zwischenspeichern	Sind die Daten 0, werden sie bei Betätigen der Schaltfläche auf 1 gesetzt. Sind die Daten 1, werden sie bei Loslassen der Schaltfläche auf 0 gesetzt.
NEIN Vorübergehend	Daten werden auf 1 gesetzt, wenn das Primitiv betätigt wird. Daten werden auf 0 gesetzt, wenn das Primitiv losgelassen wird.
NC Vorübergehend	Daten werden auf 0 gesetzt, wenn das Primitiv betätigt wird. Daten werden auf 1 gesetzt, wenn das Primitiv losgelassen wird.
Einschalten	Daten werden auf 1 gesetzt, wenn das Primitiv betätigt wird.
Ausschalten	Daten werden auf 0 gesetzt, wenn das Primitiv betätigt wird.

- Beachten Sie, dass sich „Zwischenspeichern“ und „Umschalten“ in Bezug auf den Punkt unterscheiden, an dem ein Kontrollwert von nicht Null auf Null zurückgesetzt wird. Bei „Umschalten“ werden alle Änderungen durchgeführt, wenn die Schaltfläche betätigt wird. Bei „Zwischenspeichern“ wird ein Wert ausgeschaltet, wenn die Schaltfläche losgelassen wird. Das Ergebnis entspricht eher dem Verhalten einer realen Drucktaste.
- Die Eigenschaft *Control* (Steuerung) definiert den Wert, der beim Betätigen oder Loslassen der Schaltfläche geschrieben werden soll. Dieser Wert muss schreibbar sein und wird je nach dem für die Schaltfläche definierten genauen Vorgang auf 1 oder 0 gesetzt.
- Mit der Eigenschaft *Status* wird die Beleuchtung der Schaltfläche gesteuert. Wenn sie leer gelassen wird, wird standardmäßig der Wert unter „Control“ (Steuerung) übernommen. Wenn eine komplexere Logik erforderlich ist, kann ein anderer Wert verwendet werden.
- Mit der Eigenschaft *Hold Time* (Haltezeit) wird bei Drucktastern sichergestellt, dass die mit dem gedrückten Zustand assoziierten Daten zumindest eine bestimmte Zeit lang geschrieben werden, selbst wenn die Taste gedrückt und sofort losgelassen wird. Mit ihrer Hilfe kann sichergestellt werden, dass das externe Gerät die Tastenaktivierung unter Verhältnissen erkennt, in denen die Transaktionsschreibvorgänge von Crimson unzureichend sind oder deaktiviert wurden. Es wird davon abgeraten, sich auf diese Eigenschaft zu verlassen, da bei ihrer Justierung die Kommunikationsverzögerungen im System berücksichtigt werden müssen.

Details zu den Eigenschaften *Protection* (Schutz), *Enable* (Aktivieren) und *Remote* (Remote) finden Sie im vorherigen Kapitel.

Weitere Informationen zum Ändern oder Anpassen der Schaltflächenbilder finden Sie weiter oben in diesem Kapitel.

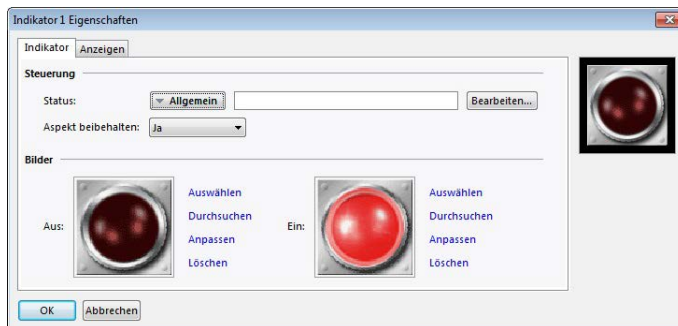
9.6 Indikatoren



Indikatoren verwenden vorausgewählte Bilder aus der Symbolbibliothek, um den Ein/Aus-Status eines Datenwerts anzuzeigen. Es werden mehr Versionen bereitgestellt als in der obigen Abbildung gezeigt. Durch Klicken auf eine Schaltfläche im Ressourcenfenster werden die verschiedenen Farbvarianten angezeigt, die verfügbar sind. Der unten abgebildete Indikator „Pilot“ ist beispielsweise in den Farben Rot, Grün, Gelb, Blau oder Weiß verfügbar:



Indikatoren verfügen über eine sehr einfache Gruppe von Eigenschaften:



Die Eigenschaft *Status* steuert die zu zeichnenden Bilder. Alle anderen Eigenschaften sind Standardeigenschaften.

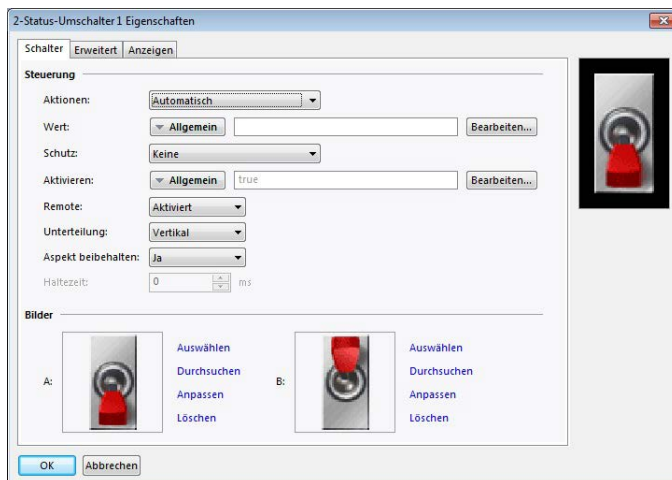
9.7 Umschalter mit Zweifachstatus



Umschalter mit Zweifachstatus verwenden vorausgewählte Bilder aus der Symbolbibliothek, um Kippschalter mit Auf- und Ab-Positionen implementieren. Es werden mehr Versionen bereitgestellt als in der obigen Abbildung gezeigt. Durch Klicken auf einen Schalter im Ressourcenfenster werden die verschiedenen Farbvarianten angezeigt, die verfügbar sind. Der Kippschalter ist beispielsweise in rot, grün oder schwarz verfügbar:



9.7.1 Schaltereigenschaften



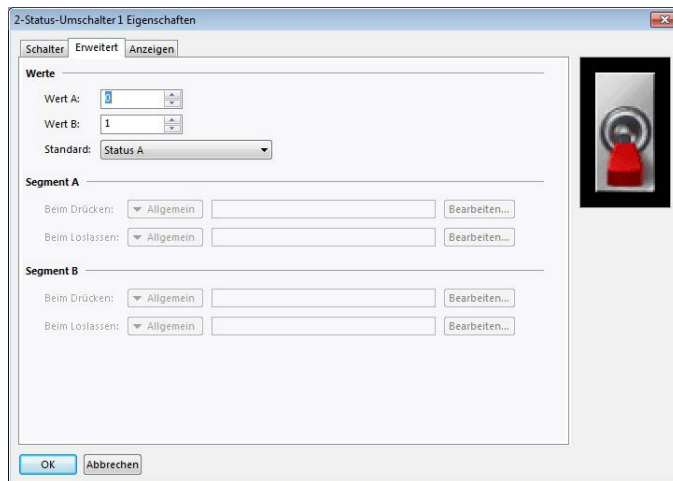
- Die Eigenschaft *Actions* (Aktionen) steuert das Verhalten des Schalters. Die drei automatischen Modi modellieren konventionelle Kippschalter oder vorgespannte Schalter. Der Modus „User-Defined“ (Benutzerdefiniert) ermöglicht Ihnen, komplexere Aktionen festzulegen, die auftreten, wenn eine Hälfte des Kippschalters gedrückt oder losgelassen wird.
- Die Eigenschaft *Value* (Wert) wird bei den automatischen Modi verwendet und bei Änderung der Schalterposition in die Datenwerte geschrieben, die den Stellungen A und B zugeordnet sind. Standardmäßig wird Stellung A durch eine Null und Stellung B durch eine Eins dargestellt. Diese Werte können jedoch mithilfe der erweiterten Einstellungen für dieses Primitiv geändert werden.
- Die Eigenschaft *Division* (Unterteilung) definiert, ob der Schalter vertikal oder horizontal betätigt wird, und definiert somit, wie Crimson das Primitiv beim Interpretieren der Berührungen durch den Benutzer unterteilen soll.
- Mit der Eigenschaft *Hold Time* (Haltezeit) wird bei vorgespannten Schaltern sichergestellt, dass die mit dem gedrückten Status verbundenen Daten zumindest eine spezifizierte Zeit lang geschrieben wer-

den, selbst wenn der Schalter umgelegt und sofort losgelassen wird. Mit ihrer Hilfe kann sichergestellt werden, dass das externe Gerät die Schalteraktivierung in Situationen erkennt, in denen die Transaktionsvorgänge von Crimson unzureichend sind oder deaktiviert wurden. Es wird davon abgeraten, sich auf diese Eigenschaft zu verlassen, da bei ihrer Justierung die Kommunikationsverzögerungen im System berücksichtigt werden müssen.

Details zu den Eigenschaften *Protection* (Schutz), *Enable* (Aktivieren) und *Remote* (Remote) finden Sie im vorherigen Kapitel.

Weitere Informationen zum Ändern oder Anpassen der Schalterbilder finden Sie weiter oben in diesem Kapitel.

9.7.2 Erweiterte Eigenschaften



- Die Eigenschaften *Value A* (Wert A) und *Value B* (Wert B) definieren die Datenwerte, die in den automatischen Modi zur Darstellung der Schalterstellung verwendet werden. Der Wert, der aus der Eigenschaft „Value“ (Wert) gelesen wird, wird mit diesen beiden Werten verglichen, um zu entscheiden, welche Stellung angezeigt werden soll. Ebenso wird bei der Änderung der Schalterstellung der entsprechende Wert geschrieben.
- Mit der Eigenschaft *Default* (Standard) wird die Stellung ausgewählt, die angezeigt werden soll, wenn die von der Eigenschaft „Value“ (Wert) gelesenen Daten weder mit Wert A noch mit Wert B übereinstimmen.
- Die Eigenschaften *On Pressed* (Bei Drücken) und *On Released* (Bei Loslassen) definieren benutzerdefinierte Verhaltensweisen, die auszuführen sind, wenn die A- und B-Teile des Schalters vom Benutzer gedrückt bzw. losgelassen werden. Bei einem vertikalen Schalter ist A die untere Hälfte und B die obere. Bei einem vertikalen Schalter ist A die linke Hälfte und B die rechte.

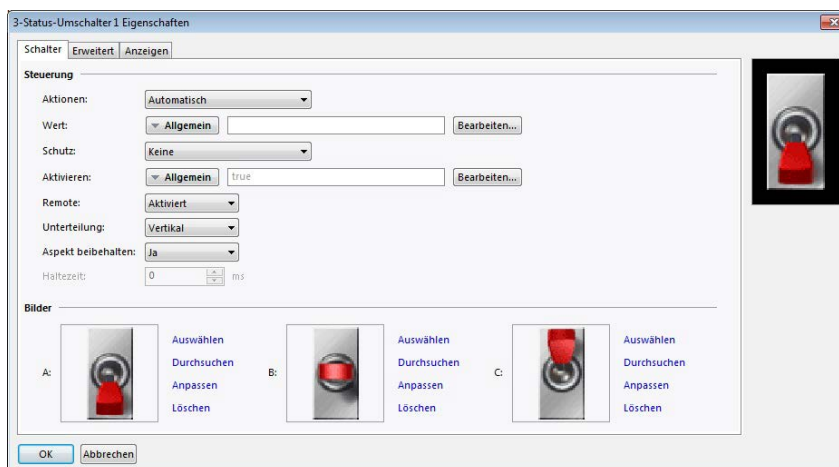
9.8 Umschalter mit Dreifachstatus



Umschalter mit Dreifachstatus verwenden vorausgewählte Bilder aus der Symbolbibliothek, um Kippschalter mit den Stellungen „Oben“, „Mitte“ und „Unten“ zu implementieren. Es werden mehr Versionen bereitgestellt als in der obigen Abbildung gezeigt. Durch Klicken auf einen Schalter im Ressourcenfenster werden die verschiedenen Farbvarianten angezeigt, die verfügbar sind. Der Kippschalter ist beispielsweise in drei Farben verfügbar:



9.8.1 Schaltereigenschaften



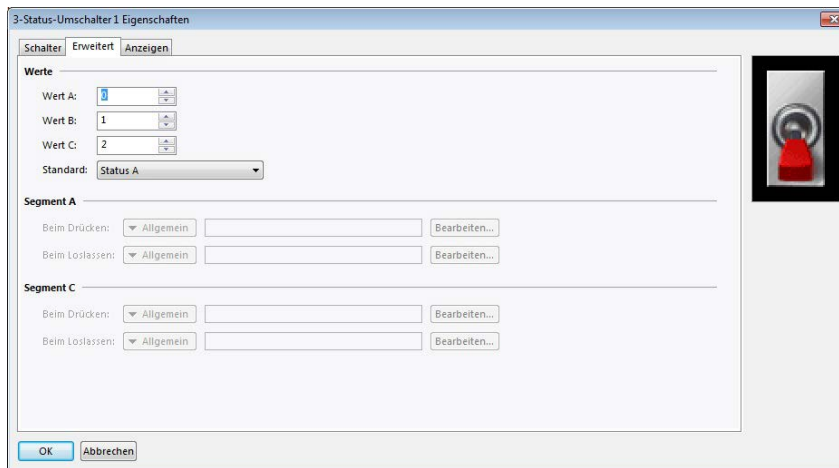
- Die Eigenschaft *Actions* (Aktionen) steuert das Verhalten des Schalters. Die vier automatischen Modi modellieren konventionelle Kippschalter oder vorgespannte Schalter. Der Modus „User-Defined“ (Benutzerdefiniert) ermöglicht Ihnen, komplexere Aktionen festzulegen, die auftreten, wenn eine Hälfte des Kippschalters gedrückt oder losgelassen wird. Beachten Sie, dass der Schalter nur um jeweils eine Position bewegt werden kann. Das heißt, dass der Schalter beim Umschalten von Stellung A in Stellung C zwangsläufig durch Stellung B bewegt wird, genau wie bei einem echten Kippschalter.
- Die Eigenschaft *Value* (Wert) wird von den automatischen Modi verwendet und bei Änderung der Schalterposition in die Datenwerte geschrieben, die den Stellungen A, B und C zugeordnet sind. Standardmäßig wird Stellung A durch eine Null, Stellung B durch eine Eins und Stellung C durch eine Zwei dargestellt. Diese Werte können jedoch mithilfe der erweiterten Einstellungen für dieses Primitiv geändert werden.
- Die Eigenschaft *Division* (Unterteilung) definiert, ob der Schalter vertikal oder horizontal betätigt wird, und definiert somit, wie Crimson das Primitiv beim Interpretieren der Berührungen durch den Benutzer unterteilen soll.

- Mit der Eigenschaft *Hold Time* (Haltezeit) wird bei vorgespannten Schaltern sichergestellt, dass die mit dem gedrückten Status verbundenen Daten zumindest eine spezifizierte Zeit lang geschrieben werden, selbst wenn der Schalter umgelegt und sofort losgelassen wird. Mit ihrer Hilfe kann sichergestellt werden, dass das externe Gerät die Schalteraktivierung in Situationen erkennt, in denen die Transaktionsschreibvorgänge von Crimson unzureichend sind oder deaktiviert wurden. Es wird davon abgeraten, sich auf diese Eigenschaft zu verlassen, da bei ihrer Justierung die Kommunikationsverzögerungen im System berücksichtigt werden müssen.

Details zu den Eigenschaften *Protection* (Schutz), *Enable* (Aktivieren) und *Remote* (Remote) finden Sie im vorherigen Kapitel.

Weitere Informationen zum Ändern oder Anpassen der Schalterbilder finden Sie weiter oben in diesem Kapitel.

9.8.2 Erweiterte Eigenschaften



- Die Eigenschaften *Value A* (Wert A), *Value B* (Wert B) und *Value C* (Wert C) definieren die Datenwerte, die in den automatischen Modi zur Darstellung der drei Schalterstellungen verwendet werden. Der Wert, der aus der Eigenschaft „Value“ (Wert) gelesen wird, wird mit diesen Werten verglichen, um zu entscheiden, welche Stellung angezeigt werden soll. Bei Änderung der Schalterposition wird der entsprechende Wert geschrieben.
- Mit der Eigenschaft *Default* (Standard) wird die Stellung ausgewählt, die angezeigt werden soll, wenn die aus der Eigenschaft „Value“ (Wert) gelesenen Daten weder mit Wert A noch mit Wert B oder C übereinstimmen.
- Die Eigenschaften *On Pressed* (Bei Drücken) und *On Released* (Bei Loslassen) definieren benutzerdefinierte Verhaltensweisen, die ausgeführt werden sollen, wenn die A- und C-Teile des Schalters vom Benutzer gedrückt oder losgelassen werden. Bei einem vertikalen Schalter ist A die untere Hälfte und C die obere. Bei einem vertikalen Schalter ist A die linke Hälfte und C die rechte.

9.9 Selektoren mit Zweifachstatus



Selektoren mit Zweifachstatus verwenden vorausgewählte Bilder aus der Symbolbibliothek, um Drehwahlschalter mit zwei Stellungen zu implementieren. Ihr Verhalten ist identisch mit dem von Umschaltern mit Zweifachstatus und sie werden anhand desselben Primitivs implementiert.

9.10 Selektoren mit Dreifachstatus

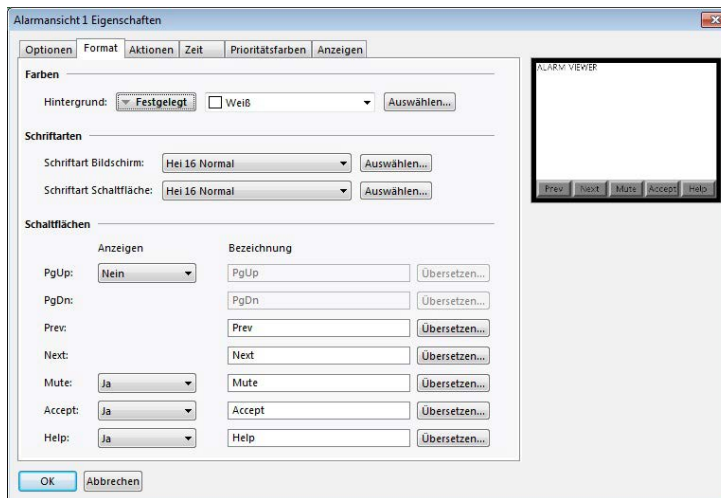


Selektoren mit Dreifachstatus verwenden vorausgewählte Bilder aus der Symbolbibliothek, um Drehwahlschalter mit drei Stellungen zu implementieren. Ihr Verhalten ist identisch mit dem von Umschaltern mit Dreifachstatus und sie werden anhand desselben Primitivs implementiert.

9.11 System-Primitive

9.11.1 Betrachterformat

Die meisten System-Primitive dienen zum Anzeigen oder Bearbeiten von Daten, die von Crimson erstellt oder abgerufen werden. Jeder Betrachter besteht aus einem Betrachtungsbereich mit einer Reihe von Schaltflächen darunter. Das Erscheinungsbild der listenbasierten Betrachter wird über die Registerkarte „Format“ des Dialogfelds „Eigenschaften“ gesteuert:

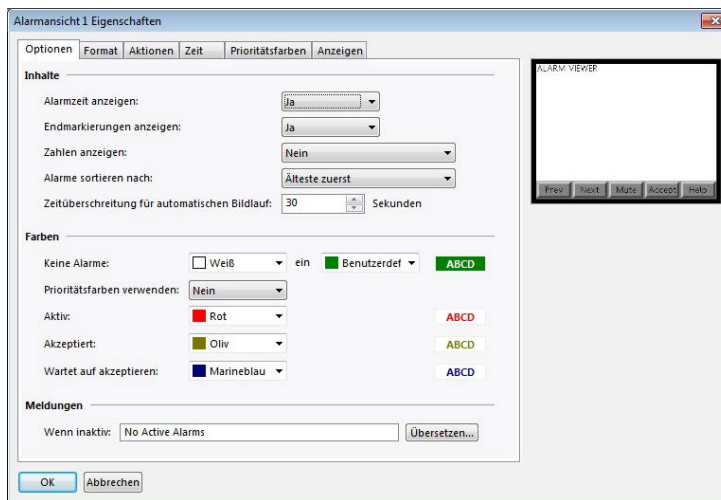


Farben und Schriftarten werden auf die übliche Art und Weise festgelegt. Im Abschnitt „Buttons“ (Schaltflächen) können einige der im unteren Bereich des Betrachters angezeigten Schaltflächen deaktiviert werden. Es ist auch möglich, die Beschriftungen zu bearbeiten oder für internationale Anwendungen zu übersetzen. Denken Sie daran, dass übersetzbare Zeichenfolgen auf Ausdrücke festgelegt werden können, damit die Beschriftung einer Schaltfläche zur Laufzeit angepasst werden kann.

9.11.2 Der Alarmbetrachter

Mit dem Alarmbetrachter können Alarme innerhalb des Systems angezeigt und angenommen werden.

9.11.2.1 Optionseigenschaften



- Mit der Eigenschaft *Show Alarm Time* (Alarmzeit anzeigen) wird angegeben, ob jedem Alarm ein Präfix mit dem Datum und der Uhrzeit seines Auftretens vorangestellt werden soll. Das zu verwendende Zeitformat wird auf der Registerkarte „Time“ (Zeit) festgelegt.
- Mit der Eigenschaft *Show End Markers* (Endmarkierungen anzeigen) können Sie angeben, ob die ersten und letzten Elemente in der Liste durch Markierungen gekennzeichnet werden sollen. Der Benutzer kann dann leichter erkennen, ob er sich an einem der Listenenden befindet.
- Mit der Eigenschaft *Show Numbers* (Zahlen anzeigen) wird angegeben, ob die Alarmzahlen angegeben werden sollen, und falls ja, wie viele Ziffern anzuzeigen sind.
- Mit der Eigenschaft *Sort Alarms By* (Alarmer sortieren nach) können Sie angeben, in welcher Reihenfolge die Alarmer im Betrachter sortiert werden sollen. Sie können die ältesten oder die jüngsten Alarmer zuerst anzeigen. Die Fähigkeit des Systems, die Uhrzeit zu erkennen, zu der ein Alarm aktiv wird, ist eine Funktion der Kommunikations- und Alarmaufgaben-Abtastzeit.
- Mit *Auto Scroll Timeout* (Zeitüberschreitung für automatischen Bildlauf) wird festgelegt, nach welchem Zeitraum der Cursor an den Anfang der Liste zurückkehrt. Die Liste wird dann wieder automatisch gescrollt, um neue Alarmer anzuzeigen. Setzen Sie diese Eigenschaft auf Null, um die Zeitüberschreitung zu deaktivieren.
- Die Eigenschaftsgruppe *Colors* (Farben) gibt die Textfarben an, die beim Anzeigen von Alarmen in verschiedenen Status verwendet werden sollen. Der aktive Status kann entweder durch eine einzelne Farbe oder durch verschiedene Farben dargestellt werden, die die Alarmprioritäten wiedergeben. Mit der Eigenschaft *Use Priority Colors* (Prioritätsfarben verwenden) wird einer der beiden Modi ausgewählt. Die Meldung „No Active Alarms“ (Keine aktiven Alarmer) unterstützt eine feste Hintergrundfarbe. Die anderen Status verwenden immer den Hintergrund des Primitivs selbst.
- Die Eigenschaft *When Inactive* (Wenn inaktiv) definiert die Zeichenfolge, die vom Primitiv angezeigt wird, wenn keine Alarmer aktiv sind. Diese Option wird in der Regel beim Erstellen mehrsprachiger Anwendungen für die Bereitstellung lokalisierter Texte verwendet.

9.11.2.2 Aktionseigenschaften

Wenn die Schaltfläche „Help“ (Hilfe) unten im Betrachter über die Registerkarte „Format“ aktiviert wurde, kann mit der Eigenschaft On Help (Bei Hilfe) auf der Registerkarte „Actions“ (Aktionen) eine Aktion definiert werden, die beim Betätigen der Schaltfläche ausgeführt werden soll. Während die Aktion ausgeführt wird, enthalten die folgenden Systemvariablen Informationen über den aktuell ausgewählten Alarm:

Variable	Inhalte
Daten	Eine gepackte Darstellung von Tag- und Alarmnummer, wobei die Tag-Nummer im Low Word steht und die Alarmnummer im High Word. Diese Variable ist ausschließlich für die Kompatibilität mit früheren Softwareversionen vorhanden.
t	Der Index des Tags, mit dem der Alarm verknüpft ist. Die Tag-Nummer für ein Tag wird rechts oben im Bearbeitungsfenster angezeigt, wenn ein Tag ausgewählt ist, und zwar direkt neben den Nach-oben- und Nach-unten-Schaltflächen, die zum Wechseln zwischen Tags verwendet werden.
i	Das mit dem Alarm verknüpfte Array-Element, wenn der Alarm aus einem Tag-Array stammt, oder andernfalls Null. Wie bereits erläutert, können Arrays über Alarme verfügen, die auf die ersten 256 Elemente des Arrays angewendet werden. Die Elemente sind von 0 bis 255 durchnummeriert.
a	Die Alarmnummer des Alarms 1, d. h. 1 für Alarm 1 und 2 für Alarm 2.

9.11.2.3 Zeiteigenschaften

Die Registerkarte „Time“ (Zeit) definiert das Format, das bei der Angabe des Zeitpunkts und des Datums verwendet werden soll, zu dem ein Alarm auftrat. Weitere Informationen finden Sie im Kapitel „Verwenden von Formaten“.

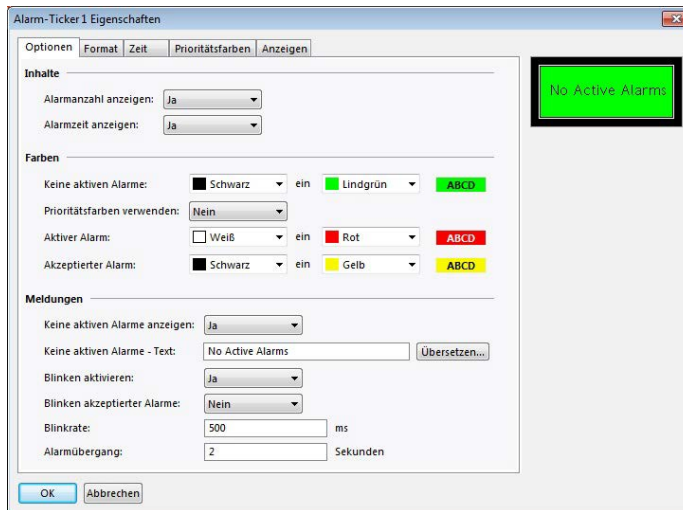
9.11.2.4 Farbeigenschaften

Die Registerkarte „Priority Colors“ (Prioritätsfarben) definiert die Farben, die auf die einzelnen Alarmprioritäten angewendet werden sollen. Diese Funktion ist nur aktiviert, wenn auf der Registerkarte „Options“ (Optionen) die Eigenschaft *Use Priority Colors* (Prioritätsfarben verwenden) ausgewählt ist.

9.11.3 Der Alarm-Ticker

Mit dem Alarm-Ticker wird eine scrollende Übersicht über die aktiven Alarme im System angezeigt. Er wird in der Regel am unteren Rand der Anzeigeseite platziert, möglicherweise auf einer Masterfolie, damit er immer sichtbar ist.

9.11.3.1 Optionseigenschaften



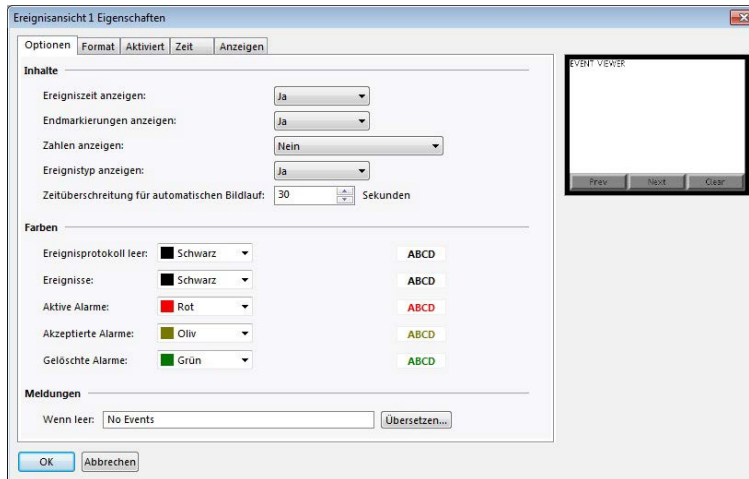
- Die Eigenschaft *Show Alarm Count* (Alarmanzahl anzeigen) steuert, ob die Anzahl der aktiven Alarme im Ticker angezeigt werden soll.
- Die Eigenschaft *Show Alarm Time* (Alarmzeit anzeigen) steuert, ob die Aktivierungszeit jedes Alarms im Ticker angezeigt werden soll.
- Die Eigenschaftsgruppe *Colors* (Farben) gibt die Textfarben an, die beim Anzeigen von Alarmen in verschiedenen Status verwendet werden sollen. Der aktive Status kann entweder durch eine einzelne Farbe oder durch verschiedene Farben dargestellt werden, die die Alarmprioritäten wiedergeben. Mit der Eigenschaft *Use Priority Colors* (Prioritätsfarben verwenden) wird einer der beiden Modi ausgewählt.
- Mit der Eigenschaft *Show No Active Alarms* (Keine aktiven Alarme anzeigen) wird festgelegt, ob das Primitiv eine Meldung anzeigt, wenn keine aktiven Alarme im System vorhanden sind, oder ob stattdessen eine leere Leiste angezeigt wird.
- Mit der Eigenschaft *No Active Alarms Text* (Text für Keine aktiven Alarme) wird der Text festgelegt, der angezeigt werden soll, wenn keine aktiven Alarme in der Datenbank vorhanden sind. Der Text kann übersetzt werden, damit er in jeder Sprache verfügbar ist.
- Die Eigenschaften *Enable Blinking* (Blinken aktivieren) und *Blink Accepted Alarms* (Blinken akzeptierter Alarme) steuern, ob aktive Alarme blinken, wenn sie angezeigt werden, und ob angenommene aktive Alarme ebenfalls blinken sollen. Die Eigenschaft *Blink Rate* (Blinkhäufigkeit) steuert, wie schnell geblinkt wird.
- Die Eigenschaft *Alarm Transition* (Alarmübergang) steuert, wie lange jeder Alarm in Crimson angezeigt wird, bevor der nächste Alarm in der aktiven Liste angezeigt wird. Der Standardwert lautet 2 Sekunden und eignet sich für fast alle Anwendungen.

Beachten Sie, dass es üblich ist, dem Alarm-Ticker eine Aktion hinzuzufügen, mit der der Benutzer zu einer Seite wechseln kann, auf der der vollständige Alarmbetrachter angezeigt wird. Das gleiche Ergebnis lässt sich auch mit einer Popup-Seite erzielen. Informationen zum Hinzufügen von Aktionen finden Sie im vorherigen Kapitel.

9.11.4 Der Ereignisbetrachter

Mit dem Ereignisbetrachter können Sie Ereignisse anzeigen und löschen, die vom System als Reaktion auf die von Daten-Tags erzeugten Alarme oder Ereignisse protokolliert werden.

9.11.4.1 Optionseigenschaften



- Mit der Eigenschaft *Show Event Time* (Ereigniszeit anzeigen) wird angegeben, ob jedem Ereignis ein Präfix mit dem Datum und der Uhrzeit seines Auftretens vorangestellt werden soll. Das zu verwendende Zeitformat wird auf der Registerkarte „Time“ (Zeit) festgelegt.
- Mit der Eigenschaft *Show End Markers* (Endmarkierungen anzeigen) können Sie angeben, ob die ersten und letzten Elemente in der Liste durch Markierungen gekennzeichnet werden sollen. Der Benutzer kann dann leichter erkennen, ob er sich an einem der Listenenden befindet.
- Mit der Eigenschaft *Show Numbers* (Zahlen anzeigen) wird angegeben, ob die Ereigniszahlen angegeben werden sollen, und falls ja, wie viele Ziffern anzuzeigen sind.
- Mit der Eigenschaft *Show Event Type* (Ereignistyp anzeigen) wird festgelegt, ob für jeden Eintrag angegeben werden soll, ob es sich um das Auftreten, die Annahme oder das Löschen eines Alarms bzw. um ein einzelnes Ereignis handelt. Wenn Alarme verwendet werden, führt die Nichtaktivierung dieser Einstellung zu Verwirrung bei den angezeigten Informationen.
- Mit *Auto Scroll Timeout* (Zeitüberschreitung für automatischen Bildlauf) wird festgelegt, nach welchem Zeitraum der Cursor an den Anfang der Liste zurückkehrt. Die Liste wird dann wieder automatisch gescrollt, um neue Ereignisse anzuzeigen. Setzen Sie diese Eigenschaft auf Null, um die Zeitüberschreitung zu deaktivieren.
- Die Eigenschaft *When Empty* (Wenn leer) definiert die Zeichenfolge, die vom Primitiv angezeigt wird, wenn das Protokoll keine Ereignisse enthält. Diese Option wird in der Regel beim Erstellen mehrsprachiger Anwendungen für die Bereitstellung lokalisierter Texte verwendet.

9.11.4.2 Aktivierungseigenschaften

Wenn die Schaltfläche „Clear“ (Löschen) unten im Betrachter über die Registerkarte „Format“ aktiviert ist, wird die Eigenschaft *Enable Clear* (Löschen aktivieren) verwendet, um das Löschen des Ereignisprotokolls zu aktivieren oder zu deaktivieren.

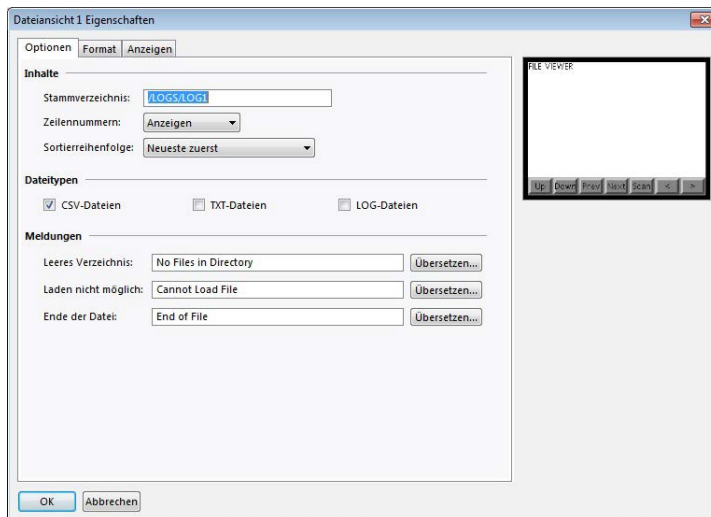
9.11.4.3 Zeiteigenschaften

Die Registerkarte „Time“ (Zeit) gibt das Format an, das bei der Angabe des Zeitpunkts und des Datums verwendet werden soll, zu dem ein Ereignis auftrat. Weitere Einzelheiten finden Sie im Kapitel „Verwenden von Formaten“.

9.11.5 Dateibetrachter

Mit dem Dateibetrachter kann der Benutzer Textdateien auf der Speicherkarte anzeigen.

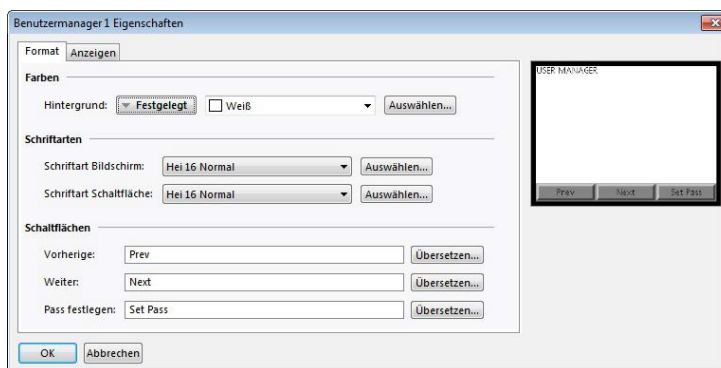
9.11.5.1 Optionseigenschaften



- Die Eigenschaft *Root Directory* (Stammverzeichnis) gibt das Verzeichnis an, das angezeigt werden soll.
- Mit der Eigenschaft *Line Numbers* (Zeilennummern) werden die Zeilennummern in der Datei ein- oder ausgeblendet.
- Mit der Eigenschaft *Sort Order* (Sortierreihenfolge) können Sie angeben, wie auf Dateien zugegriffen werden soll.
- Mit der Eigenschaftsgruppe *File Types* (Dateitypen) wird angegeben, welche Dateitypen für die Anzeige verfügbar gemacht werden sollen. Beachten Sie, dass nur Textdateien angezeigt werden können.
- Mit der Eigenschaftsgruppe *Messages* (Meldungen) werden verschiedene vom Dateibetrachter verwendete Meldungen festgelegt und womöglich auch übersetzt.

9.11.6 Benutzer-Manager

Der Benutzer-Manager wird verwendet, um das Ändern von Kennwörtern zur Laufzeit zu ermöglichen:

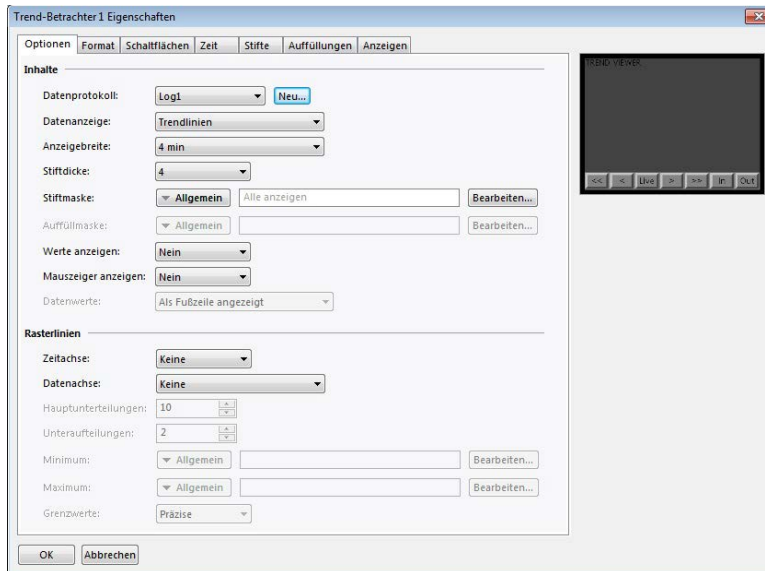


Die Haupteigenschaften befinden sich auf einer einzigen Registerkarte und sind alle konventionell.

9.11.7 Trendbetrachter

Der Trendbetrachter ermöglicht die Anzeige von Informationen aus dem Datenlogger.

9.11.7.1 Optionseigenschaften



- Mit der Eigenschaft *Data Log* (Datenprotokoll) wird das anzuzeigende Datenprotokoll ausgewählt.
- Mit der Eigenschaft *Data Display* (Datenanzeige) wird der Anzeigemodus ausgewählt. Mit „Trend Lines“ (Trendlinien) wird der Protokollinhalt als Linien im Zeitverlauf angezeigt. Mit „Trend Lines with Fill“ (Trendlinien mit Füllung) wird auch der Bereich unterhalb der Linien gefüllt. Die Füllungen werden vor den Trendlinien gezeichnet, d. h. die Linien sind über den gefüllten Bereichen sichtbar.
- Mit der Eigenschaft *View Width* (Ansichtsbreite) wird angegeben, wie viele Daten zu Beginn über das Fenster verteilt angezeigt werden sollen. Der Benutzer kann die Ansicht dann mithilfe der Schaltflächen unten im Betrachter vergrößern und verkleinern.
- Mit der Eigenschaft *Pen Weight* (Stiftstärke) festgelegt, wie dick die vom Betrachter gezeichneten Linien sind. Die Dicke wird in Viertelpixel-Schritten angegeben, wobei die Dicke 4 einem einzelnen Pixel entspricht. Teilbreiten werden mit standardmäßigen Anti-Aliasing-Verfahren implementiert.
- Mit der Eigenschaft *Pen Mask* (Stiftmaske) wird ein 32-Bit-Ganzzahlwert bereitgestellt, um bei bestimmten Kanälen die Trendlinie selektiv zu aktivieren oder zu deaktivieren. Die Option „Bit 0“ entspricht dem ersten Kanal des Datenprotokolls, „Bit 1“ dem zweiten usw. Beim Bitwert Eins wird der Kanal angezeigt, beim Bitwert Null wird er ausgeblendet. Wenn keine Angabe gemacht wird, wird das Standardverhalten verwendet, d. h. alle Kanäle werden angezeigt.
- Mit der Eigenschaft *Fill Mask* (Auffüllmaske) wird ein 32-Bit-Ganzzahlwert bereitgestellt, um die Füllung unter der Linie für bestimmte Kanäle selektiv zu aktivieren oder zu deaktivieren. Die Option „Bit 0“ entspricht dem ersten Kanal des Datenprotokolls, „Bit 1“ dem zweiten usw. Beim Bitwert Eins wird der Kanal angezeigt, beim Bitwert Null wird er ausgeblendet. Wenn keine Angabe gemacht wird, wird das Standardverhalten verwendet, d. h. alle Kanäle werden angezeigt.
- Mit der Eigenschaft *Show Values* (Werte anzeigen) wird die Anzeige der Datenwerte, die mit den einzelnen Kanälen des Datenprotokolls verknüpft sind, entweder im Livemodus oder beim Blättern mit dem Cursor aktiviert bzw. deaktiviert.

- Mit der Eigenschaft *Show Cursor* (Mauszeiger anzeigen) wird die Anzeige eines Cursors im Betrachter aktiviert bzw. deaktiviert. Der Cursor kann vom Benutzer aktiviert werden, damit ein bestimmter Zeitpunkt genau ermittelt werden kann, und um optional die zugehörigen Verläufe der Datenwerte anzuzeigen.
- Mit der Eigenschaft *Time Axis* (Zeitachse) wird festgelegt, ob für die Zeitachse Rasterlinien angezeigt werden sollen. Der Abstand der Rasterlinien wird in Crimson automatisch anhand des vom Betrachter abgedeckten Zeitraums ermittelt.
- Mit der Eigenschaft *Data Axis* (Datenachse) wird die Anzeige von Rasterlinien für die Datenachse gesteuert. Rasterlinien können manuell definiert werden, indem entweder nur die Hauptunterteilungen oder sowohl Haupt- als auch Nebenunterteilungen festgelegt werden. Zur automatischen Berechnung geben Sie die Mindest- und Höchstwerte für die Datenachse ein und lassen Crimson das optimale Rasterlinienmuster berechnen.
- Die Eigenschaften *Major Divisions* (Hauptunterteilungen) und *Minor Divisions* (Nebenunterteilungen) definieren die Anzahl der zu zeichnenden Unterteilungen, wenn manuell definierte Rasterlinien verwendet werden.
- Mit den Eigenschaften *Minimum* und *Maximum* wird der Datenbereich angegeben, der bei Verwendung von automatischen Rasterlinien angezeigt werden soll. Crimson verwendet diese Werte, um das optimale Rasterlinienmuster zu bestimmen. Datenwerte werden ebenfalls auf diese Werte skaliert und nicht auf ihre eigenen Datenlimits.
- Die Eigenschaft *Limit Values* (Grenzwerte) legt fest, wie die oberen und unteren Werte der Skala bestimmt werden. Bei der Einstellung „Precise“ (Präzise) werden die Mindest- und Höchstwerte genau verwendet, selbst wenn dies Grenzwerte ergibt, die nicht genau mit dem automatisch ausgewählten Markierungsabstand übereinstimmen. Dies kann zu unregelmäßigen Abständen zwischen Rasterlinien führen. Bei der Einstellung „Rounded“ (Gerundet) kann das Skala-Primitiv die Grenzwerte anpassen, um Teilstriche in regelmäßigen Abständen zu erzielen.

9.11.7.2 Formateigenschaften

Mit diesen Eigenschaften werden Farben und Schriftarten festgelegt. Sie funktionieren auf konventionelle Weise.

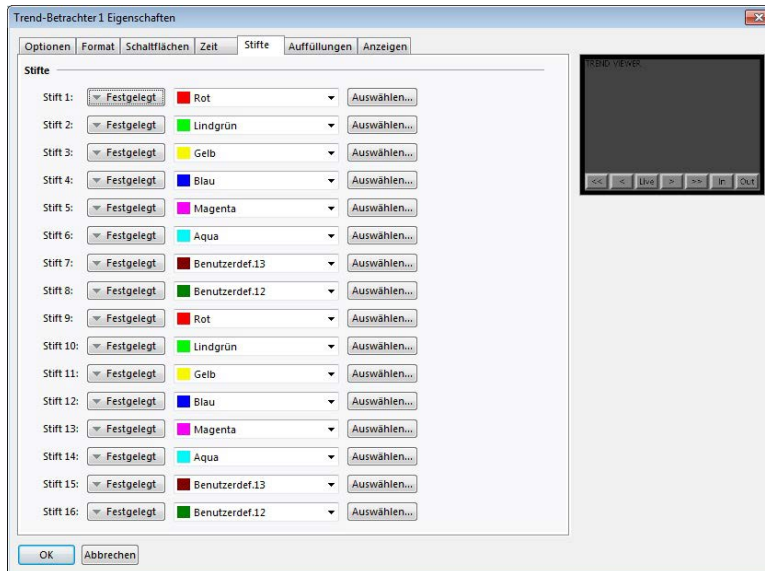
9.11.7.3 Schaltflächeneigenschaften

Diese Eigenschaften dienen zum Bearbeiten sowie optional zum Übersetzen der verschiedenen Schaltflächenbeschriftungen.

9.11.7.4 Zeiteigenschaften

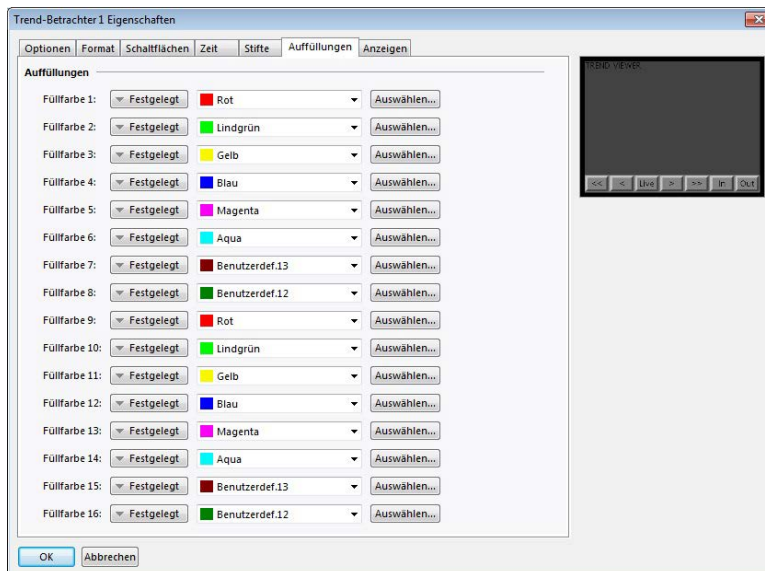
Die Registerkarte „Time“ (Zeit) dient zur Formatierung der Uhrzeit in Zeit- und Datumsangaben im Zusammenhang mit dem Datenprotokoll. Weitere Informationen finden Sie im Kapitel „Verwenden von Formaten“.

9.11.7.5 Stifteigenschaften



Mit diesen Eigenschaften werden 16 Farben angegeben, die für die Zeichnung der Trendlinien verwendet werden. Die Farben werden zyklisch verwendet, d. h. der siebzehnte Kanal hat die gleiche Farbe wie der erste Kanal. Beachten Sie, dass die Zeichnung so vieler Kanäle zu einer verwirrenden Anzeige führen kann.

9.11.7.6 Füllungseigenschaften



Mit diesen Eigenschaften werden 16 Farben angegeben, die für die Zeichnung der Füllung unterhalb der Trendlinien verwendet werden. Die Farben werden zyklisch verwendet, d. h. der siebzehnte Kanal hat die gleiche Farbe wie der erste Kanal. Beachten Sie, dass die Zeichnung so vieler Kanäle zu einer verwirrenden Anzeige führen kann.

9.11.8 Berührungskalibrierung

Das Primitiv „Berührungskalibrierung“ dient zum Kalibrieren des Touchscreens:



Die primitivenspezifischen Eigenschaften definieren die Aktionen, die ausgeführt werden sollen, wenn die Kalibrierung entweder erfolgreich war oder fehlgeschlagen ist. Diese Eigenschaften sind in der Regel so konfiguriert, dass sie zu einem Menübildschirm zurückkehren oder die Kalibrierungsseite verlassen.

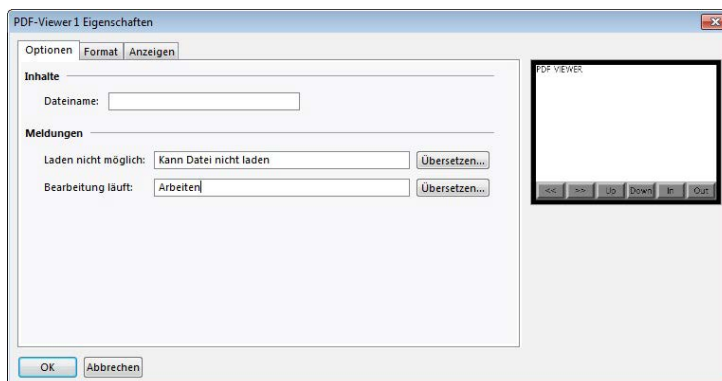
9.11.9 Berührungstester

Mit dem Berührungstester kann der Benutzer die Leistung und Kalibrierung des Touchscreens überprüfen. Bei jeder Berührung wird ein Punkt auf dem Bildschirm erzeugt, und die letzten Berührungen werden als Spur angezeigt. Abgesehen von der Sichtbarkeitssteuerung gibt es keine konfigurierbaren Eigenschaften.

9.11.10 PDF-Viewer

Mit dem Primitiv „PDF-Viewer“ können Sie eine PDF-Datei anzeigen, die auf der Speicherkarte des Geräts gespeichert ist.

9.11.10.1 Optionseigenschaften

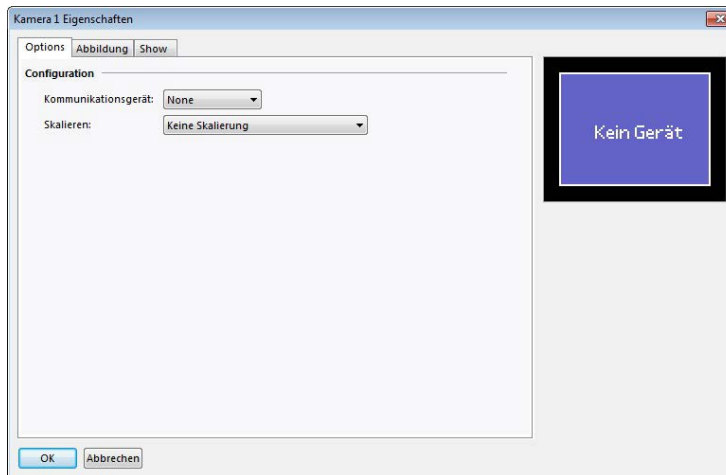


- Mit der Eigenschaft *File Name* (Dateiname) wird angegeben, welche PDF-Datei angezeigt werden soll. Die PDF-Datei muss auf der Speicherkarte des Geräts in einem Verzeichnis namens `pdf` gespeichert sein.
- Die Eigenschaft *Cannot Load* (Laden nicht möglich) gibt an, welcher Text angezeigt wird, wenn die benannte Datei nicht geladen werden kann, weil sie nicht vorhanden ist bzw. Fehler enthält.
- Die Eigenschaft *Working* (Wird bearbeitet) definiert den Text, der angezeigt wird, während eine Datei geladen und für die Anzeige verarbeitet wird.

9.11.11 Kamera

Das Kamera-Primitiv wird verwendet, um die Anzeige von Videodaten von einer externen Kamera auf einer Seite zu ermöglichen.

9.11.11.1 Optionseigenschaften



- Unter *Comms Device* (Kommunikationsgerät) wird die Quelle des Videos festgelegt, das im Primitiv angezeigt werden soll. Eine Auswahl ist hier erst möglich, nachdem ein Gerät einem Kommunikationsanschluss zugeordnet wurde.
- Mit *Scale* (Skalieren) wird die von der Quelle bereitgestellte Bildgröße geändert. Die Skalierung ist treiberabhängig und wird möglicherweise nicht von allen Kamerateypen unterstützt.

9.11.11.2 Figureigenschaften

Die Eigenschaften auf der Registerkarte „Figure“ (Figur) dienen der Steuerung des Erscheinungsbilds des Primitivs und sind konventionell.

9.12 Ältere Primitive

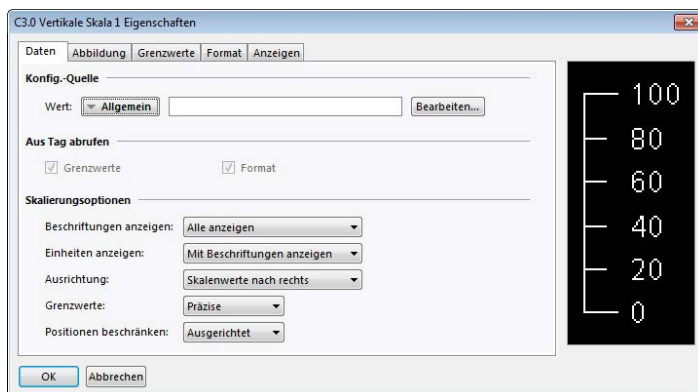
Diese Primitive werden zwecks Kompatibilität mit Crimson 2.0 und Crimson 3.0 bereitgestellt. Die meisten Primitive sind analog zu den oben beschriebenen neueren Primitive. Allerdings verwenden sie nicht die erweiterte Grafikmaschine von Crimson 3.1, um ein glatteres Erscheinungsbild zu erzielen, und sie unterstützen keine erweiterten Füll- und Randeigenschaften. Mit Ausnahme des Skala-Primitivs für vertikale Skalen werden sie hier nicht näher beschrieben. Wenn das Verhalten eines Primitivs nicht intuitiv oder konventionell ist, schlagen Sie in früheren Softwarehandbüchern nach.

9.12.1 Skala-Primitiv



Mit dem Skala-Primitiv wird eine vertikale Skala gezeichnet. Die Grenzwerte der Skala können als Konstante definiert werden oder abhängig vom Wert bestimmter Ausdrücke variieren. Die Skala kann beschriftet oder unbeschriftet sein. Etwaige Beschriftungen basieren auf einem festgelegten Formatobjekt, das optional aus einem Tag abgerufen werden kann. Die Skala funktioniert wie das Streudiagramm, das bereits in diesem Kapitel behandelt wurde, und kann daher zum Beschriften eines derartigen Diagramms verwendet werden.

9.12.1.1 Dateneigenschaften

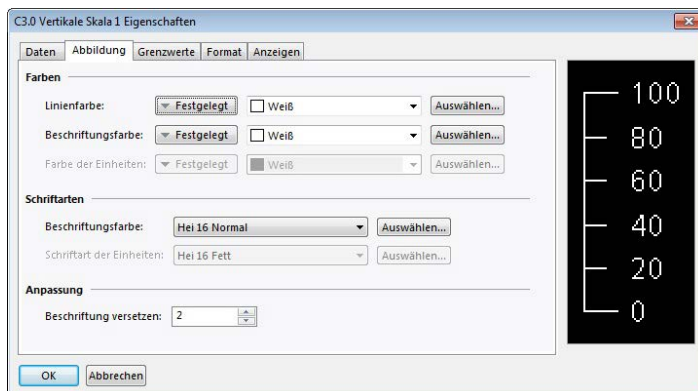


- Die Eigenschaft *Value* (Wert) definiert ein optionales Tag, mit dem Grenzwerte und Formatinformationen für die Skala abgerufen werden. Der Wert selbst wird nicht vom Primitiv verwendet, doch das Tag dient als Quelle für weitere Informationen.
- Mit den Eigenschaften *Get from Tag* (Aus Tag abrufen) wird angegeben, ob ein in der Eigenschaft „Value“ (Wert) ausgewähltes Tag als Quelle für die betreffenden Daten verwendet werden soll.
- Mit der Eigenschaft *Show Labels* (Bezeichnungen anzeigen) werden die numerischen Skalenbeschriftungen ein- bzw. ausgeblendet.
- Mit der Eigenschaft *Show Units* (Einheiten anzeigen) werden die von einem numerischen Datenformat definierten Einheiten ein- bzw. ausgeblendet. Die Einheiten können an jede Skalenbeschriftung angehängt oder vertikal am Rand der Skala gezeichnet werden.
- Die Eigenschaft *Limit Values* (Grenzwerte) legt fest, wie die oberen und unteren Werte der Skala bestimmt werden. Bei der Einstellung „Precise“ (Präzise) werden die Grenzwerte genau verwendet, selbst wenn dies Grenzwerte ergibt, die nicht genau mit dem automatisch ausgewählten Markierungsabstand übereinstimmen. Dies kann zu unregelmäßigen Skalenbeschriftungen führen, gewährleistet

jedoch, dass eine Behälterfüllung, die neben der Skala platziert wird und an das gleiche Tag gebunden ist, genau wie erforderlich gezeichnet wird. Bei der Einstellung „Rounded“ (Gerundet) kann das Skala-Primitiv die Grenzwerte automatisch anpassen, um Teilstriche in regelmäßigeren Abständen zu erzielen.

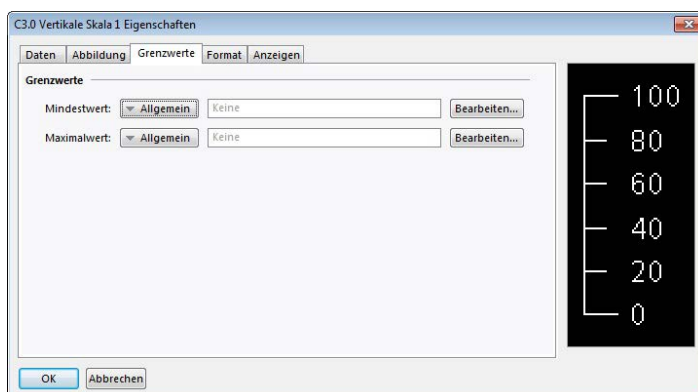
- Die Eigenschaft *Limit Positions* (Grenzwertpositionen) gibt das Verhältnis zwischen den Grenzwerten der Skala und den Einheitenbezeichnungen an. Mit der Einstellung „Aligned“ (Ausgerichtet) werden die Teilstriche und die Beschriftungen präzise ausgerichtet, wobei die äußeren Teilstriche vom Rand des Primitivs nach innen verschoben werden. Mit der Einstellung „Shifted“ (Versetzt) werden die beiden äußeren Beschriftungen in Bezug zu den Teilstrichen verschoben. Die Teilstriche für die Mindest- und Höchstwerte liegen dabei auf den Kanten des Primitivs, was etwa die Ausrichtung mit einer Behälterfüllung erleichtert.

9.12.1.2 Figureigenschaften



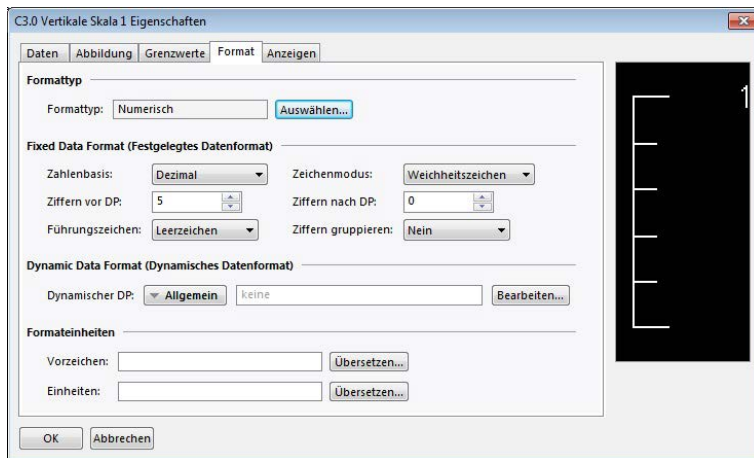
Die Eigenschaften auf dieser Seite definieren die Farben und Schriftarten für die Skala. Details zu den Standardeigenschaften finden Sie im vorherigen Kapitel. Mit der Eigenschaft *Label Shift* (Beschriftungsversetzung) können die Beschriftungen im Verhältnis zu den Teilstrichen nach oben oder unten verschoben werden. Hierdurch lassen sich bei der Arbeit mit Schriftarten mit Freiraum über oder unter den Schriftzeichen attraktivere Ergebnisse erzielen.

9.12.1.3 Grenzwerteigenschaften



Die Eigenschaften auf dieser Seite dienen zum Festlegen der in der Skala angezeigten Mindest- und Höchstwerte. Sie können Ausdrücke angeben. In diesem Fall wird die Skala zur Laufzeit von Crimson dynamisch aktualisiert, wobei Teilstriche und Beschriftungspositionen gewählt werden, die für die neuen Werte geeignet sind. Diese Einstellungen sind nicht verfügbar, wenn ein Tag als Quelle der Grenzwerte ausgewählt wurde.

9.12.1.4 Formateigenschaften



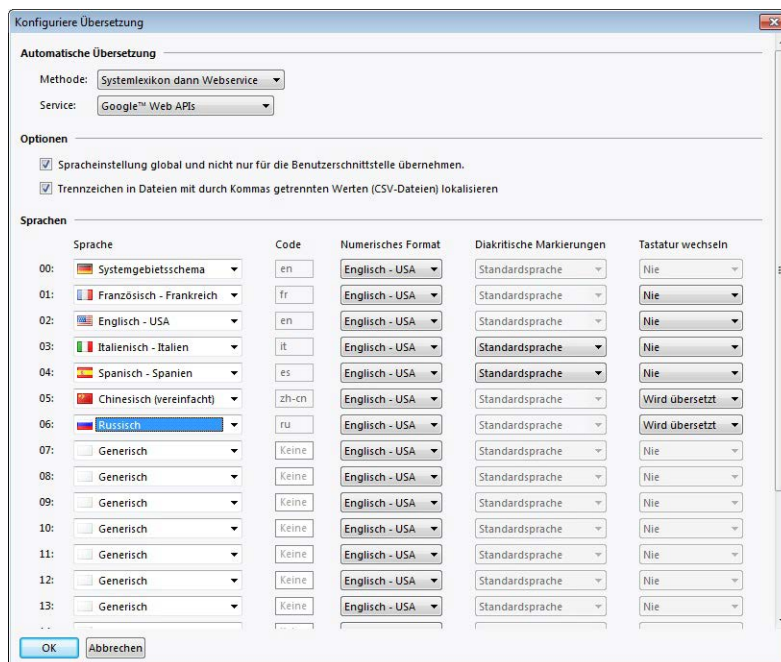
Das Feld *Format Type* (Formattyp) gibt den Formattyp an, der beim Zeichnen der Beschriftungen verwendet werden soll. Nur allgemeine oder numerische Formate werden unterstützt. Die Auswahl ist möglicherweise nicht verfügbar, wenn das Format von einem Tag abgerufen wird. Details zu den verschiedenen weiteren Eigenschaften, die bei der Auswahl eines numerischen Formats angezeigt werden, enthält der Abschnitt „Verwenden von Formaten“.

Kapitel 10 Lokalisierung

Crimson unterstützt mehrere Funktionen, mit deren Hilfe Sie Ihre Datenbank für die Bereitstellung in mehrsprachigen Umgebungen anpassen können. In diesem Kapitel wird beschrieben, wie diese Funktionen verwendet werden und wie Sie mühelos Datenbanken erstellen, die auf der ganzen Welt eingesetzt werden können.

10.1 Auswählen von Sprachen

Die erste Phase bei der Erstellung einer mehrsprachigen Datenbank besteht darin, die in Ihrem Projekt zu verwendenden Sprachen zu konfigurieren. Über die Schaltfläche „Configure Translation“ (Übersetzung konfigurieren) auf der Seite „Global“ (Allgemein) der Eigenschaften für die Benutzeroberfläche wird das folgende Dialogfeld angezeigt:



Im oberen Bereich werden die globalen Einstellungen für den Übersetzungsprozess festgelegt:

- Die Eigenschaft *Method* (Methode) gibt an, wie die Übersetzung durchgeführt werden soll. Hierauf wird im folgenden Abschnitt näher eingegangen.
- Die Eigenschaft *Service* (Dienst) wählt Sie einen von mehreren unterstützten Übersetzungsdiensten aus, den Crimson bei der automatischen Übersetzung nutzen soll.
- Die Eigenschaft *Apply language setting globally and not just to user interface* (Spracheinstellung global und nicht nur für die Benutzeroberfläche übernehmen) weist Crimson an, die aktuell ausgewählte Sprache auf alle Vorgänge anzuwenden, die übersetzbaren Text verwenden. Dazu ein Beispiel. Angenommen, es gibt ein Mehrfachstatus-Tag mit übersetzten Statusnamen, das der Datenprotokollierung unterliegt. Wenn diese Eigenschaft aktiviert ist, wird der Status des Tags im Protokoll in der aktuell ausgewählten Sprache aufgezeichnet. Wenn die Eigenschaft deaktiviert ist, wird die Standardsprache verwendet, unabhängig davon, welche Benutzeroberflächensprache ausgewählt ist.
- Die Eigenschaft *Localize separators in Comma Separated Variable (CSV) files* (Trennzeichen in Dateien mit durch Kommas getrennten Werten (CSV-Dateien) lokalisieren) weist Crimson an, das

Komma in CSV-Dateien durch eine lokalisierte Version zu ersetzen. Dies ist bei Sprachen, die das Komma als Dezimaltrennzeichen verwenden, in der Regel ein Semikolon.

Im unteren Bereich des Dialogfelds werden für jede Sprache verschiedene Eigenschaften definiert:

- Mit der Eigenschaft *Language* (Sprache) wird die erforderliche Sprache ausgewählt. Für eine Sprache gibt es möglicherweise landesspezifische Varianten. Die Einstellung „Generic“ (Generisch) kann für Sprachen verwendet werden, die von Crimson nicht direkt unterstützt werden.
- Die Eigenschaft *Code* dient zum Anzeigen oder Eingeben des zweistelligen Codes für die ausgewählte Sprache. Diese Eigenschaft wird während der automatischen Übersetzung an die Webübersetzungsdienste übergeben und zur Definition der Kopfzeile in einer Lexikondatei verwendet. Sie müssen den Code für generische Sprachen, die Crimson unbekannt sind, manuell eingeben.
- Mit der Eigenschaft *Numeric Format* (Numerisches Format) können Sie festlegen, ob in Crimson Zahlen im US-Format formatiert werden oder in einem Format, das in der aktuell gewählten Sprache üblich ist. Die numerischen Formatierungsoptionen umfassen die Verwendung von Kommas anstelle von Dezimalpunkten sowie die Platzierung von Zahlengruppierungszeichen.
- Mit der Eigenschaft *Diacritical Marks* (Diakritische Zeichen) wird die Standardeinstellung einer Sprache für die Behandlung von Akzenten bei Großbuchstaben außer Kraft gesetzt. In dem in Frankreich gesprochenen Französisch (im Gegensatz zu kanadischem Französisch) werden beispielsweise bei Großbuchstaben Akzente verwendet. Die Wiedergabe dieser Zeichen ist bei einigen Schriftarten schwieriger. Wenn Sie für diese Einstellung „Lower Case Only“ (Nur Kleinbuchstaben) auswählen, wird dieses Standardverhalten außer Kraft gesetzt.
- Mit der Eigenschaft *Switch Keyboard* (Tastatur wechseln) wird angegeben, unter welchen Umständen die Crimson Konfigurationssoftware das Tastaturlayout auf das von der Sprache verwendete Layout umschalten soll. Die Umschaltung kann erfolgen, wenn das Dialogfeld „Translation“ (Übersetzung) verwendet wird, wann immer Text in dieser Sprache bearbeitet wird, oder auch gar nicht. Die Tastaturumschaltung im Dialogfeld „Translation“ (Übersetzung) ist für Sprachen wie vereinfachtes Chinesisch standardmäßig aktiviert. Dadurch wird gewährleistet, dass der richtige Input Method Editor (Eingabemethoden-Editor) aufgerufen wird.

10.2 Konfigurieren der automatischen Übersetzung

Crimson enthält leistungsstarke Funktionen für die automatische Übersetzung, die Ihnen helfen, Ihre Datenbank für die internationale Bereitstellung anzupassen. Der Übersetzungsprozess stützt sich auf zwei Komponenten, nämlich auf ein Systemlexikon und einen webbasierten Übersetzungsdienst.

Das Systemlexikon ist eine Unicode-Textdatei mit vielen Standardwörtern und -sätzen, die in der industriellen Automatisierung und Prozesssteuerung häufig verwendet werden, sowie Übersetzungen in einer Reihe gängiger Sprachen. Dieses Lexikon kann während des Übersetzungsprozesses aufgerufen werden, sodass einige Texte sehr schnell und mit hoher Genauigkeit übersetzt werden können.

Der webbasierte Teil verwendet einen von zwei Diensten. Google WebAPIs bieten in der Regel schnellere Übersetzungen, da sie keinen Bandbreitenbeschränkungen unterliegen. Im Gegensatz dazu bietet Microsoft Translator etwas genauere Übersetzungen, ist jedoch langsamer, weil pro Minute nur eine bestimmte Anzahl von Anfragen möglich ist. Im oben abgebildeten Dialogfeld „Configure Translation“ (Übersetzung konfigurieren) können Sie einen der beiden Dienste auswählen.

Die automatische Übersetzung kann so konfiguriert werden, dass eine der Methoden oder beide verwendet werden. Wenn Sie über eine Internetverbindung verfügen, ist es im Allgemeinen besser, zuerst das Lexikon und dann den webbasierten Dienst zu verwenden. Das Lexikon kann in bestimmten Situationen allein verwendet werden, um die fragwürdigen Übersetzungen zu vermeiden, die die webbasierten Dienste mitunter bereitstellen.

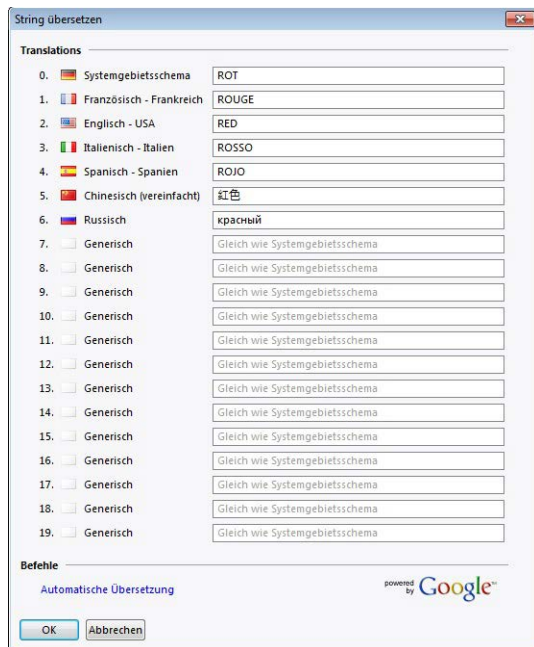
Beachten Sie, dass Textfelder nicht mehr als 1.000 Zeichen akzeptieren. Wenn eine englische Zeichenfolge nah an dieser Grenze liegt, sind die Übersetzungen möglicherweise zu lang und werden nicht von Crimson akzeptiert. Lange Übersetzungen sollten nach der automatischen Übersetzung überprüft werden, um sicherzugehen, dass sie richtig sind.

10.3 Übersetzen der Datenbank

Die Datenbankübersetzung kann auf verschiedene Weise erfolgen, wie im restlichen Teil dieses Abschnitts beschrieben.

10.3.1 Eingeben von Übersetzungen

Bei der manuellen Übersetzung klicken Sie neben jeder zu übersetzenden Zeichenfolge in der Datenbank auf die Schaltfläche „Translate“ (Übersetzen). Daraufhin wird ein Dialogfeld angezeigt, in das entweder der übersetzte Text eingegeben oder in dem die automatische Übersetzung nur für diese Zeichenfolge aufgerufen werden kann:



Bei dieser gezielten Nutzung der automatischen Übersetzung können Sie die Richtigkeit der Übersetzungen prüfen.

10.3.2 Konfigurieren der automatischen Übersetzung

Das Untermenü „Utilities“ (Werkzeuge) im Menü „File“ (Datei) enthält einen Befehl, mit dem die automatische Übersetzung auf alle Zeichenfolgen in der Datenbank angewendet wird. Die Ausführung dieses Befehls kann einige Zeit in Anspruch nehmen, insbesondere wenn ein Übersetzungsdienst mit begrenzter Bandbreite verwendet wird. Bei der Verwendung der globalen automatischen Übersetzung müssen Sie besondere Vorsicht walten lassen. Zeichenfolgen, die nicht im Systemlexikon enthalten sind, werden möglicherweise falsch übersetzt, wenn sie Fachbegriffe oder branchenspezifischen Jargon enthalten.

10.3.3 Exportieren und Importieren

Das Untermenü „Utilities“ (Werkzeuge) enthält auch Befehle zum Exportieren und Importieren von Textdateien, die den gesamten übersetzbaren Text der Datenbank enthalten. Diese Dateien können in einer Anwendung wie Microsoft Excel bearbeitet werden, wobei die Übersetzungen manuell eingegeben werden. Diese Funktion ist bei der Arbeit mit einem externen Übersetzungsdienst besonders nützlich. Das Dateiformat enthält mehrere Spalten, die die Quelle der einzelnen Zeichenfolgen angeben, damit eine bestimmte, mehrmals vorkommende Zeichenfolge je nach Kontext anders übersetzt werden kann.

10.3.4 Anwenden eines Lexikons

Neben dem oben beschriebenen Systemlexikon können Sie auch eigene Lexika erstellen, und zwar entweder von Grund auf neu oder durch Verwenden des Befehls „Export Lexicon“ (Lexikon exportieren) im Untermenü „Utilities“ (Werkzeuge). Lexikondateien sind Unicode-Textdateien, die mit einer Kopfzeile beginnen. Die Kopfzeile enthält tabulatorgetrennte Sprachcodes, wie im Dialogfeld „Configure Translation“ (Übersetzung konfigurieren) unter der Eigenschaft „Code“ angegeben. Auf die Kopfzeile folgen Zeilen mit Text. Jede Zeile enthält ein Wort oder einen Satz in jeder der definierten Sprachen.

Nachstehend ist ein Beispiel für eine Lexikondatei abgebildet:

en	fr	de
one	un	eins
two	deux	zwei
three	trois	drei

Beachten Sie, dass Text in Kleinbuchstaben eingegeben werden sollte, es sei denn, ein bestimmter Begriff beginnt immer mit einem Großbuchstaben, wie z. B. Substantive in Deutsch. Die Verwendung von Kleinbuchstaben ermöglicht Crimson die Bildung eigener Varianten in Groß- und Kleinschreibung.

10.4 Vorschau von Übersetzungen

Eine Vorschau von Übersetzungen kann im Grafikeditor angezeigt werden. Dazu wählen Sie die die entsprechende Sprache aus dem Dropdown-Menü, das über das Flaggensymbol in der Symbolleiste aufgerufen wird. Jede direkte Textbearbeitung wird auch auf die aktuell ausgewählte Sprache angewendet. Die anderen Sprachen bleiben unverändert. Die Bearbeitung innerhalb von Dialogfeldern bleibt weiterhin auf die Standardsprache beschränkt. Der Zugriff auf die anderen Sprachen erfolgt über die Schaltfläche „Translate“ (Übersetzen).

10.5 Wechseln zwischen Sprachen

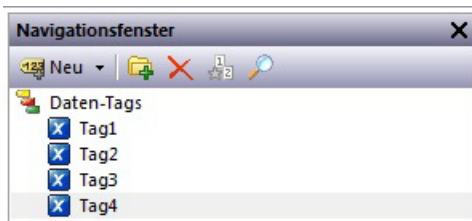
Die vom Zielgerät verwendete Sprache wird über Aufrufe an die Funktion `SetLanguage()` gesteuert. Das Argument der Funktion ist eine Zahl zwischen 0 und 19 und dient zur Auswahl der gewünschten Option. Durch den Aufruf von `SetLanguage(1)` im obigen Beispiel wird etwa Französisch ausgewählt. Mit der benutzerdefinierten Aktion `SetLanguage(2)` hingegen wird Deutsch ausgewählt. Die Funktion `GetLanguage()` kann zur Bestimmung der aktuellen Sprache verwendet werden.

Kapitel 11 Verwenden von Widgets

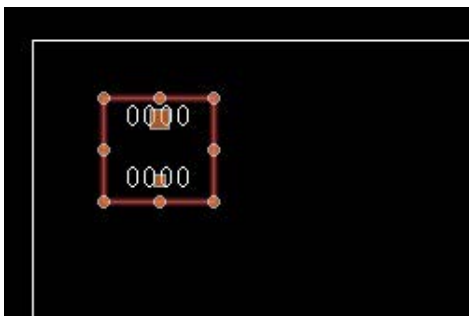
Crimson® 3.1 unterstützt eine leistungsstarke neue Funktion, die es ermöglicht, einfache Gruppen von Primitiven in leistungsstarke Gebilde namens Widgets zu verwandeln. Zusätzlich zu seinen grundlegenden Primitiven enthält ein Widget benutzerdefinierbare Datenelemente, die auf Gruppenebene bearbeitet, aber über die Komponenten des Widgets referenziert werden können. In diesem Kapitel wird erläutert, wie Sie Widgets erstellen und einsetzen.

11.1 Erstellen eines Widgets

Was genau ein Widget ist und was es leistet, können Sie am besten nachvollziehen, indem Sie eines erstellen. Dazu erstellen Sie zunächst eine leere Datenbank und fügen ihr vier numerische Tags hinzu. Übernehmen Sie die Standardeinstellungen der Tag-Eigenschaften. Dadurch erhalten Sie vier interne ganzzahlige Werte namens Tag1 bis Tag4.



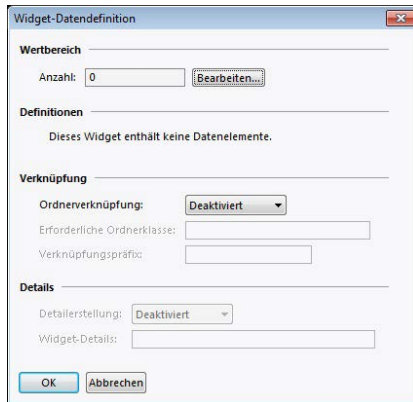
Wechseln Sie zum Abschnitt „Display Pages“ (Anzeigeseiten) und fügen Sie der Seite zwei Datenfeld-Primitive hinzu:



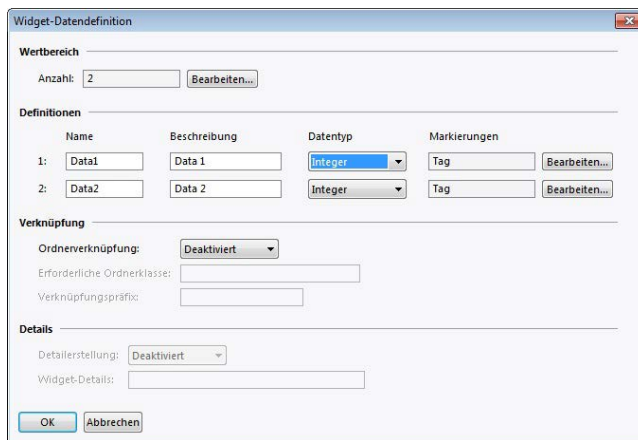
Übernehmen Sie vorerst die für die Eigenschaften angezeigten Standardwerte und wählen Sie beide Elemente aus. Klicken Sie mit der rechten Maustaste auf die Auswahl und wählen Sie im Kontextmenü den Befehl „Widgetize“ (Zum Widget machen) aus. Die Elemente werden zu einer Gruppe zusammengefasst. Außerdem wird folgendes Dialogfeld angezeigt:



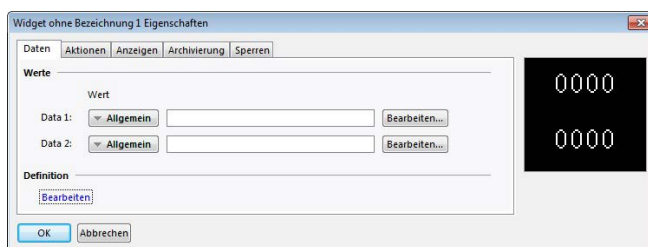
Nachdem das Widget erstellt wurde, wird dieses Dialogfeld zur Bearbeitung der Datenelemente des Widgets verwendet. Vorläufig ist jedoch noch nichts definiert. Klicken Sie im Abschnitt „Definition“ auf die Schaltfläche „Edit“ (Bearbeiten), um einige Datenelemente zu definieren:



Durch Klicken auf die Schaltfläche „Edit“ (Bearbeiten) neben dem Feld „Count“ (Anzahl) können Sie zwei Eigenschaften erstellen:

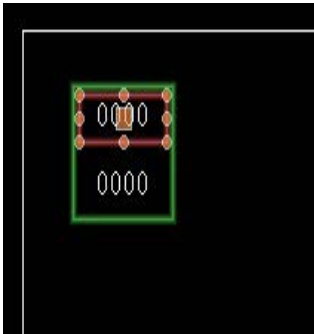


Füllen Sie die Datenfelder wie oben gezeigt aus und achten Sie darauf, den richtigen Datentyp zu wählen. Ändern Sie die Felder unter „Flags“ (Markierungen) so, dass für beide Datenelemente „Tag“ angezeigt wird. (Das Feld „Flags“ [Markierungen] kann mit der Schaltfläche „Edit“ [Bearbeiten] geändert werden, die sich neben der Eigenschaft befindet.) Klicken Sie auf „OK“, um das Dialogfeld zu schließen. Wie Sie sehen, zeigt das Widget nun Datenelemente in seinem eigenen Eigenschaften-Dialogfeld an:

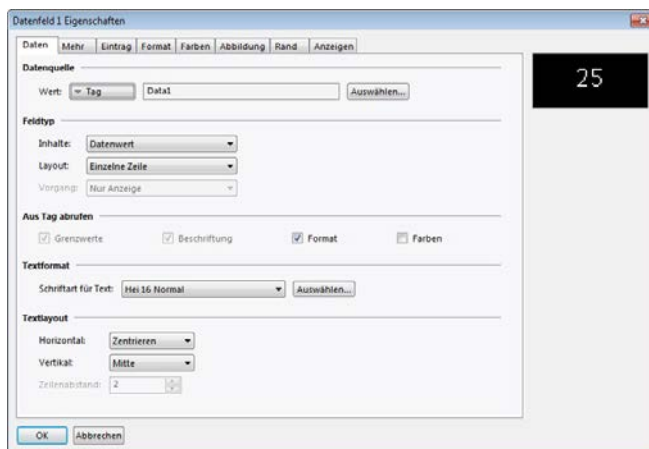


Ignorieren Sie diese vorerst und klicken Sie auf „OK“, um auch dieses Dialogfeld zu schließen.

Das Widget müsste weiterhin im Grafikeditor ausgewählt sein. Klicken Sie auf eines der im Widget enthaltenen Datenfelder, um in den Gruppenbearbeitungsmodus zu wechseln. Denken Sie daran, dass das grüne Rechteck die Gruppe kennzeichnet, die bearbeitet wird. Das rote Rechteck kennzeichnet das in dieser Gruppe ausgewählte Element:



Doppelklicken Sie auf das Datenfeld, um die zugehörigen Eigenschaften anzuzeigen:

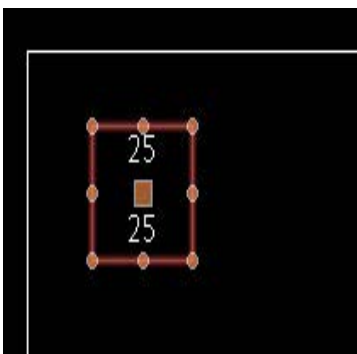


Geben in das Feld „Value“ (Wert) `Data1` ein, und sehen Sie sich das Ergebnis an.

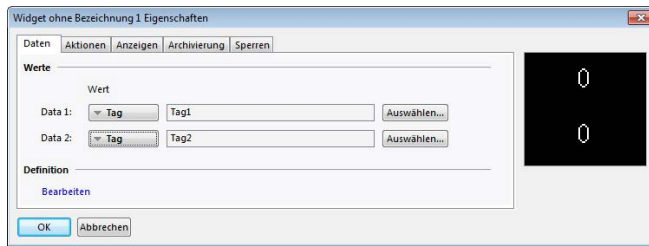
Crimson akzeptiert dies als Tag-Name, obwohl die Datenbank kein Tag namens `Data1` enthält. Tatsächlich stimmt dieser Wert mit einem der im Widget definierten Datenelemente überein und wird das Tag darstellen, das wir zuweisen, wenn wir zurückgehen und die Widget-Daten bearbeiten. (Der im Vorschaufenster angezeigte Wert von 25 ist der Standardwert, der für Widget-Datenelemente verwendet wird, die nicht zugeordnet sind.) Da „`Data1`“ als Tag gekennzeichnet sind, können wir auf dessen Eigenschaften zugreifen, es als Formatierungsquelle verwenden oder andere bei Tags übliche Vorgänge ausführen.

Wiederholen Sie diesen Schritt für das zweite Datenfeld. Geben Sie unter „Value“ (Wert) diesmal `Data2` ein.

Drücken Sie die **ESC**-Taste, bis nur das Widget ausgewählt ist. Wenn Sie zu weit gehen und die Auswahl aufheben, klicken Sie einfach auf das Widget selbst. Achten Sie dabei darauf, dass es von einem roten Rechteck umgeben ist:



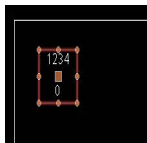
Rufen Sie jetzt die Widget-Eigenschaften auf und geben Sie diesmal Werte für die Datenelemente ein:



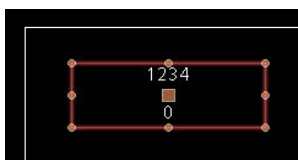
Geben Sie die oben angezeigten Werte ein und setzen Sie die Datenelemente auf Tag1 bzw. Tag2. Wie Sie sehen, werden in der Vorschau jetzt zwei Nullen angezeigt. Dies liegt daran, dass die Datenfelder im Widget ihre Daten jetzt aus Tag1 und Tag2 abrufen. Um die Angelegenheit etwas interessanter zu gestalten, klicken Sie mit der rechten Maustaste auf das Widget und öffnen Sie das Menü „Jump To“ (Springen zu).



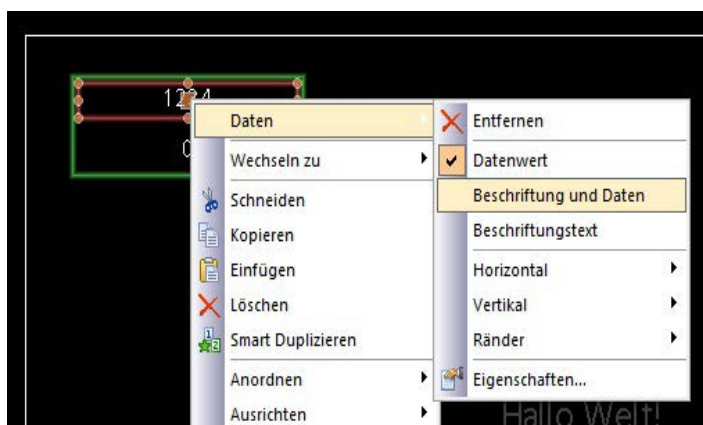
Wählen Sie „Tag1“, um zu diesem Tag zu springen, und geben Sie für die Eigenschaft „Simulate As“ (Simulieren als) den Wert 1234 ein. Drücken Sie **ALT+NACH LINKS** oder die Zurück-Taste auf der Symbolleiste und beobachten Sie, wie das Widget die Tag-Daten weiterhin verfolgt:



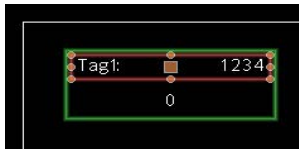
Ziehen Sie dann den rechten Griff, um das Widget etwas breiter zu machen:



Klicken Sie mit der linken Maustaste auf das obere Datenfeld, um den Gruppenbearbeitungsmodus aufzurufen. Klicken Sie dann mit der rechten Maustaste auf dasselbe Feld, um dessen Kontextmenü aufzurufen:



Öffnen Sie das Untermenü „Data“ (Daten) und wählen Sie den Befehl „Label and Data“ (Bezeichnung und Daten) aus, um dieses Datenfeld so zu konfigurieren, dass die Bezeichnung des Tags und dessen Wert angezeigt werden. Sehen Sie sich an, wie das Widget jetzt aussieht:



Wie Sie sehen, zeigt das Datenfeld die Bezeichnung aus Tag1 an. Dies bedeutet, dass der Wert von `Data1`, den wir in die Eigenschaft „Value“ (Wert) des Datenfelds eingegeben haben, vollständig mit dem Tag übereinstimmt, mit dem das Datenelement anschließend konfiguriert wurde. Wir bezeichnen die Einstellung eines Widget-Elements auf ein Tag als Bindung dieses Datenelements an dieses Tag. Die Bindung kann auf komplexere Weise erfolgen, wie noch später erörtert wird.

11.1.1 Zusammenfassung

Wir haben folgende Schritte ausgeführt:

- Wir platzierten Primitive in der Anzeige und erstellten daraus eine spezielle Gruppe namens „Widget“. Das Widget schien sich in Hinblick auf die Bearbeitung usw. wie eine normale Gruppe zu verhalten, hatte aber zusätzliche Eigenschaften.
- Wir bearbeiteten die Datendefinitionen für das Widget und erstellten zwei Datenelemente. Jedes Element erhielt einen Namen, eine Beschreibung, einen Datentyp und eine Reihe von Markierungen.
- In der Gruppenbearbeitung bearbeiteten wir die Inhalte des Widgets, stellten deren Eigenschaften auf die Datenelemente des Widgets ein und referenzierten sie anhand der Datenelement-Namen.
- Wir änderten die Datenelemente des Widgets, banden sie an Tags und stellten so den Inhalten unseres Widgets echte Tags und die zugehörigen Informationen bereit.

11.1.2 Warum dies wichtig ist

Warum also sind Widgets wichtig? Wir hätten auch einfach die Datenfelder erstellen und direkt an die Tags binden können. Warum also sollten Sie sich mehr Arbeit machen? Die Frage beantwortet sich von selbst, wenn Sie versuchen, komplexere Widgets zu erstellen:

- Mit Widgets können Datenelemente an mehreren Orten verwendet werden, und mehrere Elemente im Widget sind von demselben Tag abhängig, ohne dass Sie den Tag-Namen an mehreren Orten auswählen müssen.
- Widgets können komplexe Designs und Funktionen umfassen und ermöglichen Ihnen, diese zu replizieren und in derselben Datenbank oder in anderen Datenbanken wiederzuverwenden. Sie ermöglichen dem Benutzer, komplexe Primitive zu erstellen.
- Widgets können auf der Festplatte gespeichert und dem Ressourcenfenster hinzugefügt oder per E-Mail verteilt werden, wodurch die Zusammenarbeit zwischen Crimson-Benutzern bzw. zwischen Benutzern und technischem Support erleichtert wird.

11.1.3 Einzelheiten

Im nächsten Abschnitt werden die meisten der oben genannten Themen erneut aufgegriffen und ausführlicher erläutert.

Außerdem erfahren Sie, wie Sie Widgets noch leistungsstärker machen können.

11.2 Datendefinitionen für Widgets

Die Merkmale, auf denen die Leistung von Widgets beruht, sind die Datenelemente. Zum Bearbeiten der Datendefinition eines Widgets öffnen Sie die Eigenschaften des Widgets und klicken auf der Seite „Data“ (Daten) unter „Definitionen“ (Definitionen) auf die Schaltfläche „Edit“ (Bearbeiten):

Name	Beschreibung	Datentyp	Markierungen
1: Data1	Data 1	Integer	Tag
2: Data2	Data 2	Integer	Tag

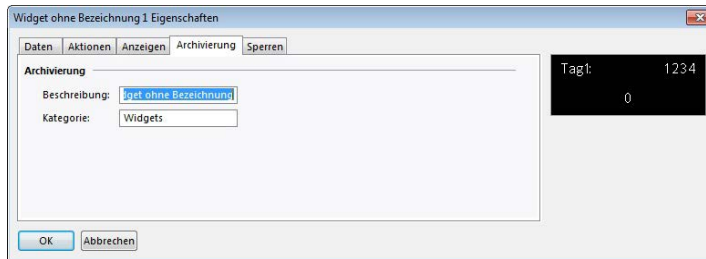
- Die Eigenschaft *Extent* (Umfang) gibt an, wie viele Datenelemente für dieses Widget erforderlich sind. Dieser Wert kann jederzeit geändert werden. Wenn Sie ihn verringern, gehen jedoch Datenelemente und deren Werte verloren. Bis zu 80 Datenelemente können definiert werden.
- Die Eigenschaft *Name* jedes Datenelements wird verwendet, um von den im Widget enthaltenen Primitiven auf das Element zu verweisen. Sie muss daher alle Anforderungen eines Tag-Namens erfüllen. Sie darf keine Leerzeichen oder Satzzeichen enthalten und muss mit einem Buchstaben beginnen.
- Die Eigenschaft *Description* (Beschreibung) für jedes Datenelements wird dazu verwendet, eine benutzerfreundlichere Version des Namens bereitzustellen, die im Dialogfeld für die Bearbeitung des Datenelements angezeigt wird. Für den Inhalt dieses Felds gibt es keine Einschränkungen.
- Die Eigenschaft *Data Type* (Datentyp) für die einzelnen Elemente definiert den erforderlichen Datentyp. Wie das Datenelement im Eigenschaften-Dialogfeld des Widgets angezeigt wird, richtet sich nach der hier ausgewählten Einstellung. Die Datentypen „Real“, „Integer“ (Ganzzahl) und „String“ /Zeichenfolge) entsprechen Ausdruckswerten. Die Datentypen „Color“ (Farbe), „Page“ (Seite) und „Action“ (Aktion) ermöglichen das Erstellen komplexerer Elemente. Seiten- und Aktionselemente können von den Primitiven des Widgets aus als Anzeigeseitenamen und Programme behandelt werden.
- Die Eigenschaft *Flags* (Markierungen) für die einzelnen Datenelemente wird zum Ändern von Elementen mit den Datentypen „Real“, „Integer“ oder „String“ verwendet. Sie unterstützt die folgenden Einstellungen:

Einstellung	Beschreibung
Tag	Der für das Datenelement eingegebene Wert muss ein Tag sein. Die Primitive im Widget können das Datenelement als Tag behandeln und auf dessen Eigenschaften, Datenformat usw. zugreifen.
Beschreibbar	Der für das Datenelement eingegebene Wert muss beschreibbar sein. Die Primitive im Widget dürfen gleichermaßen in das Datenelement schreiben.
Array	Der für das Datenelement eingegebene Wert muss der Name eines Arrays sein. Die Primitive im Widget betrachten das Datenelement als Array und müssen für den Zugriff auf einzelne Werte den Indexoperator verwenden.
Element	Der für das Datenelement eingegebene Wert muss ein Array-Element sein. Die Primitive im Widget betrachten das Datenelement als Element und können es an Funktionen übergeben, die Argumente dieses Typs erfordern.
Keine Bindung	Crimson wendet keine Ordnerbindung auf diese Eigenschaft an. Sie kann somit zur Speicherung vordefinierter Werte verwendet werden, ohne dass beim Binden Fehler generiert werden. Weitere Informationen zur Ordnerbindung finden Sie in späteren Abschnitten.

- Die Eigenschaftsgruppe *Binding* (Bindung) dient zur Steuerung einer erweiterten Funktion, die als Ordnerbindung bezeichnet wird. Sie wird im Folgenden ausführlich erörtert.
- Die Eigenschaftsgruppe *Details* dient zur Steuerung einer erweiterten Funktion, die als Detailseitenerstellung bezeichnet wird. Sie wird im Folgenden ausführlich erörtert.

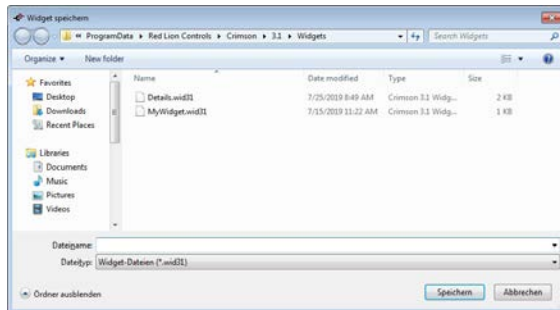
11.3 Archivieren von Widgets

Im Eigenschaften-Dialogfeld jedes Widgets befindet sich eine Registerkarte namens „Filing“ (Archivieren):



Mit den Eigenschaften *Description* (Beschreibung) und *Category* (Kategorie) wird gesteuert, wie ein Widget im Ressourcenfenster angezeigt wird, nachdem es gespeichert wurde. Alle Widgets derselben Kategorie werden in derselben Unterkategorie der Kategorie „Primitives“ (Primitive) zusammengefasst und die Widget-Beschreibung wird angezeigt, wenn der Benutzer den Mauszeiger auf ein Element hält.

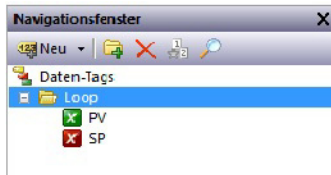
Zum Speichern eines Widgets wählen Sie es einfach aus und wählen im Menü „Edit“ (Bearbeiten) die Option „Save Widget“ (Widget speichern) oder drücken die Tastenkombination **STRG+Q**. Ein normales Speichern-Dialogfeld wird geöffnet, in dem Sie das Widget im Widget-Verzeichnis von Crimson als `wid`-Datei speichern können:



Das Ressourcenfenster wird automatisch aktualisiert, wenn diesem Verzeichnis eine Widget-Datei hinzugefügt wird. Dabei spielt es keine Rolle, ob die Änderung über Crimson erfolgt oder ob in Windows Explorer einfach eine `wid`-Datei in das Verzeichnis gezogen wird. Beachten Sie, dass Widget-Dateien eigenständig sind und zwischen Crimson-Installationen auf verschiedenen Rechnern übertragen werden können. Dies stellt eine leistungsstarke Methode zum Freigeben von Benutzeroberflächenelementen oder zum Austauschen von Elementen mit anderen Ingenieuren dar, wenn mehrere Personen an einem Projekt arbeiten.

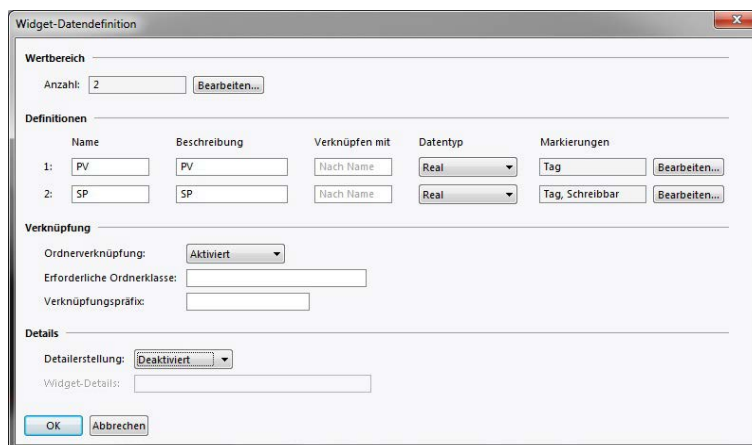
11.4 Ordnerbindung

Die Fähigkeit von Crimson, Tags in Ordnern zu organisieren, ermöglicht eine Art von objektorientiertem Design, wobei Tags, die die Eigenschaften eines Objekts darstellen, in einem Ordner gruppiert werden können, der das Objekt selbst darstellt. Betrachten Sie folgendes Beispiel:



Hier wurde ein Ordner erstellt, der eine PID-Schleife (Loop) darstellt. Außerdem wurden Tags erstellt, die auf den Prozesswert (PV) und Sollwert (SP) der Schleife verweisen. Die Tags werden im Code als `Loop.PV` und `Loop.SP` bezeichnet und es werden die Standardregeln von Crimson für die Verwendung von verschachtelten Elementen angewendet.

Mit der Ordnerbindung können Sie ein Widget erstellen, das die Objekt- und Eigenschaftsstruktur widerspiegelt, die Sie in Ihren Tags erstellt haben. Beachten Sie folgende Datendefinition:



Hier wurden Datenelemente erstellt, deren Namen mit den Namen der Tags übereinstimmen, die eine PID-Schleife bilden. Wir haben von Menschen lesbare Namen für sie bereitgestellt und beide Datenelemente als Tags markiert. Außerdem haben wir den Sollwert als beschreibbar definiert. Beachten Sie, dass jetzt für jedes Datenelement eine neue Eigenschaft namens *Bind to* (Binden an) angezeigt wird. Diese Eigenschaft wird in der erweiterten Ordnerbindung noch ausführlicher erörtert.

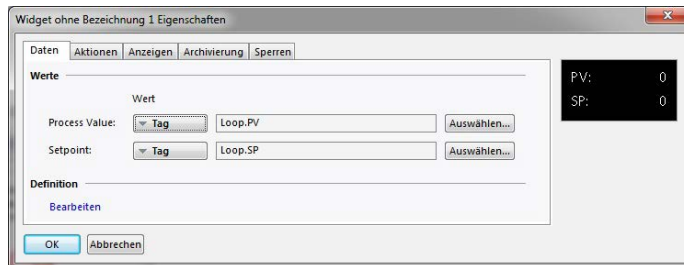
Im Abschnitt „Binding“ (Bindung) haben wir die Ordnerbindung aktiviert. Dies bedeutet, dass Crimson die automatische Bindung aller Datenelemente an Tags aus einem einzigen Quellordner unterstützen soll. Wir speichern diese Änderungen und wählen das Kontextmenü des Widgets aus. Das Menü enthält nun den neuen Befehl „Bind Widget“ (Widget binden), mit dem der Bindungsvorgang durchgeführt werden kann.

Wenn Sie den Befehl auswählen oder **STRG+B** drücken, wird folgendes Dialogfeld angezeigt:



Wenn wird den Loop-Ordner aus dem Ressourcenfenster auf das Ziel ziehen, werden die Datenelemente des Widgets automatisch an die entsprechenden Tags im Ordner gebunden.

Die Ergebnisse werden im Eigenschaften-Dialogfeld des Widgets angezeigt:



Mit anderen Worten: Jedes Datenelement wurde an jenes Tag im ausgewählten Ordner gebunden, dessen Name mit seinem eigenen Datenelementnamen übereinstimmt. Dies eröffnet Ihnen leistungsstarke Möglichkeiten: Sie können mehrere Eigenschaften definieren und in einen einzigen Vorgang binden, wodurch die Entwurfszeit verkürzt und die Wiederverwendung bereits erstellter Elemente verbessert wird.

11.5 Erweiterte Bindung

Die Ordnerbindung unterstützt eine Reihe erweiterter Optionen.

11.5.1 Klassenübereinstimmung

Die erste und einfachste Option ist die Einstellung *Required Folder Class* (Erforderliche Ordnerklasse) in den Eigenschaften des Widgets. Mit dieser Option können die bei der Bindung akzeptierten Ordner eingeschränkt werden, wodurch sich eine falsche Zuordnung von nicht übereinstimmenden Objekttypen vermeiden lässt. Die im Widget angegebene Klasse muss mit der Klasse im Ordner übereinstimmen, der gebunden werden soll. Andernfalls wird ein Fehler verursacht.

11.5.2 Bindungsvorzeichen

Die Eigenschaft *Binding Prefix* (Bindungsvorzeichen) kann bei der Verschachtelung von Widgets verwendet werden, damit untergeordnete Widgets an Unterordner des Ordners gebunden werden können, an den das übergeordnete Widget gebunden ist. Angenommen, Sie erstellen ein Doppelschleifen-Widget, das an einen Ordner gebunden werden soll, der die beiden PID-Ordner „Loop1“ und „Loop2“ enthält. Wenn Sie das Bindungspräfix der untergeordneten Widgets jeweils auf einen der Schleifennamen setzen, können Sie gewährleisten, dass die Widgets an unterschiedliche untergeordnete Ordner des Ordners gebunden werden, der auf das übergeordnete Widget gezogen wird. Wenn beispielsweise das erste untergeordnete Widget das Bindungspräfix `Loop1` hat und das übergeordnete Widget an den Ordner „Dual“ gebunden ist, werden die Eigenschaften des untergeordneten Widgets an die Ausdrücke `Dual.Loop1.PV` und `Dual.Loop1.SP` gebunden.

11.5.3 Verwenden von Binden an

Mit der Eigenschaft *Bind To* (Binden an) eines Datenelements kann der Ausdruck geändert werden, an den das Datenelement gebunden ist. Am einfachsten ist es, einen Namen einzugeben, der sich vom Namen des Datenelements unterscheidet. In diesem Fall wird dieser Name zur Auswahl des Tags verwendet, an das die Bindung erfolgen soll.

11.5.3.1 Verwenden von Punkten

Sie können auch einen Namen eingeben, der Punkte enthält. Diese wählen Tags in untergeordneten Ordnern des Quellordners aus. Wenn Sie beispielsweise `Remote.SP` eingeben, wird das betreffende Datenelement bei Bindung an den Ordner „Loop“ an den Ausdruck `Loop.Remote.SP` gebunden.

11.5.3.2 Verwenden von Caret-Zeichen

Um in der Ordnerstruktur nach oben zu gehen, können Sie dem Namen Caret-Zeichen voranstellen. Mit jedem dieser Zeichen gelangen Sie um jeweils eine Ebene nach oben. Ein Datenelement mit einer „Bind To“ (Binden an-) Einstellung `^Name` in einem Widget, das an `Dual.Loop` gebunden ist, bindet sich selbst an den Ausdruck `Dual.Name`.

11.5.3.3 Spezielle Namen

Sie können auch einen der speziellen Namen für *Bind To* (Binden an) verwenden:

Name	Ergebnis
::Path	Der vollständige Pfad des Tags, an das dieses Widget gebunden wurde, einschließlich aller übergeordneten Ordner.
::Name	Der vollständige Name des Tags, an das dieses Widget gebunden wurde, außer den übergeordneten Ordnern.
::TopPath	Der vollständige Pfad des Tags, an das das Stamm-Widget bei einem verschachtelten Bindungsvorgang gebunden wurde. Entspricht ::Path bei der nicht verschachtelten Bindung.
::TopName	Der Name des Tags, an das das Stamm-Widget bei einem verschachtelten Bindungsvorgang gebunden wurde. Entspricht ::Name bei der nicht verschachtelten Bindung.

Beachten Sie, dass jeder dieser speziellen Namen als Zeichenfolgenkonstante gleich dem erforderlichen Namen ausgewertet wird und nicht als das eigentliche Tag selbst. Sie werden in der Regel dazu verwendet, dem Benutzer Informationen zu dem Ordner bereitzustellen, an den ein Widget oder dessen Stamm-Widget gebunden wurde.

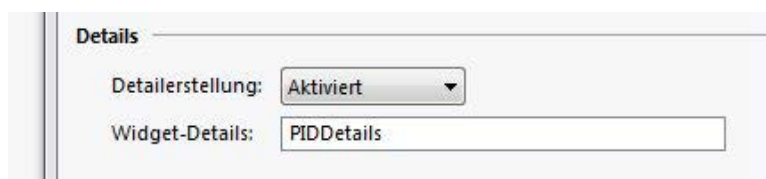
11.6 Details-Widgets

Angenommen, Sie haben ein PID-Widget erstellt, möchten aber detailliertere Statusinformationen anzeigen, wenn der Benutzer eine Schaltfläche in diesem Widget drückt. In diesem Fall könnten Sie einfach ein größeres und vielleicht komplexeres Widget erstellen, das Sie dann an dieselbe Schleife binden. Dieses Widget könnten Sie auf einer anderen Seite platzieren und dann diese Seite im ursprünglichen Übersichts-Widget auswählen. Eventuell könnten Sie ein Datenelement verwenden, um dem Widget mitzuteilen, welche Seite verwendet werden soll.

Beim Erstellen eines Details-Widgets werden all diese Schritte automatisch ausgeführt.

11.6.1 Aktivieren der Detailerstellung

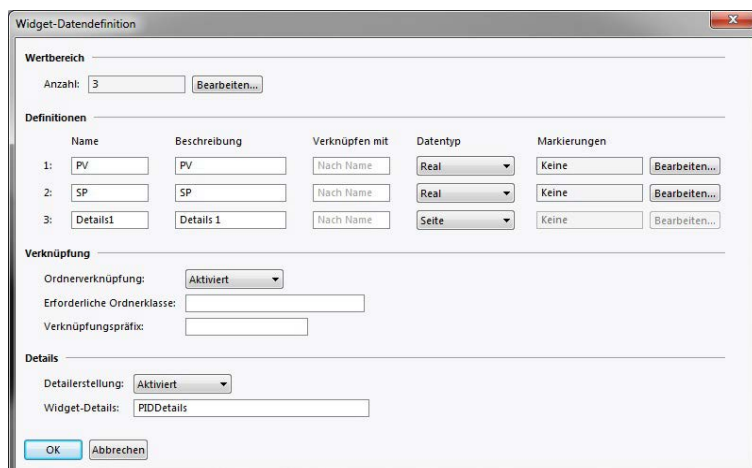
Diese Funktion wird über die Eigenschaft *Details Creation* (Detailerstellung) in der Datendefinition des Widgets gesteuert:



Mit der Eigenschaft *Details Widget* (Details-Widget) wird von einem oder mehreren Details-Widgets, die Sie auf eigenen Seiten platzieren möchten, eine durch Kommas getrennte Liste bereitgestellt. Jedes Widget wird mit dem Dateinamen angegeben, unter dem es gespeichert wurde. Im oben angezeigten Beispiel soll ein Details-Widget aus der Datei `PIDDetails.wid` im Widget-Verzeichnis von Crimson extrahiert werden.

11.6.2 Definieren von Datenelementen

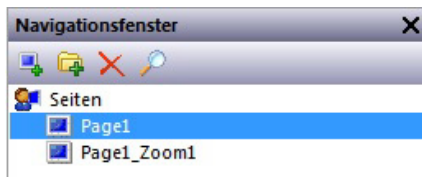
Wir müssen auch Datenelemente im Übersichts-Widget bereitstellen, damit wir auf die Namen der Seiten zugreifen können, die für die Details-Widgets erstellt werden. Diese Eigenschaften müssen die Namen `Details1`, `Details2` usw. tragen, wobei für jedes Element in der Liste der Details-Widgets ein Datenelement angegeben werden muss. Jedes Datenelement muss vom Datentyp „Page“ (Seite) sein. Im folgenden Beispiel haben wir eine einzelne Eigenschaft erstellt, die den Seitennamen unserer einzigen Detailseite beinhaltet:



11.6.3 Ergebnisse der Bindung

Wenn wir das Übersichts-Widget an unsere PID-Schleife binden, wird eine neue Seite erstellt, die das Details-Widget aufnimmt. Der Name der neuen Seite basiert auf dem Namen der Seite, die das Übersichts-Widget enthält, sowie aus dem Suffix „Zoom“ und einer Zahl, durch die der Name eindeutig ist.

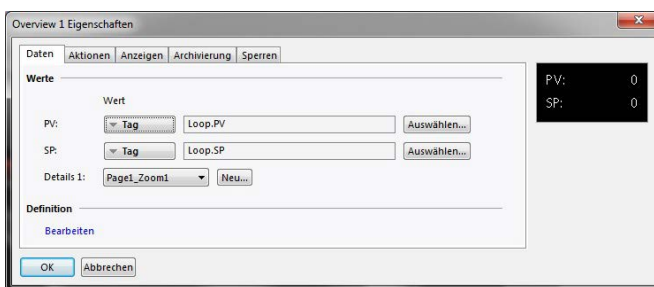
Diese Seite wird in der Navigationsliste unter der aktuellen Seite platziert:



Die auf dieser Seite erstellten Details sind an unsere Schleife gebunden:



Die Eigenschaften unseres Übersichts-Widgets werden wie folgt geändert:



In unserem Übersichts-Widget können wir einfach eine Schaltfläche definieren und über diese Schaltfläche die Aktion `ShowPopup(Details1)` aufrufen, wodurch das zugehörige Details-Widget angezeigt wird. Das Details-Widget selbst kann das Popup durch Aufrufen von `HidePopup()` schließen.

11.6.4 Mehrere Detailseiten

Falls mehrere Detailseiten erstellt wurden, erinnern Sie sich sicher daran, dass die Datenelemente `Details1`, `Detail2` usw. im Übersichts-Widget die Namen dieser Seiten enthalten. Diese Datenelemente können auch in den Details-Widgets definiert werden und werden ebenfalls auf die Namen der erstellten Seiten gesetzt. Dies ist nützlich, wenn Sie der ersten Detailseite ermöglichen möchten, zur zweiten zu navigieren usw. Dadurch werden die Seiten miteinander verknüpft. Details-Widgets können auch ein spezielles Datenelement namens `DetailsP` definieren, das der Seite mit dem Übersichts-Widget gleichgesetzt wird. Damit können Sie zur Übersicht zurückkehren. Dies wäre mit einem einfachen `GotoPrevious()` nicht möglich, wenn mehrere Detailseiten bereitgestellt werden.

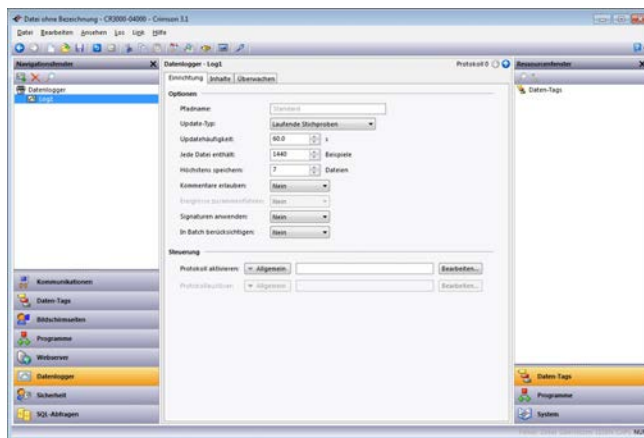
Kapitel 12 Verwenden des Datenloggers

Jetzt, da Sie den Kern Ihrer Anwendung konfiguriert haben, möchten Sie u. U. den Datenlogger von Crimson® 3.1 nutzen, um bestimmte Tag-Werte auf einer Speicherkarte aufzeichnen. Auf diese Weise aufgezeichnete Daten werden in branchenüblichen CSV-Dateien gespeichert und lassen sich anhand verschiedener Methoden leicht in Anwendungen wie Excel importieren. Zum Konfigurieren der Datenprotokollierung wählen Sie im Navigationsfenster die Kategorie „Data Logger“ (Datenlogger) aus.

12.1 Erstellen von Datenprotokollen

Datenprotokolle werden auf die übliche Weise in der Navigationsliste erstellt.

Jedes Protokoll verfügt über die folgenden Eigenschaften:



12.1.1 Einrichtungseigenschaften

- Mit der Eigenschaft *Path Name* (Pfadname) können Sie das Verzeichnis ändern, in dem das Protokoll gespeichert wird. Standardmäßig wird das Protokoll in einem Verzeichnis gespeichert, das nach dem Namen des Protokolls benannt wird. Wenn Sie ein Protokoll umbenennen, aber die zugehörigen Daten beibehalten möchten, verwenden Sie diese Eigenschaft, um das Standardverzeichnis zu überschreiben.
- Bei der Eigenschaft *Update Type* (Aktualisierungstyp) hat der Benutzer die Wahl zwischen den beiden Datenerfassungstypen *Continuous Sample* (Laufende Stichproben) und *Triggered Snapshot* (Ausgelöste Momentaufnahme).
- Mit der Eigenschaft *Update Rate* (Aktualisierungshäufigkeit) wird angegeben, wie oft Crimson eine Stichprobe der zu protokollierenden Datenelemente nimmt. Obwohl eine Dezimalstelle eingegeben werden kann, ist die Stichprobennahme nur auf 200 ms genau. Die schnellste Abtastrate ist eine Sekunde. Beachten Sie jedoch, dass bei einer so hohen Rate sehr große Datenmengen entstehen. Alle Tags im Protokoll werden mit derselben Rate abgetastet.
- Die Eigenschaft *Each File Holds* (Jede Datei enthält) wird in Verbindung mit der Eigenschaft *Update Rate* (Aktualisierungshäufigkeit) verwendet, um zu bestimmen, wie oft eine neue Datendatei erstellt wird. Im Modus *Continuous Sample* (Laufende Stichproben) ist dies die Anzahl der Stichproben, die in die einzelnen Protokolldatei aufgenommen werden. In der Regel wird dieser Wert so eingestellt, dass jede Protokolldatei eine vernünftige Datenmenge enthält. Das oben gezeigte Protokoll z. B. ist so konfiguriert, dass jeden Tag eine neue Protokolldatei verwendet wird. Im Modus *Triggered Snapshot* (Ausgelöste Momentaufnahme) enthält jede Datei nur so viele Stichproben, wie nach Eintreten des

Auslösers in dem Zeitraum protokolliert wurden, nach dessen Ablauf eine neue Protokolldatei erstellt wird. Jede neue Protokolldatei erhält einen anderen Namen.

- Mit der Eigenschaft *Retain At Most* (Höchstens speichern) wird angegeben, wie viele Protokolldateien auf der Speicherkarte aufbewahrt werden, bevor die älteste Datei gelöscht wird. Diese Eigenschaft sollte so eingestellt werden, dass der Nutzer der protokollierten Daten die Daten aus dem Crimson-Gerät extrahieren kann, bevor sie gelöscht werden. Das oben abgebildete Protokoll wurde so konfiguriert, dass Daten eine Woche lang aufbewahrt werden.
- Mit der Eigenschaft *Allow Comments* (Kommentare erlauben) wird das Hinzufügen von Kommentaren zum Datenprotokoll über die Funktion `LogComment()` aktiviert bzw. deaktiviert. Weitere Informationen zur Verwendung dieser Funktion finden Sie im Referenzhandbuch.
- Die Eigenschaft *Merge Events* (Ereignisse zusammenführen) ist aktiv, wenn die Eigenschaft *Allow Comments* (Kommentare erlauben) auf *Yes* (Ja) gesetzt ist. Mit der Eigenschaft *Merge Events* (Ereignisse zusammenführen) können Ereignisse, die mit den Logger-Tags verbunden sind, und die Ereignisse in der Monitorliste im Datenprotokoll zusammengeführt werden. Diese Ereignisse werden weiterhin in der Ereignisdatei protokolliert.
- Mit der Eigenschaft *Apply Signatures* (Signaturen anwenden) wird das Hinzufügen von kryptografischen Signaturen zum Protokoll gesteuert. Nachfolgend finden Sie weitere Informationen zu diesen Signaturen und wie sie eingesetzt werden können, um zu gewährleisten, dass die Protokollintegrität gewahrt bleibt.
- Mit der Eigenschaft *Include in Batch* (In Batch berücksichtigen) wird angegeben, ob dieses Protokoll vom Batch-Protokollierungssystem berücksichtigt oder ignoriert werden soll. Die Batch-Protokollierung wird im Folgenden noch ausführlicher behandelt.
- Mit der Eigenschaft *Log Enable* (Protokoll aktivieren) können Sie die Protokollierung ein- bzw. ausschalten. Wenn der eingegebene Ausdruck „true“ ist, wird die Protokollierung aktiviert. Ist der Ausdruck „false“, wird die Protokollierung deaktiviert. Wird kein Ausdruck eingegeben, wird die Protokollierung standardmäßig aktiviert.
- Die Eigenschaft *Log Trigger* (Protokollauslöser) ist aktiv, wenn die Eigenschaft *Update Type* (Aktualisierungstyp) auf *Triggered Snapshot* (Ausgelöste Momentaufnahme) gesetzt ist. Mit der Eigenschaft *Log Trigger* (Protokollauslöser) wird festgelegt, welches Ereignis eine Momentaufnahme der aktuellen Datenpunkte auslöst.

12.1.2 Inhaltseigenschaften

- Auf der Registerkarte *Contents* (Inhalte) können Sie angeben, welche Tags in das Datenprotokoll aufgenommen werden sollen. Tags können aus dem Ressourcenfenster in die Liste gezogen und innerhalb der Liste mithilfe von Drag-and-Drop nach oben und unten verschoben werden.

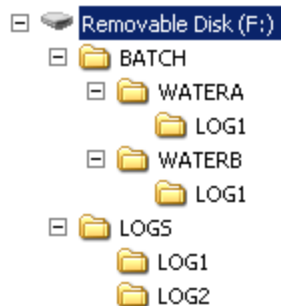
12.1.3 Monitoreigenschaften

- Die Registerkarte *Monitor* ist aktiv, wenn die Eigenschaften *Allow Comments* (Kommentare erlauben) und *Merge Events* (Ereignisse zusammenführen) beide auf *Yes* (Ja) gesetzt sind. Auf der Registerkarte *Monitor* können Sie zusätzliche Tags auswählen, deren zugehörige Ereignisse im Datenprotokoll zusammengeführt werden.

12.2 Batch-Protokollierung

Beim ersten Zugriff auf den Datenlogger wird eine globale Einstellung zur Aktivierung bzw. Deaktivierung der Batch-Protokollierung angezeigt. Bei der normalen Datenprotokollierung speichert der Datenlogger die Protokolldateien unter dem Ordernamen, der für die einzelnen Protokolle angegeben wurde. Bei der Batch-Protokollierung hingegen werden alle derart konfigurierten Protokolle auch in einem Verzeichnis gespeichert, das nach dem aktuellen Produktions-Batch benannt wurde. Dadurch können alle Protokolle, die zu einem bestimmten Batch gehören, als Gruppe aufgerufen und bearbeitet werden.

Dies soll anhand der folgenden Verzeichnisstruktur veranschaulicht werden:



Dieses Beispiel stammt von einem Zielgerät, auf dem die Batch-Protokollierung aktiviert ist und zwei Datenprotokolle konfiguriert sind. Das erste Datenprotokoll ist so eingestellt, dass es im Batch berücksichtigt wird, das zweite hingegen nicht. Wie Sie sehen, werden die Protokolldateien standardmäßig in den Verzeichnissen LOGS/LOG1 und LOGS/LOG2 gespeichert. Das erste Protokoll wird außerdem in Unterverzeichnissen des Verzeichnisses `Batch` abgelegt. Jedes Unterverzeichnis enthält die Daten, die zwischen dem Anfang und Ende des Batches erfasst wurden.

12.2.1 Steuern eines Batch

Die Batch-Protokollierung wird über eine Reihe von Funktionen gesteuert. `NewBatch(name)` erstellt einen Ordner namens `name`, wodurch der aktuelle Batch beendet und ein neuer gestartet wird. Dateien, die nach diesem Befehl aufgezeichnet werden, werden unter dem neuen Ordner gespeichert. Die Funktion `EndBatch()` stoppt den aktuellen Batch während `GetBatch()` den Namen des gegenwärtig aktiven Batch zurückgibt. Weitere Informationen finden Sie im Referenzhandbuch.

12.3 Digitale Signaturen

Crimson unterstützt das Hinzufügen kryptografischer Signaturen zu Daten-, Ereignis- und Sicherheitsprotokollen und ermöglicht Ihnen dadurch, die Integrität einer Protokolldatei zu überprüfen. Die Signaturen zeigen, ob das Protokoll manipuliert wurde oder ob Daten aus der Datei entfernt wurden. Sie geben Ihnen auch die Gewissheit, dass eine bestimmte Protokolldatei tatsächlich vom angegebenen Gerät stammt.

Durch die Aktivierung von Signaturen wird der CSV-Datei ein zusätzliches Feld hinzugefügt, um die Speicherung der erforderlichen Daten zu ermöglichen. Wenn Sie sich eine solche Datei näher anschauen, werden Sie feststellen, dass dieses zusätzliche Feld alle paar Zeilen eine große Datenmenge enthält. Bei diesen Daten handelt es sich um die digitale Signatur. Eine Signatur ist ein Wert, der von den Daten in den vorhergehenden Zeilen mathematisch in einer Art und Weise abgeleitet wird, dass es unmöglich ist, die Gültigkeit der Signatur aufrecht zu erhalten, wenn die Daten geändert werden.

Signaturen können mithilfe des Befehlszeilen-Hilfsprogramms SigCheck geprüft werden, das mit der Crimson Software bereitgestellt wird. Das Hilfsprogramm wird entweder bestätigen, dass die Datei gültig ist, oder die Zeile angeben, in der ein Prüffehler auftritt. Wenn Sie sich bezüglich der Integrität einer Datei absolut sicher sein möchten, sollten Sie auch die digitale Signatur des Hilfsprogramms SigCheck prüfen. Dadurch können Sie sicherstellen, dass Ihnen eine von Red Lion genehmigte Software vorliegt und keine geänderte Version, die ungültige Ergebnisse liefert.

Um Ihr Vertrauen in die Integrität des Signaturprozesses weiter zu stärken, ist beim technischen Support ein technischer Hinweis erhältlich, in dem der Algorithmus detailliert beschrieben wird. Sie können auch den Quellcode für SigCheck abrufen, damit Sie unabhängig prüfen können, ob die Signaturen der Protokolldateien richtig validiert werden.

12.4 Speicherung von Protokolldateien

Wie oben beschrieben, speichern Datenprotokolle ihre Daten in einer Reihe von Dateien auf der Speicherkarte des Zielgeräts. Die Dateien werden in dem Unterverzeichnis abgelegt, das in den Eigenschaften des Protokolls angegeben wurde. Dieses Verzeichnis wiederum wird unter einem Stammverzeichniseintrag namens LOGS gespeichert.

Protokolldateien werden nach der Uhrzeit und dem Datum benannt, zu der bzw. an dem das Protokoll planmäßig beginnen soll. Wenn jede Datei eine Stunde oder mehr Informationen enthält, werden die Dateien nach dem Muster JJMMTThh.CSV benannt. Dabei steht JJ für das Jahr, MM für den Monat, TT für den Tag und hh für die Stunde. Wenn jede Datei weniger als eine Stunde Informationen enthält, werden die Dateien nach dem Muster MMTThhmm.CSV benannt. Die ersten sechs Zeichen haben die oben beschriebene Bedeutung und mm steht für die Minute, in der das Protokoll begann. Diese Regeln gewährleisten, dass jede Protokolldatei einen eindeutigen Namen hat, aus dem der Zeitpunkt ihrer Erstellung hervorgeht.

Die Länge jeder Datei hängt von den Eigenschaften *Update Rate* (Aktualisierungshäufigkeit) und *Each File Holds* (Jede Datei enthält) ab. Bei einer Aktualisierungshäufigkeit von 5 Sekunden und 360 Stichproben zum Beispiel enthält jede Datei $(5 \times 360) / 60 = 30$ Minuten Daten. Verwenden Sie daher das Format MMTThhmm.CSV für den Dateinamen. Alle 30 Minuten wird eine neue Datei erstellt, und zwar entweder zur vollen oder zur halben Stunde.

12.5 Der Protokollierungsvorgang

Der Datenlogger von Crimson führt zwei separate Vorgänge aus. Im ersten Vorgang wird jeder Datenpunkt in der von den Eigenschaften jedes Protokolls angegebenen Häufigkeit erfasst und die Daten werden in einem Puffer im RAM des Zielgeräts abgelegt. Der zweite Vorgang wird alle zwei Minuten ausgeführt und schreibt die Daten aus dem Speicher auf die Speicherkarte.

Diese Struktur bietet mehrere Vorteile:

- Schreibvorgänge auf die Speicherkarte beginnen garantiert immer in einem Zwei-Minuten-Intervall, d. h. um genau 2, 4 oder 6 Minuten usw. nach der vollen Stunde. Wenn das Zielgerät Hot-Swapping unterstützt, können Sie den Beginn des nächsten Schreibvorgangs abwarten. Wenn die LED für die Speicherkartenaktivität nicht mehr blinkt, wissen Sie, dass der nächste Schreibvorgang frühestens bei Beginn des nächsten Zwei-Minuten-Intervalls versucht wird. Sie haben also Zeit, die Speicherkarte zu entfernen, ohne eine Beschädigung von Daten befürchten zu müssen. Solange Sie eine neue Karte vor Ablauf von vier Minuten einsetzen, bleiben alle Daten erhalten.
- Schreibvorgänge auf die Karte sind deutlich effizienter, weil die kontinuierliche Aktualisierung der Datenstrukturen im Dateisystem der Karte bei jeder einzelnen Stichprobe entfällt. Bei Protokollen, bei denen für die Stichprobennahme sehr hohe Datenraten konfiguriert wurden, würde die Bandbreite einer typischen Speicherkarte nicht gestatten, die Daten ohne einen solchen Pufferungsvorgang zuverlässig aufzuzeichnen.

Weil Daten bis zu zwei Minuten lang nicht auf die Speicherkarte übertragen werden, können Protokolldaten in diesem Umfang verloren gehen, wenn das Terminal heruntergefahren wird. Wenn das Zielgerät während eines Schreibvorgangs heruntergefahren wird, wird zudem möglicherweise die Speicherkarte beschädigt. Um sicherzustellen, dass eine solche Beschädigung nicht permanent ist, verwendet Crimson ein Journaling-System, das Schreibvorgänge im zusätzlichen permanenten Speicher im Terminal zwischenspeichert. Wenn das Gerät erkennt, dass beim Herunterfahren ein Schreibvorgang unterbrochen wurde, wird der Schreibvorgang beim Hochfahren wiederholt. Hierdurch wird eine etwaige Beschädigung rückgängig gemacht, und die Karte wird repariert.

Wenn Sie eine Speicherkarte aus einem Panel entfernen möchten, das eine Datenprotokollierung durchführt, müssen Sie das oben beschriebene Verfahren hinsichtlich der Aktivitäts-LED beachten und dürfen das Gerät erst herunterfahren, wenn die Aktivität abgeschlossen ist. Wenn Sie nicht sicher sind, ob das Terminal ordnungsgemäß heruntergefahren wurde, fahren Sie es wieder hoch, warten Sie, bis die Schreibvorgang abgeschlossen ist, und fahren Sie das Terminal ordnungsgemäß herunter. Die Karte kann dann sicher entfernt werden.

Da das Entfernen einer Speicherkarte ein wenig kompliziert ist, bietet Crimson eine Vielzahl weiterer Methoden für den Zugriff auf Protokolldateien, bei denen keine Karten entfernt werden müssen. Diese Methoden werden im Folgenden beschrieben.

12.6 Zugriff auf Protokolldateien

Für den Zugriff auf Protokolldateien gibt es fünf Methoden:

- Sie können die Speicherkarte als Laufwerk auf einem PC anmelden, wie am Anfang dieses Handbuchs beschrieben. Die Protokolle können dann über Windows Explorer kopiert werden. Diese Methode hat einige Nachteile hinsichtlich der potenziellen Auslastung der Speicherkarte nach der ersten Anmeldung in Windows.
- Sie können den Webserver verwenden, der im nächsten Kapitel beschrieben wird. Wenn der Webserver aktiviert ist, können Protokolldateien über eine TCP/IP-Verbindung abgerufen werden, und zwar entweder über einen Webbrowser wie Microsoft Internet Explorer oder mithilfe des automatisierten Prozesses des Hilfsprogramms WebSync, das mit Crimson geliefert wird.
- Sie können den FTP-Server verwenden, um externen Clients zu gestatten, eine Verbindung zum Crimson-Gerät herzustellen und die Protokolle herunterzuladen. Weitere Informationen finden Sie im Kapitel „Verwenden von Diensten“.
- Sie können den Sync Manager (Synchronisierungsmanager) verwenden, um die Dateien regelmäßig an einen FTP-Server zu übertragen. Weitere Informationen finden Sie im Kapitel „Verwenden von Diensten“.
- Sie können das automatische Kopieren der Protokolldateien auf einen USB-Stick aktivieren, indem Sie in der Kategorie „Communication“ (Kommunikation) die Option „Memory Stick“ (Speicher-Stick) konfigurieren. Weitere Einzelheiten finden Sie im Kapitel „Verwenden von Kommunikationen“.

Kapitel 13 Verwenden des Webserver

Mit dem Webserver von Crimson können verschiedene Daten über TCP/IP-Verbindungen offen gelegt werden, entweder mithilfe von Modems oder mithilfe der Ethernet-Anschlüsse des Zielgeräts. Dies ermöglicht den Fernzugriff auf Diagnoseinformationen oder auf die vom Datenlogger aufgezeichneten Werte. Zur Konfiguration des Webserver wählen Sie im Navigationsfenster die Kategorie „Web Server“ (Webserver) aus.

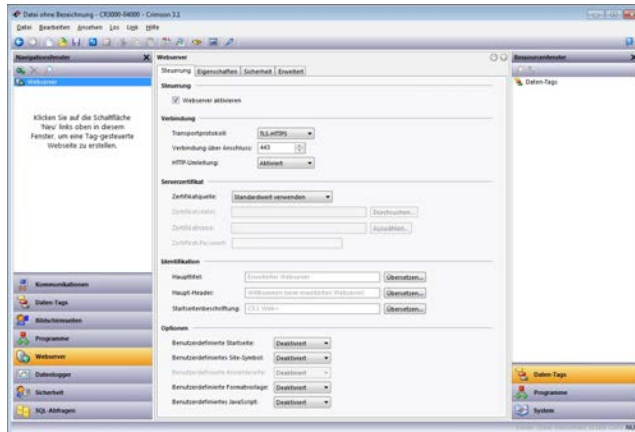
13.1 Wechseln zwischen Versionen

Beachten Sie Folgendes: Wenn Sie den Crimson 3.1-Webserver auf einem Gerät testen und dieses Gerät anschließend auf Crimson 3.0 zurückstellen, müssen Sie möglicherweise die Taste **F5** drücken (oder je nach Webbrowser **STRG+F5** oder **UMSCHALT+F5**), um den Seitencache des Browsers zu leeren und die Crimson 3.0-Version der aktuellen Seite wieder zu laden. Zwar haben viele Seiten unterschiedliche URLs und sind daher gegen dieses Problem gefeit, aber bei der Startseite und der Fernanzeige-Seite kann es in Ihrem Browser beim Wechsel zwischen Versionen zu Verwechslungen kommen.

13.2 Webserver-Eigenschaften

Die Eigenschaften des Webserver werden über den Stammeintrag in der Navigationsliste aufgerufen.

13.2.1 Steuerungseigenschaften

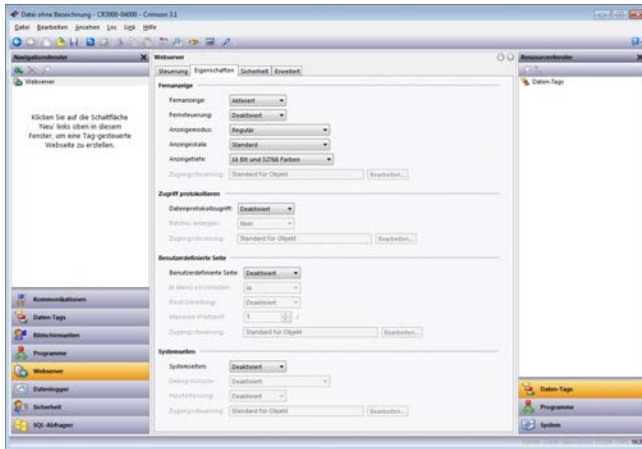


- Mit der Eigenschaft *Enable Web Server* (Webserver aktivieren) können Sie den Webserver aktivieren bzw. deaktivieren. Wenn der Server aktiviert ist, wartet das Bedienfeld auf eingehende Anforderungen und erfüllt diese dann nach Bedarf. Wenn der Server deaktiviert ist, werden Verbindungen mit diesem Port verweigert. Denken Sie daran, dass der Server nur funktioniert, wenn über die Kategorie „Communications“ (Kommunikationen) eine Ethernet- oder Modemschnittstelle dafür konfiguriert wurde.
- Die Eigenschaft *Transport Protocol* (Transportprotokoll) dient zur Auswahl von TCP/IP und HTTP oder TLS und HTTPS. Letzteres Protokoll ist sicherer, weil die zwischen dem Server und seinen Clients übertragenen Daten verschlüsselt werden und ein Mechanismus für die Prüfung der Serveridentität bereitgestellt wird. TLS-HTTPS erfordert bei der Konfiguration jedoch mehr Aufwand und es ist ein gültiges Serverzertifikat erforderlich, um zu verhindern, dass Webbrowser eine unsichere Verbindung bemängeln.
- Die Eigenschaft *Connect Via Port* (Verbindung über Anschluss) gibt die TCP-Portnummer an, an der der Webserver überwacht. Port 80 ist der vom HTTP-Protokoll verwendete Standardport und Port 443 ist der Standardport für HTTPS. Diese Einstellungen dürften sich für Ihre Anwendung eignen, es sei denn, es wird ein Portzuordnungsgerät verwendet.
- Die Eigenschaft *HTTP Redirect* (HTTP-Umleitung) ist im HTTPS-Modus verfügbar und weist den Webserver an, alle HTTP-Anforderungen an Port 80 an die entsprechende Seite an Port 443 umzuleiten. Dadurch können die IP-Adresse oder der Hostname des Webserver ohne das Präfix https:// direkt in einen Browser eingegeben werden. Gleichzeitig wird gewährleistet, dass die sichere Version der Webseite aufgerufen wird. Beachten Sie, dass der für HTTPS verwendete Port zwar über die oben gezeigte Einstellung geändert werden kann, dass bei der HTTP-Umleitung aber immer Port 80 verwendet wird.
- Die Eigenschaftsgruppe *Server Certificate* (Serverzertifikat) ist im HTTPS-Modus verfügbar und gibt an, wie der Server das zur Identifizierung der Site verwendete Zertifikat abrufen soll. Sie können ein Zertifikat in Form einer PFX-Datei mit zugehörigem Kennwort angeben oder ein Zertifikat aus dem privaten Zertifikatspeicher von Windows auswählen. Wenn Sie kein Zertifikat angeben, wird das Standardserverzertifikat von Crimson verwendet. Weitere Informationen zum Arbeiten mit Zertifikaten finden Sie im folgenden Abschnitt.
- Mit der Eigenschaftsgruppe *Identification* (Identifizierung) können Sie Textzeichenfolgen angeben, die zur Identifizierung des Servers verwendet werden. Diese können zur Unterscheidung mehrerer Termini-

nals in einem Netzwerk verwendet werden. Dadurch wird gewährleistet, dass auf das richtige Gerät zugegriffen wird. Der *Main Title* (Haupttitel) wird in der Titelleiste des Webbrowsers angezeigt. Der *Main Header* (die Hauptkopfzeile) wird über dem Hauptmenü des Webservers angezeigt. Das *Home Label* (die Startseitenbeschriftung) wird links oben in der Menüleiste angezeigt, die auf jeder Webseite vorhanden ist.

- Mit der Eigenschaft *Custom Home Page* (Benutzerdefinierte Startseite) wird die vom Webserver angezeigte Standard-Startseite überschrieben. Wenn diese Eigenschaft aktiviert ist, sucht der Server im Verzeichnis `WEB` der Speicherkarte nach einer Datei namens `default.htm`. Die Crimson 3.1-Installation enthält im Verzeichnis `WebServer/Samples` eine Beispiel-Startseite. Das Beispiel ist identisch mit der Standardseite, enthält aber Tag-Daten und eine Meldung, dass die Seite angepasst wurde.
- Mit der Eigenschaft *Custom Logon Page* (Benutzerdefinierte Anmeldeseite) können Sie die Seite überschreiben, die vom Webserver verwendet wird, um Anmeldedaten im Authentifizierungsmodus „Form“ (Formular) zu erfassen. Wenn diese Eigenschaft aktiviert ist, sucht der Server im Verzeichnis `WEB` der Speicherkarte nach einer Datei namens `logon.htm`. Die Crimson 3.1-Installation enthält im Verzeichnis `WebServer/Samples` eine Beispiel-Anmeldeseite. Das Beispiel ist identisch mit der Standardseite, enthält aber den Hinweis, dass sie geändert wurde.
- Mit der Eigenschaft *Custom Style Sheet* (Benutzerdefinierte Formatvorlage) wird den vom Webserver bereitgestellten Standardseiten eine benutzerdefinierte CSS-Datei beigefügt. Der Kopfzeilenabschnitt dieser Seiten enthält eine Formatvorlage namens `/custom/css/custom.css`. Wenn diese Eigenschaft aktiviert ist, werden Verweise auf diese Datei einer Datei namens `custom.css` im Verzeichnis `WEB` der Speicherkarte zugeordnet. Wenn die Eigenschaft deaktiviert ist, geben derartige Verweise eine leere Datei zurück. Eine benutzerdefinierte CSS-Datei kann dazu verwendet werden, das Erscheinungsbild des Standard-Webservers zu überschreiben und Farben sowie Schriftarten usw. zu ändern. Die Crimson 3.1-Installation enthält im Verzeichnis `WebServer/Samples` ein Beispiel-Stylesheet. Das Muster ändert die Hintergrundfarbe der Seiten.
- Mit der Eigenschaft *Custom JavaScript* (Benutzerdefiniertes JavaScript) wird den vom Webserver bereitgestellten Standardseiten ein benutzerdefiniertes Skript beigefügt. Der Kopfzeilenabschnitt dieser Seiten enthält ein Skript namens `/custom/css/custom.js`. Wenn diese Eigenschaft aktiviert ist, werden Verweise auf diese Datei einer Datei namens `custom.js` im Verzeichnis `WEB` der Speicherkarte zugeordnet. Wenn die Eigenschaft deaktiviert ist, geben derartige Verweise eine leere Datei zurück. Sobald eine Webseite geladen wurde, versucht der Browser, eine JavaScript-Funktion namens `doCustom` auszuführen, damit das benutzerdefinierte Skript das Dokumentmodell ändern oder andere Anpassungen implementieren kann. Die Crimson 3.1-Installation enthält im Verzeichnis `WebServer/Samples` ein Beispielskript. Das Beispiel verwendet JQuery und bestimmte vordefinierte Anker innerhalb der Webseitenstruktur, um zusätzliche Menüpunkte hinzuzufügen. Weitere Informationen zu diesen Ankern finden Sie im nächsten Kapitel.

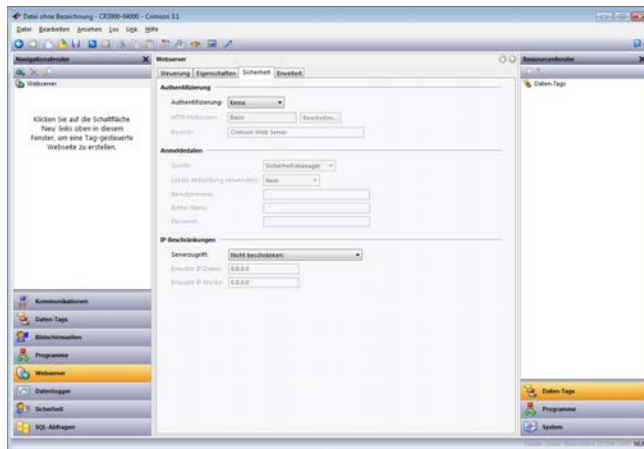
13.2.2 Einstellung der Eigenschaften



- Mit der Eigenschaft *Remote Viewing* (Fernanzeige) können Sie eine Funktion aktivieren bzw. deaktivieren, mit der ein Webbrowser zum Anzeigen des aktuellen Inhalts auf dem Display des Zielgeräts verwendet werden kann. Diese Funktion ist sehr nützlich, wenn Sie Probleme, die ein Bediener mit dem Bedienfeld oder dem damit gesteuerten Gerät hat, per Fernzugriff diagnostizieren müssen.
- Mit der Eigenschaft *Remote Control* (Fernbedienung) können Sie eine Funktion aktivieren bzw. deaktivieren, welche die Funktion „Fernanzeige“ erweitert. Bei aktivierter Funktion kann ein Webbrowser verwendet werden, um das Drücken von Schaltflächen oder Anzeigeprimitiven zu simulieren und dadurch das Bedienfeld bzw. das damit gesteuerte Gerät per Fernzugriff zu bedienen. Diese Funktion ist zwar sehr nützlich, aber Sie müssen mithilfe der verschiedenen Sicherheitsparameter sicherstellen, dass keine Manipulation möglich ist.
- Mit der Eigenschaft *Display Mode* (Anzeigemodus) wird gesteuert, wie die Fernanzeige im Webbrowser wiedergegeben wird. Im Modus „Regular“ (Normal) wird die Anzeige als Teil einer normalen Seite wiedergegeben, einschließlich einer Menüleiste, die die Navigation zu anderen Seiten ermöglicht. Im Modus „Full Window“ (Vollbild) wird die Fernanzeige auf das gesamte Browserfenster skaliert, wobei der Skalierungsfaktor beim Ändern der Fenstergröße angepasst wird. Dieser Modus ist nützlich für mobile Geräte oder wenn die Anzeige für die Wiedergabe in einem normalen Fenster zu groß ist. Die Modi „Bare Display“ (Bloße Anzeige) zeigen nur das Display des Geräts ohne Einfassung oder Tasten an. Diese Modi sind für den Einsatz in den Webbrowsern optimiert, die in Großbildfernseher integriert sind.
- Mit der Eigenschaft *Display Scale* (Anzeigegröße) wird die Skalierung des Anzeigebildes im Webbrowser gesteuert. Die meisten Geräte skalieren die Anzeige 1:1, doch Geräte mit geringer Auflösung verwenden möglicherweise standardmäßig 2:1. Dieses Verhalten kann mit dieser Einstellung außer Kraft gesetzt werden, um stets die Skalierung von 1:1 zu erzwingen.
- Mit der Eigenschaft *Display Depth* (Anzeigtiefe) wird gesteuert, wie viele Farben für die Wiedergabe des Anzeigebildes verwendet werden. Die Standardeinstellung 16 Bit unterstützt etwa 32.000 unterschiedliche Farben und bietet einen guten Kompromiss zwischen Leistung und Genauigkeit. Die Einstellung 24 Bit erzeugt eine perfekte Kopie der auf dem Display des Geräts angezeigten Informationen, doch es werden mindestens 50 % mehr Daten verwendet. (Durch die Verwendung von mehr Bits wird auch die Möglichkeit der Komprimierung verringert. Bei dieser Einstellung ändert sich also mehr als nur die Bitzahl.) Bei der Einstellung 8 Bit wird die gleiche palettenbasierte Technik wie in Crimson 3.0 eingesetzt. Hierdurch wird die Datennutzung um mindestens 50 % gesenkt, doch die Genauigkeit wird ebenfalls erheblich verringert. Diese Einstellung wird nur für gebührenpflichtige Anwendungen oder solche mit geringer Bandbreite empfohlen.
- Mit der Eigenschaft *Data Log Access* (Datenprotokoll-Zugriff) können Sie den Webzugriff auf die vom Datenlogger erstellten Dateien aktivieren bzw. deaktivieren. Diese Funktion muss aktiviert werden, wenn der Webserver von einem Remote-Client-Programm verwendet werden soll, um automatisch Datenprotokolle zu synchronisieren.

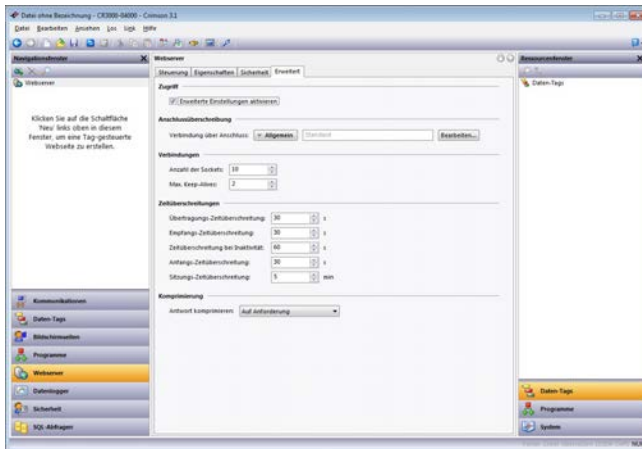
- Mit der Eigenschaft *Show Batches* (Batches anzeigen) wird die Anzeige von Batches im Datenprotokollbereich aktiviert bzw. deaktiviert. Wenn diese Option aktiviert ist, kann der Benutzer die aktuellen und vorherigen Batch-Protokolle zusätzlich zu den aktuellen Echtzeitprotokollen prüfen.
- Mit der Eigenschaft *Custom Site* (Benutzerdefinierte Site) können Sie ein Merkmal aktivieren oder deaktivieren, mit dem Dateien, die im Verzeichnis `WEB` der Speicherkarte gespeichert sind, über den Webserver zugänglich gemacht werden können. Dieses Merkmal wird im nächsten Abschnitt beschrieben und im nächsten Kapitel ausführlich erörtert. Beachten Sie, dass die benutzerdefinierte Startseite und die Anmeldeseite implementiert werden können, ohne diese Funktion zu aktivieren.
- Mit der Option *Include in Menu* (In Menü einschließen) wird gesteuert, ob die benutzerdefinierte Site explizit in das Standardmenü des Webserver sowie in die Menüleiste aufgenommen wird, die auf jeder Seite vorhanden ist. Sie müssen diese Funktion möglicherweise nicht aktivieren, wenn Sie die Startseite modifiziert oder die Menüstruktur mit JQuery geändert haben.
- Mit der Funktion *Root Direct* (Root-Umleitung) wird der Server angewiesen, unaufgelöste Verweise auf das Stammverzeichnis des Servers in das Verzeichnis `WEB` auf der Speicherkarte umzuleiten. Dies wird für neue Anwendungen nicht empfohlen, weil Red Lion sich das Recht vorbehält, andere Stammverzeichniseinträge zu erstellen, die Ihre Auswahl außer Kraft setzen können.
- Mit der Eigenschaft *Maximum Wait* (Maximale Wartezeit) wird festgelegt, wie lange Crimson 3.1 auf externe Kommunikationsdaten warten soll, bevor eine Webseite angezeigt wird. Dies gilt nur für benutzerdefinierte Webseiten mit Tag-Daten, die die im nächsten Kapitel beschriebenen eingebetteten Textwerte verwenden. Wenn diese Datenelemente sich nicht in der Kommunikationsprüfung befinden, muss Crimson sie vor dem Senden der Seite lesen. Diese Einstellung begrenzt die Wartezeit, um zu gewährleisten, dass Offlinedaten die Serverleistung nicht beeinträchtigen.
- Die Eigenschaft *System Pages* (Systemseiten) wird verwendet, um die verschiedenen Systemseiten zu aktivieren bzw. zu deaktivieren, die optional dem Webserver hinzugefügt werden können. Diese Seiten sind für die Inbetriebnahme und für Tests vorgesehen. **Sie sollten auf Produktionsgeräten nicht grundsätzlich aktiviert bleiben.** Insbesondere die Systemkonsole sollte nicht im Befehlsmodus aktiviert sein, es sei denn, es werden entsprechende Sicherheitseinstellungen zur Zugriffsbeschränkung verwendet.
- Mit der Eigenschaft *Debug Console* (Debug-Konsole) wird die Debug-Konsole ein- bzw. ausgeblendet. Die Seite kann entweder im Modus „Display Only“ (Nur Anzeige) oder im Modus „Command“ (Befehlsmodus) aktiviert sein. Im ersten Modus werden lediglich Diagnoseinformationen angezeigt, die von verschiedenen Teilen der Crimson-Laufzeitsoftware ausgegeben werden, während der zweite Modus die Ausführung verschiedener Systembefehle ermöglicht. Diese Befehle bieten erweiterte Funktionen, die den Betrieb des Geräts stören können. **Der Befehlsmodus sollte daher auf Produktionsgeräten nicht aktiviert bleiben.** Informationen zur Debug-Konsole finden Sie im Kapitel „Erweitertes Debugging“.
- Mit der Eigenschaft *Packet Capture* (Paketerfassung) kann eine Systemseite aktiviert bzw. deaktiviert werden, auf der die Daten von den Ethernet-Ports des Geräts erfasst werden können. Diese Funktion kann verwendet werden, um einige oder alle Daten zu erfassen, die über einen bestimmten Port übertragen werden, was das Debugging von Verbindungsproblemen vereinfacht. Außerdem kann so die unverschlüsselte Form der SSL-gesicherten Kommunikation überwacht werden. **Um Datenbeeinträchtigungen zu vermeiden, sollte sie daher auf einem Produktionsgerät nicht aktiviert bleiben.**
- Mit den verschiedenen Eigenschaften unter *Access Control* (Zugangssteuerung) kann der Zugriff auf die einzelnen Funktionen mithilfe einer Sicherheitsbeschreibung auf bestimmte Benutzer beschränkt werden. Die Eigenschaften sind nur verfügbar, wenn die Sicherheit aktiviert ist und die Crimson 3.1-Benutzerdatenbank für die Authentifizierung verwendet wird.

13.2.3 Sicherheitseigenschaften



- Die Eigenschaft *Authentication* (Authentifizierung) definiert, wie Crimson 3.1 Benutzer authentifiziert, die eine Verbindung mit dem Webserver herstellen. Wenn Sie „None“ (Keine) auswählen, wird die gesamte Authentifizierung deaktiviert und anonymer Zugriff auf die Ressourcen des Servers gewährt. Bei Auswahl von „Form“ (Formular) wird ein HTML-Formular verwendet, um die Eingabe des Benutzernamens und Kennworts zu ermöglichen. Dabei wird eine benutzerdefinierte Anmeldeseite verwendet, sofern der Server entsprechend konfiguriert ist. Diese Methode wird für HTTPS-Verbindungen empfohlen, weil sie attraktiver und besser anpassbar ist sowie die Möglichkeit bietet, sich auf konsistente und sichere Weise abzumelden. Sie sollte nicht über HTTP verwendet werden, weil sie das Kennwort in Klartext sendet. Wenn Sie „HTTP“ auswählen, wird die HTTP-eigene Authentifizierungsmethode verwendet, wobei normalerweise ein Popup im Webbrowser des Clients angezeigt wird. Wenn diese Methode ausgewählt ist, muss die Eigenschaft *HTTP-Method* (HTTP-Methode) so eingestellt werden, dass sie die zu verwendende Methode auswählt. Die Methode „Basic“ (Einfach) wird nicht für HTTP empfohlen, da sie das Kennwort wiederum in Klartext sendet. Bei der HTTP-Authentifizierung muss auch ein *Realm* (Bereich) angegeben werden. Dieser wird im Browser-Popup angezeigt, um die Identität der Site zu bestätigen.
- Mit der Eigenschaftsgruppe *Credentials* (Anmeldedaten) wird angegeben, wie Crimson 3.1 Informationen über Benutzernamen und Kennwörter erhalten soll. Unter *Source* (Quelle) sollte in der Regel „Security Manager“ (Sicherheitsmanager) übernommen werden, damit über die Kategorie „Security“ (Sicherheit) der Datenbank mehrere Benutzer mit unterschiedlichen Berechtigungen erstellt werden können. Bei älteren Datenbanken ist es manchmal notwendig, einen einzelnen Benutzer anzugeben, indem Sie „Source“ (Quelle) auf „Webserver“ setzen und die Informationen manuell eingeben. Diese Funktion ist auch in der Testphase nützlich, wenn ein Benutzer schnell erstellt werden muss. Beachten Sie, dass nur die Einstellung „Security Manager“ (Sicherheitsmanager) zulässt, dass die Sicherheit auf der Funktions- und Seitenebene festgelegt wird.
- Mit der Gruppe *IP Restrictions* (IP-Beschränkungen) wird der Webserverzugriff auf Hosts beschränkt, deren IP-Adresse mit der angegebenen Maske und den Daten übereinstimmt. Sie können den gesamten Zugriff oder aber nur die Fernsteuerung und Datenbearbeitung beschränken.

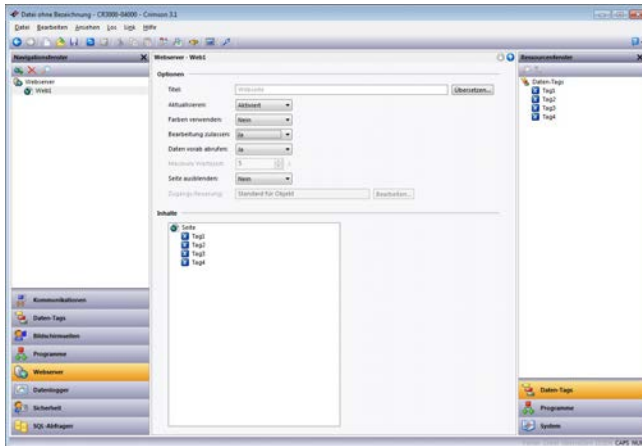
13.2.4 Erweiterte Eigenschaften



- Die Eigenschaft *Enable Advanced Settings* (Erweiterte Einstellungen aktivieren) aktiviert bzw. deaktiviert die anderen Einstellungen auf dieser Seite. Sie müssen in der Regel keine dieser Einstellungen anpassen, außer vielleicht „Port Override“ (Anschluss-Außerkraftsetzung), falls Sie eine Crimson 3.0-Datenbank importieren, die einen Ausdruck als Portnummer verwendet.
- Die anderen Einstellungen werden über die Sprechblasenhilfe erläutert. Passen Sie sie nur an, wenn Sie ein fortgeschrittener Benutzer sind und die Auswirkungen verstehen oder wenn das technische Support-Team von Red Lion Sie dazu aufgefordert hat. Die Standardeinstellungen dürften für nahezu alle Anwendungen geeignet sein.

13.3 Hinzufügen von Webseiten

Zusätzlich zu den oben beschriebenen Merkmalen unterstützt der Webserver das Anzeigen generischer Webseiten, die jeweils eine vordefinierte Liste von Tag-Werten enthalten. Diese Seiten werden wie gewohnt im Navigationsfenster erstellt. Jede Webseite hat die folgenden Eigenschaften:



- Die Eigenschaft *Title* (Titel) dient der Identifizierung der Webseite in dem Menü, das dem Benutzer über den Webbrowser angezeigt wird. Obwohl der Titel übersetzbar ist, verwendet Crimson 3.1 nur die US-amerikanische Version des Textes.
- Mit der Eigenschaft *Refresh* (Aktualisieren) wird angegeben, ob die Webseite aktualisiert wird, sobald neue Daten eingehen. In früheren Versionen wurde die ganze Seite aktualisiert, doch Crimson 3.1 verwendet AJAX, um nur die Datenwerte zu aktualisieren. Die Aktualisierung erfolgt daher ohne Flackern, und es muss keine Aktualisierungshäufigkeit angegeben werden.
- Mit der Eigenschaft *Use Colors* (Farben verwenden) wird angegeben, ob Farben, die durch die Färbung eines Tags definiert sind, beim Rendern dieser Seite verwendet werden sollen. Wenn diese Option aktiviert ist, ändert sich die im Webbrowser angezeigte Farbe je nach Tag-Status. Weitere Informationen finden Sie im Kapitel „Verwenden von Daten-Tags“.
- Mit der Eigenschaft *Allow Editing* (Bearbeiten erlauben) wird die Bearbeitung von Daten-Tags über diese Seite aktiviert. Wenn diese Option aktiviert ist, wird für jeden Datenwert eine Schaltfläche „Edit“ (Bearbeiten) angezeigt, damit der Benutzer diesen Wert ändern kann. Wenn für das Tag Sicherheits-einstellungen definiert sind, muss der auf dem Webserver angemeldete Benutzer über ausreichende Rechte zum Ändern des Tags verfügen. Wenn Sie diese Funktion aktivieren, wird empfohlen, die Authentifizierung zu verwenden.
- Die Eigenschaft *Fetch Data* (Daten vorladen) gibt an, ob die Tags für diese Seite immer in der Kommunikationsprüfung platziert werden sollen, um die Seite mit minimaler Verzögerung anzuzeigen. Wenn sie deaktiviert ist, liest Crimson 3.1 die Tags, bevor die Seite an den Browser gesendet wird, und wartet maximal bis zum Ende der von der Eigenschaft *Maximum Wait* (Maximale Wartezeit) angegebenen Zeitspanne.
- Mit der Eigenschaft *Hide Page* (Seite ausblenden) können Sie eine Seite aus den Standardmenüs ausblenden, die vom Webserver angezeigt werden. Sie wird zum Erstellen von Webseiten verwendet, die über die im nächsten Kapitel beschriebenen AJAX-Mechanismen aufgerufen werden können, um Ihre eigene benutzerdefinierte Webseite mit Tag-Daten in Echtzeit bereitzustellen. Die Eigenschaften „Access Control“ (Zugangssteuerung) und „Allow Editing“ (Bearbeiten erlauben) werden verwendet, um zu entscheiden, ob je nach Identität des aktuell angemeldeten Benutzers Lese- oder Schreibzugriff bereitgestellt wird.
- Mit der Eigenschaft *Access Control* (Zugangssteuerung) kann der Zugriff auf diese Seite mithilfe einer Sicherheitsbeschreibung auf bestimmte Benutzer beschränkt werden. Die Eigenschaft ist nur verfü-

bar, wenn die Sicherheit aktiviert ist und die Crimson 3.1-Benutzerdatenbank für die Authentifizierung verwendet wird.

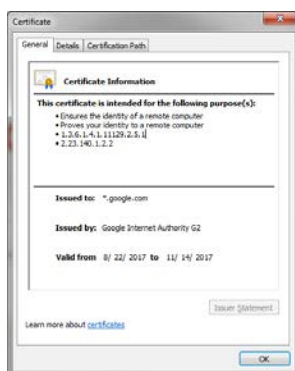
- Mit der Eigenschaft *Contents* (Inhalte) können Sie angeben, welche Tags in die Seite aufgenommen werden sollen. Tags können aus dem Ressourcenfenster in die Liste gezogen und innerhalb der Liste mithilfe von Drag-and-Drop nach oben und unten verschoben werden.

13.4 Arbeiten mit Zertifikaten

Wenn Sie bereits alles über Zertifikate wissen, fahren Sie mit [Section 13.4.2.1](#) fort, wo Sie die verschiedenen Möglichkeiten zur Erlangung eines Zertifikats kennen lernen, oder mit [Section 13.4.2.4](#), wo Sie erfahren, wie Sie schnell loslegen können.

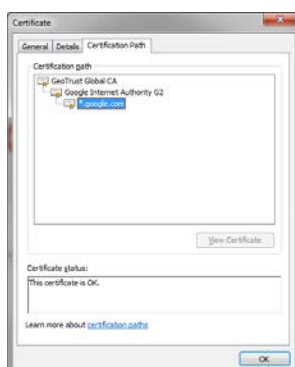
13.4.1 Einführung

Webserver, die HTTPS verwenden, müssen ein Zertifikat bereitstellen, das bestätigt, dass der Server tatsächlich Eigentum des Unternehmens ist, auf das die URL verweist. Wenn Sie beispielsweise über HTTPS auf Google zugreifen, stellt dessen Server Ihrem Browser folgendes Zertifikat bereit:



Hier sind zwei Punkte wichtig: Erstens ist das Zertifikat dafür vorgesehen, die Identität eines Remotecomputers zu gewährleisten, und zweitens gilt es für Standorte mit der Domänenadresse `*.google.com`. Ihr Webbrowser vergleicht diesen Namen mit der Adresse, die Sie in die Adressleiste eingegeben haben. Wenn die Adressen übereinstimmen, dürfen Sie diese Site besuchen.

Um die Gültigkeit eines Zertifikats zu gewährleisten, muss es von einem anderen Zertifikat signiert sein. Dieses Zertifikat muss wiederum von einem anderen Zertifikat signiert sein usw., bis ein Satz an Zertifikaten erreicht ist, bei denen Ihr Browser sicher sein kann, dass sie von vertrauenswürdigen Organisationen ausgestellt wurden, die nur Zertifikate an Personen aushändigen, die zur Nutzung der entsprechenden Namen berechtigt sind. Zum Beispiel hat das oben genannte Google-Zertifikat den folgenden Zertifizierungspfad:



Für das Zertifikat, auf das Ihr Browser sich stützt, bürgt die Google Internet Authority, die vermutlich dem Google-Konzern angehört und die für die Ausgabe von Zertifikaten an andere Teile des Unternehmens zuständig ist. Doch Ihr Browser weiß nicht wirklich, was es mit dieser Internet Authority (Internetbehörde) auf sich hat und ob sie ihre Zertifikate wirklich nur für gültige Google-Domänen ausstellt. Stattdessen verlässt Ihr Browser sich darauf, dass dieses Zertifikat wiederum von GeoTrust signiert wurde. GeoTrust ist ein auf die Ausgabe vertrauenswürdiger Zertifikate spezialisiertes Unternehmen, das bekanntlich durch geeignete Verfahren dafür sorgt, dass ausschließlich Google auf ein Zertifikat mit einem Namen zugreifen kann, der auf eine google.com-Domäne verweist. Das GeoTrust-Zertifikat ist eines der so genannten vertrauenswürdigen Stammzertifikate Ihres Browsers. Diese Vertrauenskette wird verwendet, um zu bestätigen, dass Sie tatsächlich mit Google sprechen und nicht mit einem Hacker, der Ihre Identität zu stehlen versucht.

13.4.2 Verwenden von Zertifikaten

All dies bedeutet, dass Sie, wenn Sie HTTPS verwenden, ein Zertifikat angeben müssen, das mit dem Namen übereinstimmt, der für den Zugriff auf den Server verwendet wird. Es muss möglich sein, dieses Zertifikat auf eines der vertrauenswürdigen Stammzertifikate zurück zu verfolgen, die von den entsprechenden Webbrowsern verwendet werden. Es gibt mehrere Möglichkeiten, dies zu erreichen. Wenn Sie Tests durchführen und einfach eine schnelle Antwort benötigen, fahren Sie fort mit [Section 13.4.2.4](#), damit Sie schneller loslegen können. Andernfalls werden die verschiedenen Optionen in den folgenden Abschnitten im Detail erläutert.

13.4.2.1 Beziehen eines kommerziellen Zertifikats

Idealerweise sollten Sie sicherstellen, dass der Zugriff auf Ihr Zielgerät über einen DNS-Namen erfolgt, der einer Domäne angehört, die Ihrem Unternehmen gehört. Zum Beispiel könnten Sie einen DNS-Eintrag namens `hmi01.hmis.mycom.com` erstellen, der sich auf ein bestimmtes HMI bezieht. Wenn Ihr Unternehmen nachweisen kann, dass es der Eigentümer der Domäne `mycom.com` ist, wird einer der vertrauenswürdigen Zertifikatanbieter bereit sein, ein Zertifikat auszustellen, das entweder diesen bestimmten Namen oder vielleicht auch Namen mit dem Suffix `hmis.mycom.com` bestätigt. Da diese Zertifikate auf eines der vertrauenswürdigen Stammverzeichnisse zurückverfolgbar sein werden, die auf nahezu jedem Gerät der Welt installiert sind, kann der Server von jedem Client validiert werden. Der Nachteil ist, dass Sie mit Ihrer IT-Abteilung (und vielleicht auch Ihrer Rechtsabteilung) zusammenarbeiten und bei der Ausstellung und Verlängerung eines Zertifikats Gebühren entrichten müssen. Zu den kommerziellen Zertifikatanbietern gehören GeoTrust, Thawte, GoDaddy und Comodo.

13.4.2.2 Nutzen einer lokalen Zertifizierungsstelle

Wenn Sie kein öffentliches Zertifikat möchten, können Sie eventuell die Zertifizierungsstelle Ihres Unternehmens nutzen. Viele Unternehmen verfügen über eine interne Zertifizierungsstelle, die Zertifikate für Server, Workstations und Benutzer innerhalb der Organisation ausgibt. Die Zertifikate lassen sich nicht immer auf eines der vertrauenswürdigen Stammzertifikate zurückverfolgen, die Sie auf einem normalen Computer vorfinden würden. In diesem Fall wurde jeder PC in der Organisation so konfiguriert, dass das Stammzertifikat der Zertifizierungsstelle als vertrauenswürdig gilt. Wenn der Zugriff auf den Webserver nur über Geräte erfolgen soll, die zu Ihrem Unternehmen gehören und die dessen Zertifizierungsstelle vertrauen, wird diese Methode gut funktionieren. Wie zuvor müssen Sie über einen DNS-Namen auf Ihr Gerät verweisen, der der Kontrolle Ihrer Organisation unterliegt, und Sie müssen mit Ihrer IT-Abteilung zusammenarbeiten, damit das Zertifikat ausgestellt wird. Es fallen keine Gebühren an.

13.4.2.3 Verwenden eines selbst signierten Zertifikats

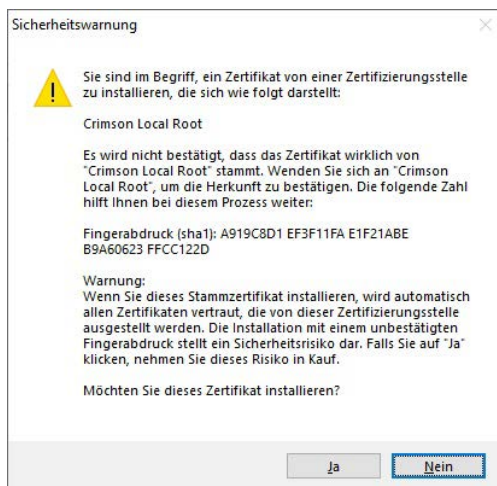
Wenn Sie keinen Zugang zu einer öffentlichen oder privaten Zertifizierungsstelle haben, können Sie ein selbst signiertes Zertifikat verwenden. Hierbei handelt es sich um ein Zertifikat, das keine Vertrauenskette aufweist und im Grunde genommen seine eigene Gültigkeit bescheinigt. Damit ein selbst signiertes Zertifikat funktioniert, muss das Zertifikat selbst auf jedem Gerät, das auf den Webserver zugreifen wird, als vertrauenswürdiges Stammzertifikat installiert werden. Dies bedeutet gelegentlich, dass es im vertrauenswürdigen Stammspeicher des Betriebssystems installiert wird. Es kann auch der eigenen Liste des Browsers hinzugefügt werden. Das ist etwas

zeitaufwändig, wenn mehrere Geräte auf den Server zugreifen oder wenn Sie nicht im Voraus wissen, welche Geräte Zugriff benötigen. Selbst signierte Zertifikate können mit Befehlszeilentools für Linux oder Windows PowerShell generiert werden. Es gibt auch mehrere Webseiten, die ein selbst signiertes Zertifikat für Sie erstellen. Unabhängig davon, welche Methode Sie wählen, müssen Sie darauf achten, dass der Name mit dem DNS-Eintrag übereinstimmt, mit dem auf den Server verwiesen wird. Außerdem muss das Zertifikat im PFX-Format vorliegen. Crimson 3.1 akzeptiert derzeit keine Zertifikate, deren Schlüssel in externen Schlüsseldateien gespeichert sind.

13.4.2.4 Verwenden des Standardzertifikats

Inzwischen dürfte Ihnen klar geworden sein, dass all dies einen recht hohen Arbeitsaufwand bedeutet, insbesondere wenn Sie lediglich einen Webserver zu Evaluierungszwecken testen möchten. Glücklicherweise bietet Crimson 3.1 eine einfachere Lösung in Form eines Standardzertifikats.

Wenn für Ihr Gerät über die Option „Zero Config“ auf der Seite „Network“ (Netzwerk) der Kategorie „Communications“ (Kommunikationen) ein lokaler Name konfiguriert wurde, generiert das Crimson 3.1-Gerät beim Starten ein Zertifikat für diesen Namen. Wenn Sie keinen Namen eingegeben haben, wird stattdessen der Standardname verwendet. Dieser Standardname lautet `red-xx-yy-zz.local`, wobei die Buchstabenpaare durch die zweite Hälfte der MAC-Adresse des Geräts ersetzt werden. Dieses automatische Zertifikat wird vom lokalen Crimson-Stammzertifikat signiert. Der oben beschriebenen Vertrauenskette gemäß müssen Sie dieses Stammzertifikat als vertrauenswürdig einstufen, damit auch die damit signierten Zertifikate als vertrauenswürdig angesehen werden können. Zur Installation des lokalen Crimson-Stammzertifikats klicken Sie auf den Link direkt unter den Zero Config-Einstellungen, woraufhin Folgendes angezeigt wird:



Wenn Sie sich davon überzeugen möchten, dass Sie das richtige Zertifikat installieren, sollte der Fingerabdruck wie folgt aussehen:

A919C8D1 EF3F11FA E1F21ABE B9A60623 FFCC122D

Sobald das lokale Crimson-Stammzertifikat installiert ist, können Sie über den lokalen Namen auf Ihr Gerät verweisen. Es wird dann kein Zertifikatfehler angezeigt. Ein Gerät namens `test` kann beispielsweise von einer beliebigen Stelle in dem selben Subnetz aufgerufen werden, indem Sie `https://test.local` in die Adressleiste Ihres Webbrowsers eingeben. Das Präfix `https://` kann bei einigen Browsern weggelassen werden, wenn „HTTP Redirect“ (HTTP-Umleitung) aktiviert ist.

Von der Bereitstellung des Standardzertifikats in Produktionsumgebungen wird abgeraten.

13.4.2.5 Jenseits des Subnetzes

Wenn Sie von außerhalb des Subnetzes über das Standardzertifikat auf ein Gerät zugreifen möchten, können Sie zwar den lokalen Namen verwenden, müssen aber der HOSTS-Datei auf dem Clientrechner einen Eintrag

hinzufügen. Beim Testen können Sie eine Eingabeaufforderung öffnen und den lokalen Namen pingen, um die von DHCP zugeordnete IP-Adresse zu ermitteln. Alternativ können Sie zur manuellen Port-Konfiguration wechseln und stattdessen eine feste IP-Adresse verwenden. Beachten Sie, dass die lokale Namensauflösung weiterhin mit manuellen Adressen funktioniert.

Angenommen, Ihr Gerät heißt `test.local` und hat die IP-Adresse 192.168.1.250.

Öffnen Sie eine Windows-Eingabeaufforderung mit Administratorrechten¹ und ändern Sie das Verzeichnis in:

```
C:\Windows\System32\drivers\etc
```

Führen Sie in diesem Verzeichnis `NOTEPAD HOSTS` aus und fügen Sie am Ende der Datei eine Zeile hinzu:

```
192.168.1.250 test.local
```

Speichern Sie die Datei. Verweise auf `test.local` werden in die erforderliche IP-Adresse aufgelöst.

13.4.2.6 Untätig bleiben

Sie könnten auch beschließen, sich die Mühe zu sparen, und einfach kein Zertifikat bereitzustellen oder keine Vertrauensbeziehung zwischen den Browsern ihrer Clients und dem Webserver herzustellen. Dies führt dann dazu, dass der Browser bei jedem Besuch der Site einen Fehler anzeigt. Viele moderne Browser werden den Zugriff auf die Webseite vollständig blockieren, es sei denn, der Benutzer ist gewillt, ein paar Umwege in Kauf zu nehmen. Eine weitere Folge ist, dass Ihre Benutzer sich mit der Zeit daran gewöhnen, Sicherheitswarnungen einfach zu ignorieren. Dies könnte sie jedoch teuer zu stehen kommen, wenn sie einmal auf einer wirklich unsicheren Site landen. Von dieser Herangehensweise wird daher dringend abgeraten.

1. Wenn Sie nicht über Administratorrechte verfügen, lesen Sie die Kommentare oben bezüglich der Zusammenarbeit mit Ihrer IT-Abteilung und wenden Sie sich an Ihren Systemadministrator.

13.5 Verwenden einer benutzerdefinierten Webseite

Die Standardwebseiten mögen zwar einen schnellen und einfachen Zugriff auf die Daten im Terminal ermöglichen, doch vielleicht empfinden Sie es als frustrierend, dass Sie keinen Einfluss auf die Formatierung haben und Ihre künstlerischen Fähigkeiten ungenutzt bleiben. Bevor Sie sich jedoch daran machen, eine Site ganz nach Ihren Vorstellungen zu entwerfen, sollten Sie überlegen, ob die benutzerdefinierte Formatvorlage und die JavaScript-Optionen, die bereits erörtert wurden, womöglich Ihren Anforderungen genügen, insbesondere wenn Sie nur Formate und Farben anpassen möchten.

Wenn Sie mehr als einfache Formatänderungen benötigen, haben Sie dank der Crimson 3.1-Funktion „Custom Site“ (Benutzerdefinierte Site) die Möglichkeit, mit dem HTML-Editor Ihrer Wahl eine ganz individuelle Webseite zu erstellen. Sie können dem HTML-Code spezielle Befehle hinzufügen, um die verschiedenen Fragmente der Standardwebseite oder aber Live-Daten aus Crimson 3.1-Tags einzuschließen. Nachdem die Dateien erstellt wurden, können sie auf der Speicherkarte Ihres Geräts im Unterverzeichnis WEB gespeichert werden. Dann ist es möglich, über den Webserver unter dem Pfad /custom auf die Dateien zuzugreifen.

Das nächste Kapitel enthält ausführlichere Informationen zu diesem Prozess.

Kapitel 14 Erstellen benutzerdefinierter Webseiten

Dieses Kapitel enthält Informationen zum Erstellen einer benutzerdefinierten Webseite, die von einem Crimson® 3.1-Gerät gehostet werden soll. Es werden HTML- und JavaScript-Kenntnisse vorausgesetzt. Um eine Webseite zu erstellen, die gut in die Crimson-Standardwebseite integriert ist, wären auch Grundkenntnisse von Bootstrap und JQuery ein Vorteil.

14.1 Namenskonventionen

In früheren Versionen von Crimson waren Dateinamen auf das 8.3-Format beschränkt. Eine mit FAT32 formatierte Speicherkarte ermöglicht Crimson 3.1, längere Namen und Erweiterungen zu verwenden. Da das Ablagesystem weiterhin nicht zwischen Groß- und Kleinschreibung unterscheidet, dürfen Sie sich nicht auf die Groß- und Kleinschreibung verlassen, um Seiten voneinander zu unterscheiden. Abgesehen davon können Sie eine Verzeichnisstruktur mit beliebig tiefer Verschachtelung verwenden, sofern alles unter dem Verzeichnis `WEB` abgelegt ist. Dem Webserver von Crimson ist nahezu jede gängige Dateierweiterung bekannt. Er antwortet mit dem richtigen MIME-Typ, wenn eine Datei angefordert wird. Benutzerdefinierte MIME-Typen werden derzeit nicht unterstützt.

14.2 Ressourcen

Die Standardwebseiten von Crimson stützen sich auf Bootstrap und JQuery, um weite Teile ihrer Funktionalität zu implementieren. Jede Webseite enthält daher die folgenden Formatvorlagen und Skriptdateien:

```
<link href="/assets/css/bootstrap.min.css" rel="stylesheet">
<link href="/assets/css/bootstrap-theme.min.css" rel="stylesheet">
<link href="/assets/css/ie10-viewport-bug-workaround.css" rel="stylesheet">
<link href="/assets/css/theme.css" rel="stylesheet">
<link href="/custom/css/custom.css" rel="stylesheet">
<!--[if lt IE 9]>
<script type="text/javascript" src="/assets/js/html5shiv.min.js"></script>
<script type="text/javascript" src="/assets/js/respond.min.js"></script>
<![endif]-->
<script type="text/javascript" src="/assets/js/jquery.min.js"></script>
<script type="text/javascript" src="/assets/js/bootstrap.min.js"></script>
<script type="text/javascript" src="/assets/js/ie10-viewport-bug-workaround.js"></script>
<script type="text/javascript" src="/assets/js/session.js"></script>
<script type="text/javascript" src="/custom/js/custom.js"></script>
```

Sie können so viele dieser Ressourcen verwenden, wie Sie benötigen, um Ihre Site zu implementieren, und dabei die oben stehenden URLs verwenden. Wir empfehlen, die Crimson-eigenen Versionen der Bootstrap- und JQuery-Bibliotheken zu verwenden, damit Sie sie Ihrer Site nicht manuell hinzufügen müssen. Die aktuellen Versionen von Bootstrap und JQuery sind 3.3.7 bzw. 1.12.4. Wenn Crimson später zu neueren Versionen wechselt, die Kompatibilitätsprobleme verursachen, wird eine Eigenschaft hinzugefügt, mit der Sie wählen können, welche Versionen Sie für Ihre benutzerdefinierte Webseite verwenden möchten.

Das Skript `session.js` erfüllt eine besondere Funktion. Es schreibt alle URLs neu, auf die von `<a>`-Elementen verwiesen wird, so dass die an die Seite übergebene Sitzungs-ID in ihre eigene URL aufgenommen wird. Crimson und Ihr Browser können dann beim Zwischenspeichern von Seiten zwischen den einzelnen Sitzungen unterscheiden. Dies ist wichtig, da unterschiedliche Benutzer je nach ihren Sicherheitseinstellungen unterschiedliche Benutzeroberflächenoptionen haben und die Zwischenspeicherung möglicherweise nicht richtig auf diese Änderungen reagiert, es sei denn, der Sitzungswert ist in der URL enthalten. Wenn Sie sich auf diese Funktion verlassen, sollten Sie unbedingt `session.js` sowie Folgendes nach dem schließenden `</body>`-Tag einfügen:

```
<script type="text/javascript">$(document).ready(doSession);</script>
```

Die Funktion `doSession` ist auch für den Aufruf von `doCustom` in einer benutzerdefinierten JavaScript-Datei sowie für den Aufruf der Standardfunktionen zuständig, die für den Fernzugriff und die Datenanimation verwendet werden.

14.3 Serverbefehle

Der Crimson-Webserver bietet keine Unterstützung für CGI, ASP, ASPX oder andere serverseitige Skripterstellungsfunktionen. Es werden jedoch serverseitige Befehle bereitgestellt, um anzugeben, dass eine Seite nicht zwischengespeichert werden soll, und um eingebettete Tag- oder Systemdaten in eine Seite aufzunehmen.

14.3.1 Servere Enthalten

Ein serverseitiges Include kann durch Einfügen des folgenden HTML-Codes durchgeführt werden:

```
<%Include filename%>
```

Der Teil `filename` muss durch den URL-ähnlichen Pfad der erforderlichen Datei ersetzt werden. Beachten Sie, dass der URL-ähnliche Pfad für benutzerdefinierte Webseite-Dateien mit `/custom` beginnt und nicht mit dem Präfix `\WEB` für das Verzeichnis, in dem die Dateien gespeichert sind.

Zwei Standarddateien werden bereitgestellt und sind häufig in benutzerdefinierten Seiten enthalten. Die Datei `/stdhead.htm` enthält den gesamten HTML-Code, der zum Laden der oben beschriebenen Standardbibliotheken und Formatvorlagen erforderlich ist. Sie kann dem Abschnitt `<head>` jeder Seite hinzugefügt werden, um die Duplizierung dieses Inhalts zu vermeiden. Die Datei `/stdnavbar.htm` enthält den HTML-Code, der zur Implementierung der Menüleiste erforderlich ist, die von der Standardwebseite bereitgestellt wird. Wenn Sie möchten, dass Ihre benutzerdefinierten Seiten diese Navigationsmethode verwenden, fügen Sie dies am Anfang des Abschnitts `<body>` jeder Seite ein.

14.3.2 Steuern der Zwischenspeicherung

Die folgende Anweisung kann zum Deaktivieren der Zwischenspeicherung für eine Seite verwendet werden:

```
<%NoCache%>
```

Die Zwischenspeicherung muss in der Regel deaktiviert werden, wenn eine Seite dynamische Daten enthält.

14.3.3 Eingebettete Daten

Daten können mit folgender Syntax in einer Seite eingebettet werden:

```
<%=class.name%>
```

14.3.3.1 Tagdaten

Für den `class`-Wert von `Tags` werden die folgenden `name`-Werte unterstützt:

Name	Ergebnis
TagName	Der formatierte Wert des angegebenen Tags. Beispielsweise fügt <code>Tags.Tag1</code> den Wert von <code>Tag1</code> ein. Auf den Namen kann ein Array-Index in eckigen Klammern folgen, der den Zugriff auf ein bestimmtes Element ermöglicht. Beispielsweise wird mit <code>Tags.Tag2[10]</code> das elfte Element (denken Sie mal darüber nach) von <code>Tag2</code> angezeigt.

Zusätzlich zur Syntax `<%=Tags.Name%>` können Sie einen Tag-Wert einfügen, indem Sie die Legacy-Sequenz `[[n]]` verwenden und `n` durch die Indexnummer oder den Namen des betreffenden Tags ersetzen. Die Indexnummer eines Tags wird rechts oben im Bearbeitungsfenster angezeigt, wenn ein Tag in der Kategorie „Data Tags“ (Daten-Tags) ausgewählt ist. Sie entspricht mehr oder weniger der Reihenfolge, in der die Tags erstellt wurden. Indexnummern bieten zwar einen etwas schnelleren Zugriff auf Tag-Daten als Namen, dennoch wird die neuere Syntax bevorzugt.

14.3.3.2 Globale Daten

Für einen `class`-Wert von `Global` werden die folgenden `name`-Werte unterstützt:

Name	Ergebnis
Home	Die Eigenschaft <i>Home Label</i> (Startseitenbeschriftung) des Webservers.
Titel	Die Eigenschaft <i>Main Title</i> (Haupttitel) des Webservers.
Header	Die Eigenschaft <i>Main Header</i> (Hauptkopfzeile) des Webservers.
MainNavBar	Der Code zum Implementieren der Hauptnavigationsleiste oben auf jeder Seite. Dieser Wert wird aufgerufen, wenn <code>stdnavbar.htm</code> enthalten ist. Platzhalter-Spannen mit den IDs <code>c3navbar-head</code> , <code>c3navbar-home</code> und <code>c3navbar-tail</code> werden eingeschlossen, um die Manipulation der Navigationsstruktur mit JQuery zu ermöglichen. Weitere Informationen finden Sie in den Ausführungen zur Datei <code>sample.js</code> im vorherigen Kapitel.
MainMenu	Der Code zum Implementieren des Hauptmenüs, das auf der Standard-Startseite angezeigt wird. Sie können ihn in einer benutzerdefinierten Startseite verwenden, um eine Replikation zu vermeiden. Platzhalterzeilen mit den IDs <code>c3-menu-head</code> und <code>c3-menu-tail</code> werden bereitgestellt, um die Manipulation der Menüstruktur mit JQuery zu ermöglichen. Weitere Informationen finden Sie in den Ausführungen zur Datei <code>sample.js</code> im vorherigen Kapitel.

14.3.3.3 Datenindex-Daten

Für den `class`-Wert `DataIndex` werden die folgenden `name`-Werte unterstützt:

Name	Ergebnis
Dropdown	Der Code, der zum Implementieren der Dropdown-Liste der Standardwebseiten verwendet wird, die über das Konfigurationstool von Crimson erstellt wurden, wobei nach einer Änderung nur die Seiten einbezogen werden, auf die Benutzer zugreifen können.
List	Der Code, der zum Implementieren der menüartigen Liste der Standardwebseiten verwendet wird, die über das Konfigurationstool von Crimson erstellt wurden, wobei nach einer Änderung nur die Seiten eingezogen werden, auf die Benutzer zugreifen können.

14.3.3.4 Datenansichts-Daten

Für den `class`-Wert `DataView` werden die folgenden `name`-Werte unterstützt:

Name	Ergebnis
Header	Der Code zum Erstellen der Kopfzeile, der oben auf einer bestimmten Standardwebseite angezeigt wird. Die Seitenzahl muss in der URL als Parameter <code>page</code> enthalten sein.
TagHead	Der Code zum Erstellen der Kopfzeile der Tabelle, die oben auf einer bestimmten Standardwebseite angezeigt wird. Die Seitenzahl muss in der URL als Parameter <code>page</code> enthalten sein.
TagList	Der Code zum Erstellen des Inhalts der Tabelle, die oben auf einer bestimmten Standardwebseite angezeigt wird. Die Seitenzahl muss in der URL als Parameter <code>page</code> enthalten sein.

14.4 AJAX-Updates

14.4.1 Lesen von Tags

Die Standardwebseiten, die Tag-Daten enthalten, werden durch eine GET-Anforderung an die Ressource `/ajax/dataview-read.htm` aktualisiert. Die URL muss die erforderliche Seitenzahl als Parameter `page` enthalten, wobei der Wert als Dezimalwert codiert wird. Die Seitenzahlen beginnen bei Null und entsprechen im Wesentlichen der Reihenfolge, in der die Standardseiten erstellt wurden. Sie sind nicht über die Benutzeroberfläche von Crimson zugänglich, können jedoch leicht von den Verknüpfungen abgeleitet werden, die von der Standardwebseite zum Referenzieren dieser Seiten verwendet werden. Sie können diese URL verwenden, um die gleichen Daten für Ihre eigenen Zwecke zu erhalten, vielleicht um eine benutzerdefinierte Seite zu animieren, die ebenfalls Tags enthält. Die Anforderung gibt eine Textdatei mit einer Zeile für jedes Tag zurück. Jede Zeile enthält den formatierten Wert des Tags, optional gefolgt von den Vorder- und Hintergrundfarben des Tags, wenn diese für die betreffende Seite aktiviert sind. Wenn Sie eine Standardwebseite nur für diesen Zweck erstellen, stellen Sie in der Regel die Eigenschaft *Hide Page* (Seiten ausblenden) ein, um zu verhindern, dass sie in den Standardmenüs angezeigt wird.

14.4.2 Schreiben von Tags

Die Standardwebseiten, die Tag-Daten enthalten, führen Tag-Schreibvorgänge über eine GET-Anforderung an die Ressource `/ajax/dataview-write.htm` aus. Wie oben muss diese URL die erforderliche Seitenzahl als Parameter `page` enthalten, aber zusätzlich das erforderliche Tag als Parameter `tag` angeben. Die Tag-Nummer ist ein auf Null basierender Index für die Liste der Tags, die über die Eigenschaft „Contents“ (Inhalte) für diese Seite konfiguriert sind. Das Tag wird auf den Wert festgelegt, der im Parameter `data` angegeben wird, der ebenfalls in der URL enthalten sein muss. Ein erfolgreicher Schreibvorgang erzeugt den Rückgabecode 200 mit einer Meldung im Antworttext, die die Änderung bestätigt. Ein Schreibvorgang, der keine Änderung bewirkt, hat zwar denselben Rückgabecode, aber einen leeren Textteil. Der Rückgabecode 403 bedeutet, dass der Schreibvorgang aufgrund eines Sicherheitsproblems oder ungültiger Daten fehlgeschlagen ist. Der Vorgang schlägt auch fehl, wenn die betreffende Seite nicht so konfiguriert wurde, dass Schreibvorgänge möglich sind. Noch einmal: Wenn Sie eine Standardwebseite nur für diesen Zweck erstellen, stellen Sie in der Regel die Eigenschaft *Hide Page* (Seiten ausblenden) ein, um zu verhindern, dass sie in den Standardmenüs angezeigt wird.

14.5 Bereitstellung von Sites

Zur Bereitstellung Ihrer benutzerdefinierten Webseite kopieren Sie diese in das Verzeichnis `\WEB` auf der Speicherkarte, die im Zielgerät installiert werden soll. Zum Kopieren der Dateien melden Sie entweder die Karte als Laufwerk an, wie in früheren Kapiteln dieses Handbuchs beschrieben; verwenden einen geeigneten Kartenschreiber, der mit Ihrem PC verbunden ist; oder übertragen die Dateien über den FTP-Server. Aktivieren Sie die benutzerdefinierte Site in den Webserver-Eigenschaften. Falls Sie die benutzerdefinierte Site in das Menü eingefügt haben, erscheint die Site sowohl in der Menüleiste als auch in der Hauptmenüliste. Wenn die Site ausgewählt wird, wird im Verzeichnis `\WEB` eine Datei namens `default.htm` angezeigt. Nach diesem Punkt erfolgt die Navigation mithilfe der Verknüpfungen auf der Seite.

Kapitel 15 Verwenden des Sicherheitssystems

Crimson® 3.1 enthält leistungsstarke Funktionen, mit denen Sie festlegen können, welche Bediener auf welche Anzeigeseiten und andere Funktionen Zugriff haben und welche Bediener Änderungen an bestimmten Daten vornehmen dürfen. Die Software enthält außerdem eine Protokollierungsfunktion, mit der Änderungen an Datenwerten sowie der Zeitpunkt und Verursacher dieser Änderungen aufgezeichnet werden können.

15.1 Grundsätzliches zum Thema Sicherheit

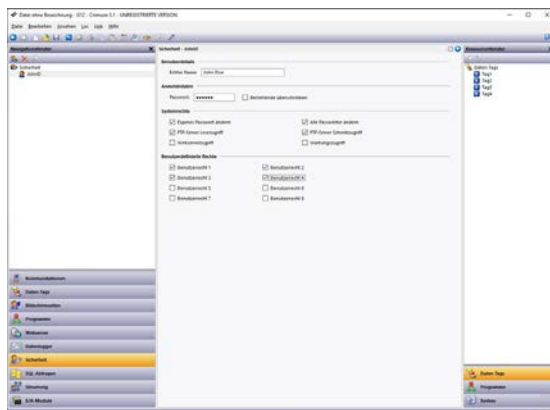
In den folgenden Abschnitten werden einige der grundlegenden Konzepte beschrieben, auf denen das Sicherheitssystem beruht.

15.1.1 Objektbasierte Sicherheit

Das Sicherheitssystem von Crimson ist objektbasiert. Dies bedeutet, dass Sicherheitseigenschaften auf eine Anzeigeseite oder ein Tag angewendet werden und nicht auf das Benutzeroberflächenelement, das auf die Seite zugreift oder eine Änderung an dem Tag vornimmt. Das Gegenstück hierzu ist der subjektbasierte Ansatz. Bei diesem Ansatz müssen Sie in der Regel darauf achten, Sicherheitseinstellungen auf jedes einzelne Benutzeroberflächenelement anzuwenden, das zugangsbeschränkte Daten ändern könnte. Der Ansatz von Crimson vermeidet diese Duplizierung und gewährleistet, dass ein Tag, zu dessen Schutz Sie sich einmal entschieden haben, in Ihrer gesamten Datenbank geschützt bleibt.

15.1.2 Benannte Benutzer

Crimson unterstützt die Fähigkeit, eine beliebige Anzahl von Benutzern zu erstellen, die jeweils über einen Benutzernamen, einen echten Namen und ein Kennwort verfügen. Bei dem Benutzernamen handelt es sich um eine Zeichenfolge ohne Leerzeichen, bei der die Groß-/Kleinschreibung nicht beachtet wird und die zum Identifizieren des Benutzers bei der Anmeldung dient. Der echte Name ist in der Regel eine längere Zeichenfolge, die in Protokolldateien verwendet wird, um die Identität eines Benutzers, der Änderungen vornimmt, in von Menschen lesbarem Format aufzuzeichnen. Beachten Sie, dass Sie diese Felder auch anders einsetzen können, wenn dies für Ihre Anwendung sinnvoll ist: Sie können z. B. Benutzer erstellen, die Gruppen von Einzelpersonen oder vielleicht Rollen darstellen, z. B. Bediener, Vorgesetzte und Manager.



15.1.3 Benutzerrechte

Jeder Benutzer erhält keines bis mehrere Zugriffsrechte. Ein Benutzer ohne Rechte kann auf Objekte zugreifen, bei denen lediglich die Identität des Benutzers erfasst werden muss. Benutzer mit mehreren Rechten jedoch können auf Objekte zugreifen, bei denen das Vorhandensein dieser Rechte vorausgesetzt wird. Rechte sind in Systemrechte und Benutzerrechte unterteilt. Systemrechte steuern den Zugriff auf Funktionen in der Crimson-Software. Benutzerrechte sind für allgemeine Verwendungszwecke vorgesehen. Zum Beispiel könnte Benutzerrecht 1 in Ihrer Datenbank verwendet werden, um den Zugriff auf Produktionsziele zu beschränken. Dieses Recht würde nur jenen Benutzern erteilt, denen Sie gestatten möchten, derartige Dinge zu ändern.

15.1.4 Zugangssteuerung

Für Objekte, die Sicherheitsmaßnahmen unterliegen, gibt es eine entsprechende Eigenschaft namens *Access Control* (Zugangssteuerung).

Bei der Bearbeitung dieser Eigenschaft wird Folgendes angezeigt:

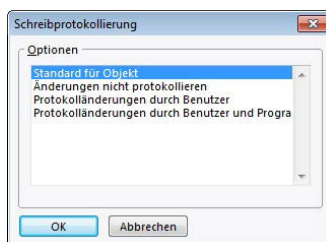


Mit diesen Einstellungen können Sie festlegen, wer auf das Element zugreifen darf: jeder Benutzer; jeder Bediener, dessen Identität bekannt ist; oder Benutzer mit bestimmten Benutzerrechten. Sie können auch angeben, ob ein Tag von einem Programm geändert werden darf, das aus einem anderen Grund als wegen einer Benutzeraktion ausgeführt wird. Mit dieser Funktion können Sie gewährleisten, dass keine Änderungen an vertraulichen Daten im Hintergrund vorgenommen werden, selbst wenn ein Programmierfehler versucht, eine solche Änderung vorzunehmen.

15.1.5 Protokollierung von Schreibvorgängen

Tags verfügen auch über eine Eigenschaft namens *Write Logging* (Schreibprotokollierung).

Bei der Bearbeitung dieser Eigenschaft wird Folgendes angezeigt:



Durch Ihre Auswahl geben Sie an, ob Änderungen, die von Benutzern oder Programmen an Tags vorgenommen werden, protokolliert werden sollen. Diese Funktion ermöglicht Ihnen, Änderungen an Ihrem System in einem Audit-Trail zu erfassen. Dies wiederum erleichtert das Auffinden von Fehlern und das Bereitstellen von Details zur Qualitätskontrolle bei der Prozesskonfiguration. Lassen Sie bei der Protokollierung von Änderungen, die von

Programmen vorgenommen werden, Vorsicht walten. Manche Datenbanken protokollieren in diesem Zusammenhang unüberschaubare Datenmengen.

15.1.6 Standardzugriff

Um den Konfigurationsprozess zu beschleunigen, bietet Crimson die Möglichkeit, für zugeordnete Tags, interne Tags und Anzeigeseiten Parameter für den Standardzugriff und die Schreibprotokollierung festzulegen. Die Unterscheidung zwischen zugeordneten und nicht zugeordneten Tags ist wichtig bei Systemen, bei denen alle Änderungen an externen Daten aufgezeichnet werden müssen, bei denen Crimson-interne Daten aber ohne die Notwendigkeit eines solchen Audit-Trails manipuliert werden können.

15.1.7 Anmeldung auf Anforderung

Das Sicherheitssystem von Crimson unterstützt sowohl konventionelle Anmeldungen als auch Anmeldungen auf Anforderung. Eine konventionelle Anmeldung kann auftreten, wenn über ein Benutzeroberflächenelement wie eine Drucktaste die Aktion „Log On User“ (Benutzer anmelden) aktiviert oder die Funktion `UserLogOn()` aufgerufen wird. Eine Anmeldung auf Anforderung findet statt, wenn der Bediener versucht, eine Aktion auszuführen, ohne über ausreichende Zugriffsrechte zu verfügen, und wenn es innerhalb derselben Aktion nicht zu einem fehlgeschlagenen Anmeldeversuch gekommen ist. Angenommen, ein Benutzer drückt eine Taste, die ein Programm ausführt, das eine Reihe von Werten zurücksetzt. Sobald das Programm versucht, einen Wert zu ändern, für den ein Sicherheitszugriff erforderlich ist, fordert das System zur Eingabe der Anmeldeinformationen auf. Diese Methode schränkt die Interaktion des Bedieners ein und führt zu einem reaktionsschnelleren System.

15.1.8 Wartungszugriff

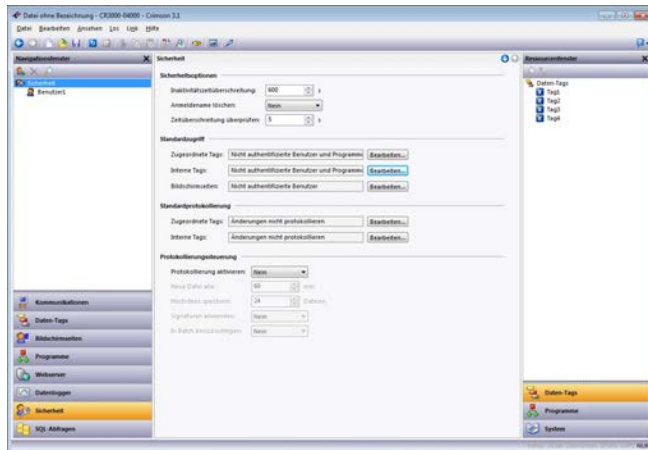
Das System bietet außerdem ein Funktion namens „Maintenance Mode“ (Wartungsmodus), mit dem das Timeout wegen Benutzerinaktivität während der Inbetriebnahme des Systems außer Kraft gesetzt werden kann. Dieser Modus wird aktiviert, wenn einer Anzeigeseite das Recht „Wartungszugriff“ als Zugangsvoraussetzung zugewiesen wurde und der aktuelle Benutzer aufgrund dieses Rechts Zugriff auf die Seite erhalten hat. Die Verwendung dieses Modus verhindert, dass Sie sich beim Testen des Systems wiederholt anmelden müssen.

15.1.9 Vor Verwendung prüfen

Mit der Funktion „Check Before Operate“ (Vor Verwendung prüfen) können Sie den Benutzer zwingen, jede Änderung an einem sensiblen Datenelement zu bestätigen. Die Funktion wird aktiviert, indem Sie in der Sicherheitsbeschreibung eines Daten-Tags die entsprechende Einstellung auswählen. Wenn eine Änderung an einem Tag vorgenommen wird, für das diese Funktion aktiviert ist, wird ein Popup mit den alten und neuen Werten sowie der Aufforderung angezeigt, die Änderungen zu bestätigen. Diese Funktion ist unabhängig davon, ob gegenwärtig ein Benutzer angemeldet ist, und wird zusätzlich zu den Benutzerrechten eingesetzt, die für das Durchführen der Änderung erforderlich sind. Sie ist außerdem unabhängig von den Vorgängen, die beim Erstellen der Benutzeroberfläche zum Schutz von Aktionen definiert wurden.

15.2 Sicherheitseinstellungen

Der Zugriff auf die Sicherheitseinstellungen des Systems erfolgt über das Stammelement in der Kategorie „Security“ (Sicherheit):

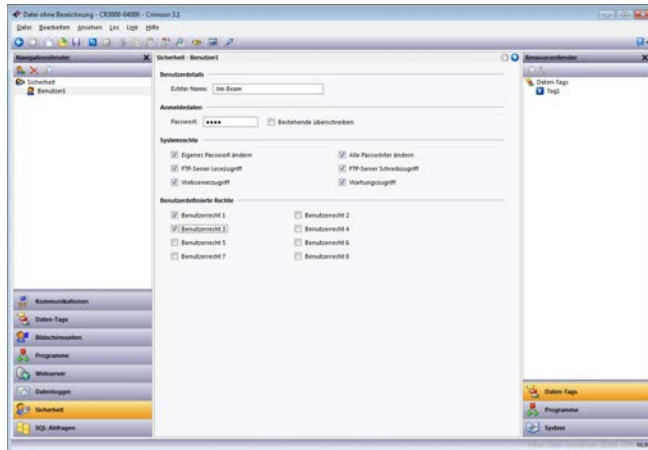


Die verfügbaren Eigenschaften lauten wie folgt:

- Mit der Eigenschaft *Inactivity Timeout* (Timeout bei Inaktivität) wird angegeben, wie viel Zeit ohne Benutzereingabe verstreichen muss, bevor der aktuelle Benutzer automatisch abgemeldet wird. Ein zu hoher Wert für diese Einstellung führt zu einem unsicheren System, während ein zu niedriger Wert zu einem System führt, das für Bediener zu umständlich ist.
- Mit der Eigenschaft *Clear Logon Name* (Anmeldename löschen) geben Sie an, ob der Benutzername gelöscht werden soll, bevor der Bediener zur Anmeldung aufgefordert wird. Wenn diese Einstellung deaktiviert ist, wird der vorherige Benutzername angezeigt, und nur das Kennwort muss erneut eingegeben werden. Die Aktivierung dieser Funktion sorgt für mehr Sicherheit und ist in manchen Branchen möglicherweise erforderlich, um die Sicherheitsstandards zu erfüllen.
- Über die Eigenschaften unter *Default Access* (Standardzugriff) wird der Zugriff auf verschiedene Objekte geregelt, falls es für ein Element kein spezieller Zugriff definiert ist. Informationen zu den Einstellungen finden Sie oben im Abschnitt „Zugangssteuerung“.
- Mit den Eigenschaften unter *Default Logging* (Standardprotokollierung) können Sie angeben, ob Änderungen an zugeordneten und nicht zugeordneten Tags protokolliert werden sollen, falls keine speziellen Protokollierungskriterien für ein Tag definiert wurden. Es ist nicht möglich, den programmgesteuerten Zugriff standardmäßig zu protokollieren, da diese Art der Protokollierung nur nach sorgfältiger Überlegung eingesetzt werden sollte, um eine übermäßige Protokollierungsaktivität zu vermeiden.
- Die Eigenschaften unter *Logging Control* (Protokollierungssteuerung) definieren, ob und wie die Sicherheitsprotokolle erstellt werden sollen. Wie die Daten geschrieben und Dateien benannt werden, erfahren Sie im Kapitel „Verwenden des Datenloggers“.

15.3 Erstellen von Benutzern

Benutzer werden in der Navigationsliste mithilfe der üblichen Methoden erstellt und bearbeitet.



Jeder Benutzer hat die folgenden Eigenschaften:

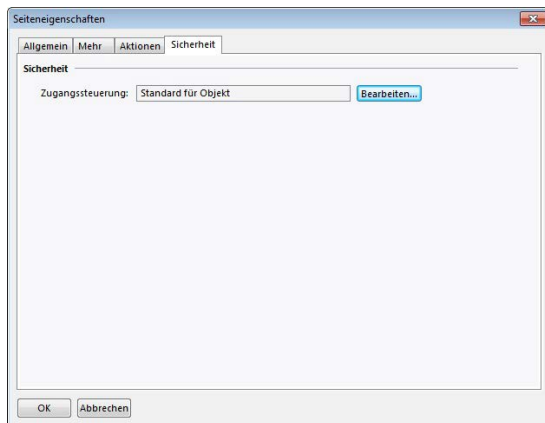
- Die Eigenschaft *Real Name* (Echter Name) dient zum Aufzeichnen der Identität des Benutzers in Sicherheitsprotokollen und wird auch im Primitiv „Sicherheitsmanager“ angezeigt, das zum Ändern von Kennwörtern zur Laufzeit verwendet wird. Wenn maximale Sicherheit erforderlich ist, sollte der Benutzername sich nicht einfach aus dem echten Namen herleiten lassen.
- Die Eigenschaft *Password* (Kennwort) gibt ein Initialkennwort für diesen Benutzer vor. Beim Kennwort wird die Groß- und Kleinschreibung beachtet und es besteht aus alphanumerischen Zeichen. Wenn das Kontrollkästchen *Override Existing* (Bestehende überschreiben) aktiviert ist, werden alle Änderungen, die vom Zielgerät aus an diesem Kennwort vorgenommen wurden, überschrieben, wenn diese Datenbank heruntergeladen wird.
- Die Eigenschaften unter *System Rights* (Systemrechte) dienen dazu, einem Benutzer die Möglichkeit zu gewähren, bestimmte Systemaktionen auszuführen. Die Eigenschaften im Zusammenhang mit Kennwortänderungen sind selbsterklärend, und Einzelheiten zum Benutzer des Wartungsmodus finden Sie oben.
- Die Eigenschaften unter *Custom Rights* (Individuelle Rechte) dienen dazu, einem Benutzer bestimmte Rechte zu erteilen, die dann in der Datenbank genutzt werden können, um Zugriff auf Gruppen von Tags oder Anzeigeseiten zu ermöglichen. Über die genaue Nutzung dieser Rechte entscheidet der Systementwickler.

15.4 Festlegen der Tag-Sicherheit

Für jedes Tag gibt es eine Registerkarte namens „Security“ (Sicherheit), auf der die Einstellungen für Zugriffssteuerung und Schreibprotokollierung für dieses Tag festgelegt werden. Wenn Sie keine speziellen Einstellungen festlegen, verwendet das System die entsprechenden Standardeinstellungen, je nachdem, ob eine Zuordnung zu externen Daten besteht.

15.5 Festlegen der Seitensicherheit

Die Einstellungen für die Zugangssteuerung für Anzeigeseiten werden über das Dialogfeld „Eigenschaften“ der Anzeigeseiten definiert:



Wenn Sie keine Einstellungen vornehmen, werden die Standardeinstellungen verwendet.

15.6 Sicherheitsbezogene Funktionen

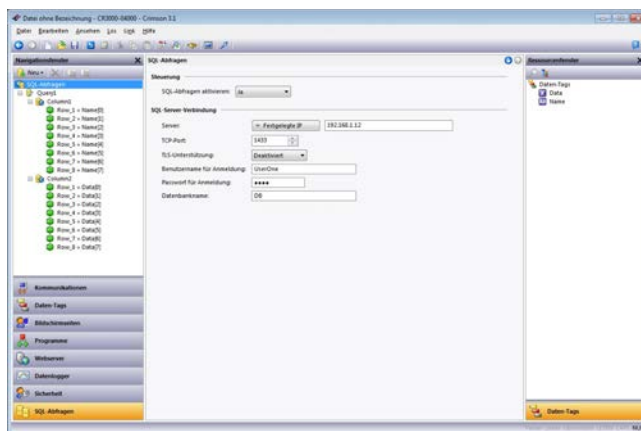
Weitere Informationen zu den Funktionen `UserLogOn()`, `UserLogOff()` und `TestAccess()` finden Sie im Referenzhandbuch. Die zuletzt genannte Funktion ist nützlich, wenn viele Werte von einem Programm aus geändert werden, denn sie gestattet Ihnen, schon früh im Code eine Zugriffsprüfung zu erzwingen und dadurch Änderungen zu vermeiden, die später wegen unzureichender Benutzerrechte zum Fehlschlagen von Vorgängen führen.

Kapitel 16 Verwenden von SQL-Abfragen

Die Kategorie „SQL Queries“ (SQL-Abfragen) wird zum Erstellen von SQL-Abfragen verwendet, mit denen Daten aus einer Microsoft SQL Server-Datenbank extrahiert und in Tags gespeichert werden können. Diese Tags können dann an anderer Stelle in der Crimson 3.1-Datenbank verwendet werden, um die Produktion zu steuern, Rezepteneinstellungen zu aktualisieren oder Produktivitätsinformationen zu dem Werk anzuzeigen, in dem das Gerät installiert ist.

16.1 Konfigurieren des Servers

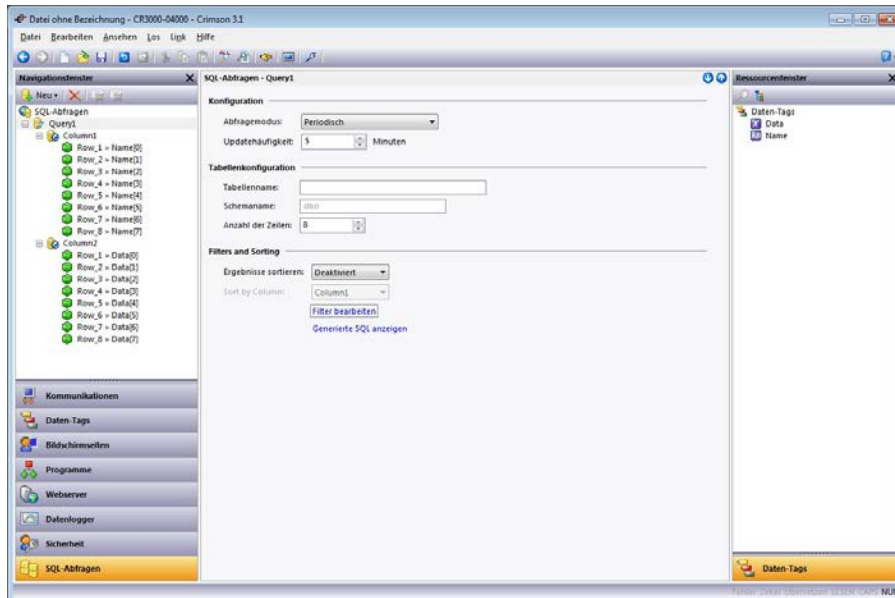
Der erste Schritt beim Zugriff auf SQL-Daten besteht darin, den Microsoft SQL Server zu konfigurieren, von dem die Daten abgerufen werden sollen. Hierzu wählen Sie den Stammeintrag in der Navigationsliste aus, die in der Kategorie „SQL Queries“ (SQL-Abfragen) angezeigt wird:



- Mit der Eigenschaft *Enable SQL Queries* (SQL-Abfragen aktivieren) wird die Funktion „SQL Queries“ (SQL-Abfragen) aktiviert.
- Die Eigenschaft *Server* (Server) wird verwendet, um die Adresse des Microsoft SQL-Servers anzugeben, mit dem das Gerät verbunden wird. Diese können Sie als IP-Adresse, als festgelegten DNS-Namen, als durch ein String-Tag bestimmten DNS-Namen oder als numerischen Ausdruck angeben.
- Mit der Eigenschaft *TCP Port* (TCP-Port) wird der TCP-Port angegeben, mit dem das Gerät verbunden wird. Dieser Port muss mit jenem Port übereinstimmen, der auf dem Server für den TCP/IP-Zugriff konfiguriert ist. Weitere Auskünfte hierzu erteilt Ihr Datenbankadministrator.
- *Logon Username* (Anmeldebenutzername) und *Logon Password* (Anmeldekennwort) sind die Anmeldedaten, die beim Herstellen der Verbindung an den Server gesendet werden. Beim Kennwort wird immer die Groß-/Kleinschreibung beachtet. Ob die Groß-/Kleinschreibung auch beim Benutzernamen beachtet wird, hängt vom Server ab. Der Benutzer muss über die erforderlichen Rechte verfügen, um Tabellendaten zu lesen.
- Mit der Eigenschaft *Database Name* (Datenbankname) wird der Name der Datenbank auf dem Server angegeben, aus der die SQL-Abfragen Daten extrahieren. Wenden Sie sich an Ihren Datenbankadministrator, um sicherzugehen, dass Sie den richtigen Namen angeben.

16.2 Erstellen von Abfragen

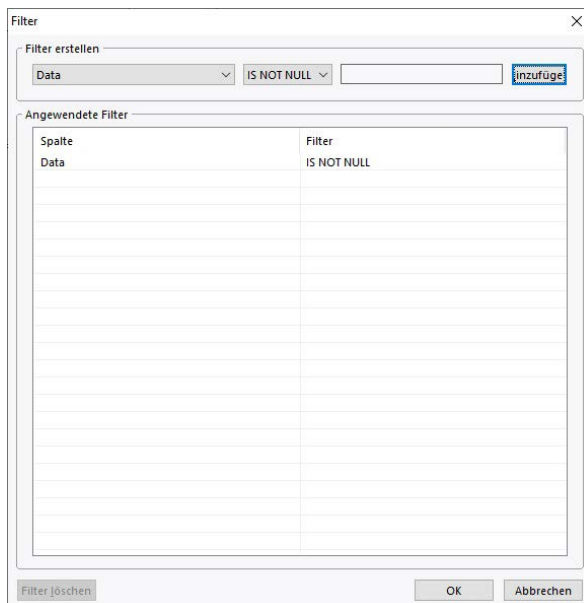
SQL-Abfragen werden auf die übliche Weise im Navigationsbereich erstellt. Jede Abfrage hat die folgenden Eigenschaften:



- Mit der Eigenschaft *Query Mode* (Abfragemodus) können Sie angeben, wie die Abfrage ausgeführt werden soll. Wenn die Eigenschaft auf „Periodic“ (Regelmäßig) eingestellt ist, wird die Abfrage in der Häufigkeit ausgeführt, die mit der Eigenschaft *Refresh Rate* (Aktualisierungshäufigkeit) angegeben wird. Wenn die Eigenschaft auf *On Demand* (Auf Anforderung) gesetzt ist, kann die Abfrage ausgeführt werden, indem Sie die Funktion `RunQuery()` aufrufen und den Namen der Abfrage als Parameter übergeben oder indem Sie die Funktion `RunAllQueries()` aufrufen. Wenn die Eigenschaft auf „Disabled“ (Deaktiviert) gesetzt ist, wird die Abfrage nicht ausgeführt.
- Mit der Eigenschaft *Table Name* (Tabellenname) wird der Name der Tabelle angegeben, die Gegenstand der Abfrage ist. Sie sollte einer geeigneten Tabelle innerhalb der SQL-Datenbank entsprechen, die beim Konfigurieren des Servers festgelegt wurde.
- Die Eigenschaft *Schema Name* (Schemaname) dient zum Überschreiben des Standardschemanamens `dbo`, wenn zum Verwalten der Tabellensichtbarkeit Schemata eingesetzt werden. Sie sollte dem Schema entsprechen, über das der angegebene Benutzer auf die angegebene Tabelle zugreift.
- Mit der Eigenschaft *Number of Rows* (Anzahl der Zeilen) wird angegeben, wie viele Zeilen von der Datenbank angefordert werden, wenn die Abfragen ausgeführt werden. Sie wird als `TOP`-Element in der `SELECT`-Abfrage an die Datenbank übergeben. Die unten stehenden Angaben unter „Filters and Sorting“ (Filter und Sortierreihenfolge) bestimmen genau, welche Zeilen zurückgegeben werden.
- Mit der Eigenschaft *Sort Results* (Ergebnisse sortieren) wird die Klausel `ORDER BY` der Abfrage definiert und somit die Sortierreihenfolge der resultierenden Zeile bestimmt. Dies wirkt sich sowohl auf die Reihenfolge aus, in der die Daten in den zugeordneten Tags gespeichert werden, als auch auf die Wirkung der `TOP`-Klausel, die durch die Eigenschaft *Number of Rows* (Anzahl der Zeilen) gesteuert wird. Wenn *Sort Results* (Ergebnisse sortieren) aktiviert ist, wird mit der Eigenschaft „Sort by Column“ (Nach Spalte sortieren) die Spalte angegeben, die als Sortierschlüssel verwendet werden soll.

16.2.1 Filtern der Ergebnisse

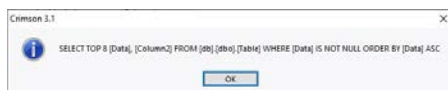
Sie können die Zeilen filtern, die von jeder Abfrage zurückgegeben werden, indem Sie die Option „Edit Filters“ (Filter bearbeiten) auswählen:



Eine Filterbedingung kann hinzugefügt werden, indem Sie im ersten Feld des Abschnitts „Create Filter“ (Filter erstellen) die erforderliche Spalte auswählen und dann den gewünschten Operator und Datenwert auswählen. Klicken Sie anschließend auf die Schaltfläche „Add“ (Hinzufügen), um die Bedingung an die Liste anzuhängen. Beachten Sie, dass Filterdatenwerte Konstanten sind und nicht auf Crimson 3.1-Tags oder andere Ausdrücke gesetzt werden können. Bedingungen können entfernt werden, indem Sie sie in der Liste „Applied Filters“ (Angewendete Filter) auswählen und auf „Delete Filter“ (Filter löschen) klicken.

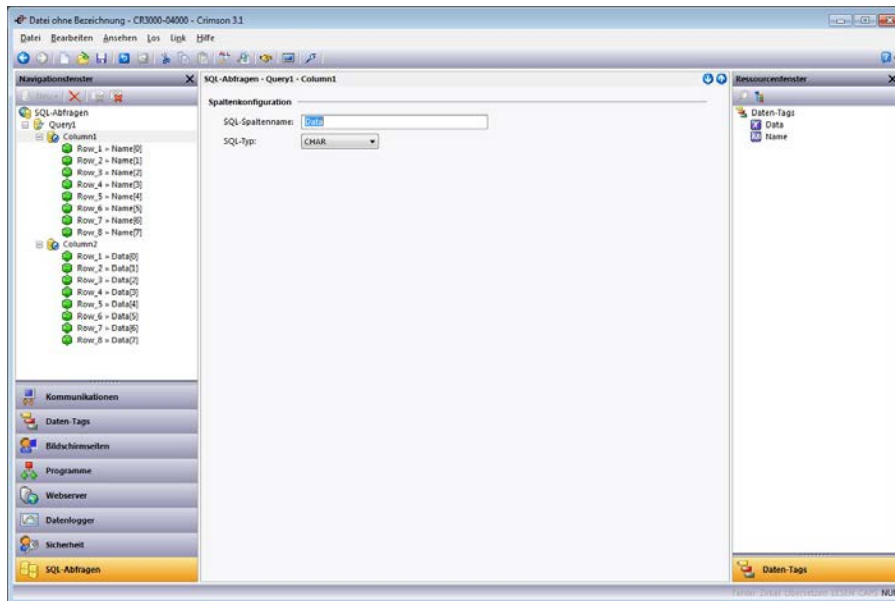
16.2.2 Prüfen des SQL-Codes

Sobald eine Abfrage definiert wurde, kann mit der Option „View Generated SQL“ (Generierte SQL anzeigen) die SQL geprüft werden, die an den Server gesendet wird. In der folgenden Abbildung sehen Sie eine Abfrage, die bis zu acht Zeilen mit zwei Spalten liest, Filter anwendet und eine der Spalten als Sortierschlüssel verwendet:



16.3 Erstellen von Spalten

Wenn eine Abfrage erstellt wird, legt Crimson 3.1 im Navigationsfenster eine einzelne Spalte an, um die zurückgegebenen Daten darzustellen. In der Regel werden weitere Spalten benötigt. Diese können erstellt werden, indem Sie die gewünschte Abfrage auswählen und links oben in der Navigationsliste unter der Schaltfläche „New“ (Neu) die Option „Column“ (Spalte) auswählen. Jede Spalte hat die folgenden Eigenschaften:

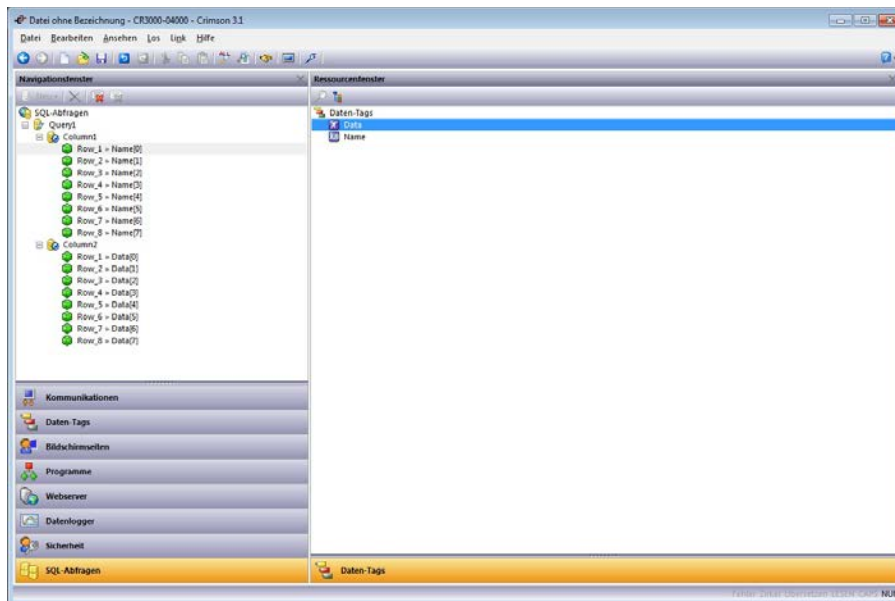


- Mit der Eigenschaft *SQL Column Name* (SQL-Spaltenname) geben Sie an, wie der Name der Spalte in der SQL-Tabelle lautet, in der die Abfrage durchgeführt werden soll. Er sollte mit dem Namen übereinstimmen, der beim Erstellen der Tabelle angegeben wurde.
- Mit der Eigenschaft *SQL Type* (SQL-Typ) können Sie den Datentyp der Spalte auswählen. Er sollte mit dem Typ übereinstimmen, der beim Erstellen der Tabelle angegeben wurde. Damit wird die Zuordnung von SQL-Tags zu Crimson 3.1-Tags gesteuert.

16.4 Zugeordnete Tags

Für jede Spalte, die in der Abfrage enthalten ist, werden in der Navigationsliste mehrere Zeileneinträge angezeigt, deren Anzahl sich nach der Anzahl der Zeilen richtet, die in den Abfrageeigenschaften ausgewählt sind. Sie können Tags per Drag-and-Drop aus dem Ressourcenfenster ziehen, um eine Zuordnung zwischen den Abfrageergebnissen und den Tags zu erstellen, die zum Speichern der resultierenden Daten verwendet werden. In der Regel verwenden Sie ein Array, um die einzelnen Spalten darzustellen, und jede Zeile wird einem bestimmten Array-Element zugeordnet.

Das folgende Beispiel zeigt, wie die einzelnen Zeilen der einzelnen Spalten einem Element in einem Crimson-Array zugeordnet wird. Die Ergebnisdaten können somit schnell vom Rest der Crimson 3.1-Datenbank abgerufen werden:



Kapitel 17 Verwenden von IEC-61131

Die Kategorie „Control“ (Steuerung) der Datenbank wird zum Erstellen und Verwalten von Programmen verwendet, die in einer der vier von Crimson® 3.1 unterstützten IEC-61131 Sprachen geschrieben wurden. (Die Sprache „Sequential Function Charts“ (Ablaufsprache) wird derzeit nicht unterstützt.) Diese Programme können in regelmäßigen Abständen ausgeführt werden, um steuerungs-basierte Aktivitäten im Zusammenhang mit den E/A-Modulen durchzuführen, die als Plug-in-Erweiterungsmodul oder über Kommunikationsverbindungen mit dem Gerät verbunden sind. Die Programme können Ihre eigenen lokalen Variablen verwenden, und globale Variablen können zum Speichern von gemeinsam genutzten Informationen verwendet werden. Es stehen zahlreiche Systemfunktionen zur Verfügung, sodass in Ihrem Crimson-Gerät umfangreiche Steuerungs-lösungen erstellt werden können.

17.1 Was bietet die Kategorie Steuerung?

Beachten Sie, dass Vieles von dem, was mit der IEC-61131-Engine erreicht werden kann, theoretisch auch über die Kategorie „Programs“ (Programme) von Crimson möglich ist. Die Kategorie „Control“ (Steuerung) bietet jedoch einige bedeutende Vorteile. Erstens eignen sich die von der Kategorie „Control“ (Steuerung) unterstützten Programmiersprachen viel besser für die Maschinen- und Prozesssteuerung. Die in der Kategorie „Programs“ (Programme) verwendete C-ähnliche Sprache hingegen ist eher für die Datenverarbeitung und die Verwaltung der Benutzeroberfläche geeignet. Zweitens werden Programme in der Kategorie „Control“ (Steuerung) auf einer periodischen, halbdeterministischen Basis ausgeführt, während Programme in der Kategorie „Programs“ (Programme) eher als Reaktion auf Datenänderungen oder Benutzeraktionen ausgeführt werden. Drittens ist die Bibliothek mit Funktionsbausteinen der Kategorie „Control“ (Steuerung) besser für Steuerungsanwendungen geeignet, während die Programme von Crimson für die Interaktion mit Bibliotheksfunktionen entwickelt wurden, die sich auf die Kommunikation und die Benutzeroberfläche beziehen.

17.2 Erlernen von IEC-61131

Dieses Kapitel ist nicht dafür vorgesehen, Ihnen die Programmierung in IEC-61131 beizubringen. Dies würde den Rahmen dieses Handbuchs sprengen. Zu diesem Thema gibt es jedoch zahlreiche hervorragende Onlineresourcen. Die Mitarbeiter des technischen Supports von Red Lion sind gern bereit, Ihre Fragen zu Crimson zu beantworten und Ihnen auch generell behilflich zu sein. Es ist ihnen aber nicht möglich, Sie über das Telefon zu einem IEC-61131-Programmierer zu machen. Auf der Webseite von Red Lion finden Sie Informationen zu empfohlenen Schulungsressourcen für IEC-61131.

17.3 Arbeiten mit Programmen

Programme sind die grundlegenden Ausführungseinheiten in der IEC-61131-Engine von Crimson.

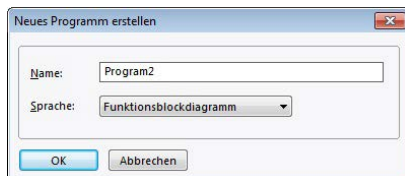
17.3.1 Programmtypen

Es werden drei Programmtypen unterstützt:

- Hauptprogramme werden von der IEC-61131-Engine entweder bei jedem Abtasten oder alle paar Sekunden aufgerufen. Ihre Datenbank sollte mindestens ein Hauptprogramm enthalten und kann viele mehr enthalten. Die Ausführungsreihenfolge der Hauptprogramme wird vom Projektobjekt in der Navigationsliste über den unten beschriebenen Prozess verwaltet.
- Unterprogramme werden von anderen Programmen aufgerufen. Sie werden nur ausgeführt, wenn ein Hauptprogramm sie entweder direkt oder über ein anderes Unterprogramm aufruft. Sie können dazu verwendet werden, Ihre Steuerungslösung in überschaubarere Abschnitte aufzuteilen oder Teile der Lösung in Abhängigkeit von bestimmten Bedingungen auszuführen.
- Benutzerdefinierte Funktionsbausteine sind wie Unterprogramme. Sie haben jedoch speziell definierte Eingangs- und Ausgangsvariablen, sodass sie in einem anderen Programm platziert werden können, um eine bestimmte Funktion zu implementieren. Zum Beispiel kann ein benutzerdefinierter Funktionsbaustein in einem FBD-Programm (Function Block Diagram, Funktionsbausteinsprache) platziert werden, wobei die Eingänge und Ausgänge mit anderen Bausteinen oder E/A-Punkten verbunden sind.

17.3.2 Erstellen von Programmen

Die Programme werden wie gewohnt vom Navigationsfenster aus über die Kategorie „Control“ (Steuerung) unter dem Abschnitt „Programs“ (Programme) erstellt. Unter der Schaltfläche „New“ (Neu) gibt es Menüoptionen, mit denen Programme der drei oben beschriebenen Typen erstellt werden können. Wenn ein Programm erstellt wird, werden Sie aufgefordert, den Namen des Programms und die zu verwendende IEC-61131-Sprache anzugeben:



Die unterstützten Sprachen sind:

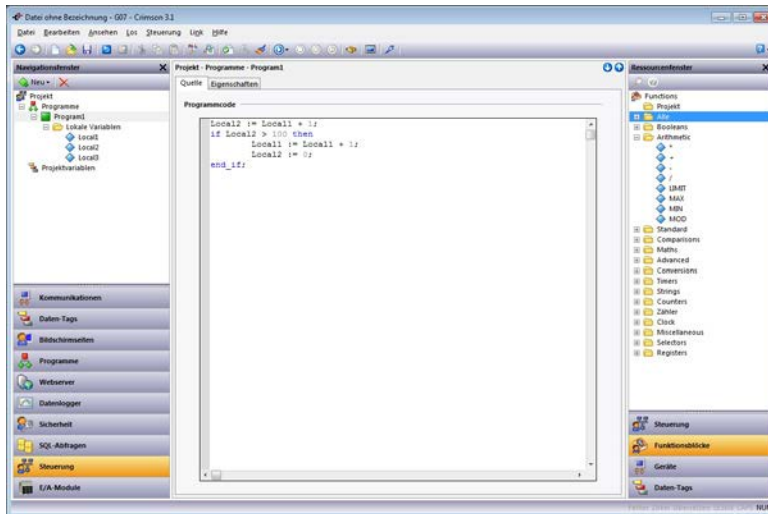
- Strukturierter Text (Structured Text), eine höhere textbasierte Sprache, die der Programmiersprache PASCAL ähnelt. Strukturierter Text unterstützt Funktionsaufrufe für den Zugriff auf andere Programme und die Funktionsbibliothek sowie verschiedene Steuerungsstrukturen, mit denen Schleifen erstellt und Möglichkeiten zur Entscheidungsfindung hinzugefügt werden können.
- Funktionsbausteinsprache (Function Block Diagram), eine grafische Sprache, bei der Funktionsbausteine miteinander verbunden werden können, um den Datenfluss zwischen diesen darzustellen. Jeder Funktionsbaustein hat ein oder mehrere Ein- und Ausgänge. Diese können mit anderen Bausteinen oder E/A-Punkten verbunden werden, die IEC-61131-Variablen darstellen.
- Kontaktplan (Ladder Diagram), die grafische Sprache, die traditionell für die Programmierung von SPS verwendet wird. Sie stellt den Stromlauf von einer Schiene zu einer anderen anhand einer Reihe von simulierten Relaispulen und Kontakten dar, die IEC-61131-Variablen zugeordnet sind.
- Anweisungsliste (Instruction List), eine niedere textbasierte Sprache, die einer Form von Assembler Code ähnelt. Sie verfügt nicht über die hochrangigen Konstrukte von Strukturierter Text, und die Flussteuerung wird mithilfe von Sprung- und Aufrufbefehlen implementiert. Anweisungsliste wird für neue Anwendungen nicht empfohlen und ist in den neueren IEC-61131-Standards als veraltet gekennzeichnet.

17.3.3 Bearbeiten von Programmen

Wenn ein Programm ausgewählt wird, wird im Bearbeitungsfenster ein geeigneter Editor angezeigt. Das Erscheinungsbild des Editors hängt davon ab, in welcher Sprache das Programm erstellt wurde. In den folgenden Abschnitten werden die einzelnen Editoren nacheinander beschrieben.

17.3.3.1 Der Editor für Strukturierter Text

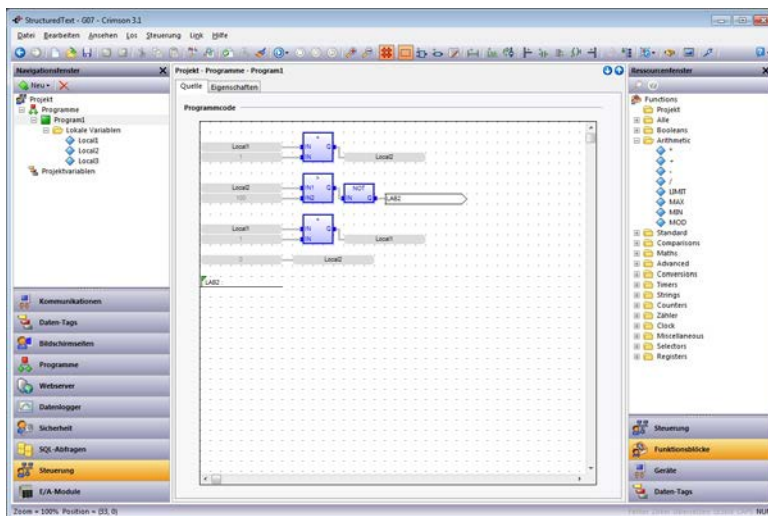
In der folgenden Abbildung sehen Sie den Editor für Programme, die in Strukturierter Text geschrieben wurden:



Anweisungen werden genau wie bei jedem anderen Windows-Texteditor eingegeben und bearbeitet. Variablen, Operatoren und Funktionsaufrufe können aber per Drag-and-Drop aus dem Ressourcenfenster in den Text gezogen werden. Wie bei anderen Editoren wird der Code durch Drücken von **STRG+T** überprüft. Etwaige Fehler werden in der globalen Fehlerliste erfasst, wo sie über die Tastenfolge **F8** und **F4** abgerufen werden können.

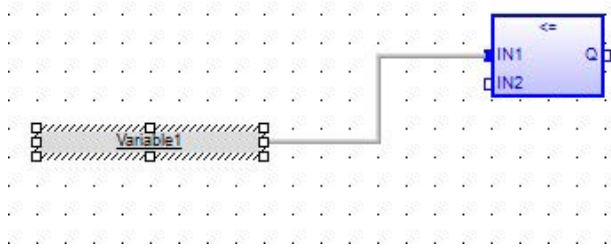
17.3.3.2 Der Editor für die Funktionsbausteinsprache

In der folgenden Abbildung sehen Sie den Editor für Programme, die in der Funktionsbausteinsprache geschrieben wurden:



Variablen und sowohl Standard- als auch benutzerdefinierte Funktionsbausteine sind im Ressourcenfenster verfügbar und können im Programm platziert werden. Funktionsbausteine können Variablen enthalten, die an ihre

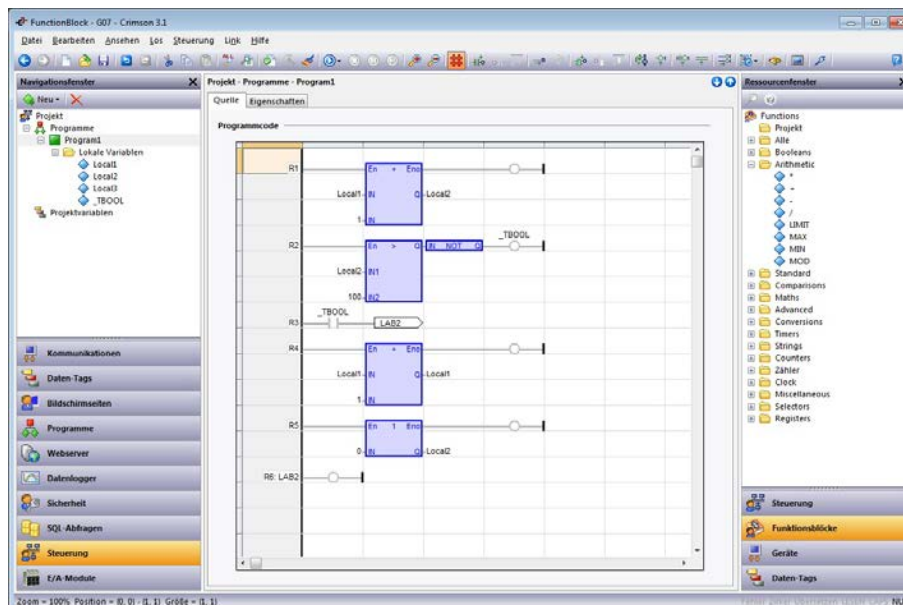
Ein- und Ausgänge gebunden sind, wenn die entsprechende Option in der Symbolleiste aus dem Menü „Options“ (Optionen) ausgewählt wird. Sobald Elemente hinzugefügt wurden und Sie den Cursor auf einem der Eingangs- oder Ausgangsanschlusspunkte platzieren, können Sie eine Linie zu einem anderen Baustein ziehen, um eine Verbindung zu erstellen und somit den Datenfluss anzugeben. In der folgenden Abbildung sehen Sie, wie eine Variable anhand dieses Verfahrens mit dem Eingang eines Vergleichsoperators verknüpft wird:



Die Symbolleiste des Editors für die Funktionsbausteinsprache enthält Befehle, mit denen Sie die Vergrößerungseinstellung der Diagrammansicht steuern, das Raster ein- oder ausblenden sowie dem Programm eine Vielzahl von Standardelementen hinzufügen können. Wie bei anderen Editoren wird der Code durch Drücken von **STRG+T** überprüft. Etwaige Fehler werden in der globalen Fehlerliste erfasst, wo sie über die Tastenfolge **F8** und **F4** abgerufen werden können.

17.3.3.3 Der Kontaktplan-Editor

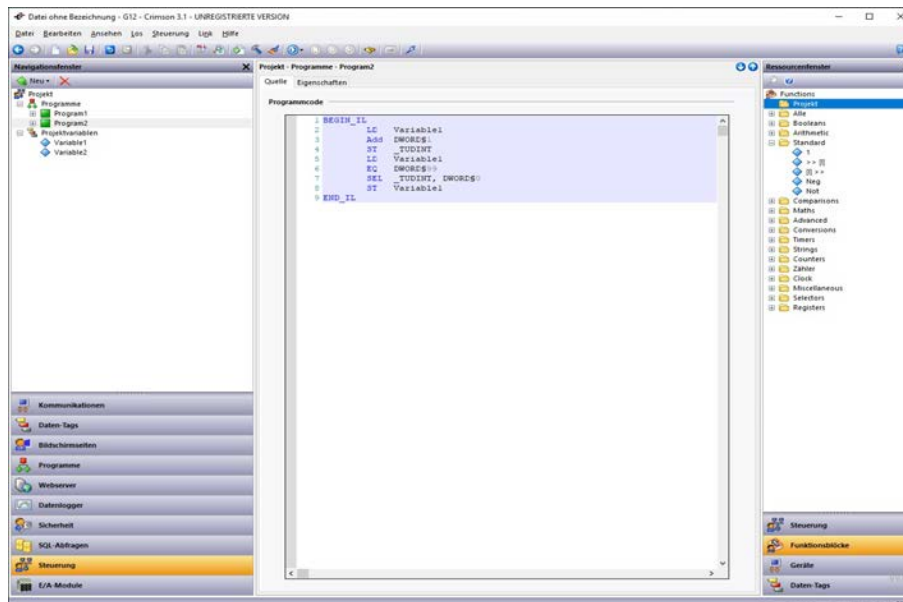
In der folgenden Abbildung sehen Sie den Editor für Programme, die in der Sprache Kontaktplan geschrieben wurden:



Wie bei den in der Funktionsbausteinsprache geschriebenen Programmen sind auch hier Variablen und sowohl Standard- als auch benutzerdefinierte Funktionsbausteine im Ressourcenfenster verfügbar und können im Programm platziert werden. Darüber hinaus können Sie in der Symbolleiste auf verschiedene Standardelemente von Kontaktplan zugreifen und somit einfacher Kontakte und Spulen hinzufügen. Die Symbolleiste des Editors für die Funktionsbausteinsprache enthält Befehle, mit denen Sie die Vergrößerungseinstellung der Diagrammansicht steuern, das Raster ein- oder ausblenden sowie dem Programm eine Vielzahl von Standardelementen hinzufügen können. Wie bei anderen Editoren wird der Code durch Drücken von **STRG+T** überprüft. Etwaige Fehler werden in der globalen Fehlerliste erfasst, wo sie über die Tastenfolge **F8** und **F4** abgerufen werden können.

17.3.3.4 Der Editor für Anweisungsliste

In der folgenden Abbildung sehen Sie den Editor für Programme, die in der Sprache Anweisungsliste geschrieben wurden:



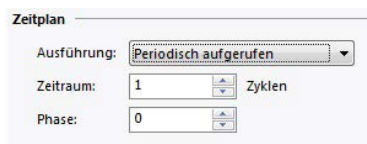
Anweisungen werden wie bei jedem beliebigen Windows-Texteditor eingegeben und bearbeitet. Beachten Sie, dass Anweisungsliste als niedere Programmiersprache im Ressourcenfenster lediglich den Zugriff auf Variablen gestattet. Wie bei anderen Editoren wird der Code durch Drücken von **STRG+T** überprüft. Etwaige Fehler werden in der globalen Fehlerliste erfasst, wo sie über die Tastenfolge **F8** und **F4** abgerufen werden können.

17.3.4 Konvertieren von Programmen

Programme können von einer Sprache in eine andere konvertiert werden. Klicken Sie dazu in der Navigationsliste mit der rechten Maustaste auf ein Programm und wählen Sie im Kontextmenü eine der Konvertierungsoptionen aus. Beachten Sie, dass die Konvertierung nicht rückgängig gemacht werden kann und die globale Rückgängig-Liste löscht. Beachten Sie auch, dass nicht alle Sprachkonstruktionen konvertiert werden können. Wenn die Umwandlung fehlschlägt, wird eine entsprechende Fehlermeldung mit einer Begründung angezeigt.

17.3.5 Programmeigenschaften

Über die Registerkarte „Properties“ (Eigenschaften) im Bearbeitungsfenster können Sie auf die Eigenschaften eines Programms zuzugreifen:



- Mit der Eigenschaft *Execution* (Ausführung) können Sie angeben, wann das Programm ausgeführt wird. Bei allen Programmen, die keine Hauptprogrammen sind, wird sie immer auf „Called by Another Program“ (Durch ein anderes Programm aufgerufen) gesetzt, und die restlichen Eigenschaften werden ausgeblendet. Bei Hauptprogrammen bewirkt die Einstellung „Called on Each Cycle“ (Bei jedem Zyklus aufgerufen), dass das Programm jedes Mal aufgerufen wird, wenn die IEC-61131-Engine eine

Abtastung durchführt. Die Einstellung „Called Periodically“ (Regelmäßig aufgerufen) bewirkt, dass das Programm mit einer Häufigkeit aufgerufen wird, die von den anderen Eigenschaften definiert wird. Die Einstellung „Do Not Execute“ (Nicht ausführen) kann dazu verwendet werden, ein Programm während des Testens zu deaktivieren.

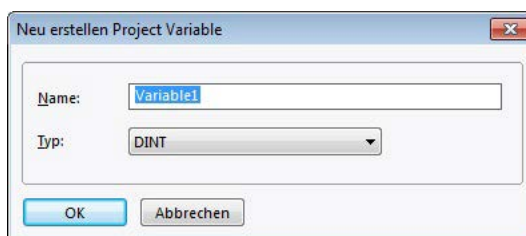
- Mit den Eigenschaften *Period* (Zeitraum) und *Phase* (Phase) wird gesteuert, wie häufig ein Hauptprogramm mit periodischer Ausführung ausgeführt wird. Die Eigenschaft *Period* (Zeitraum) gibt an, wie viele Abtastungen zwischen einem Aufruf des Programms und dem nächsten verstreichen. Die Multiplizierung dieses Werts wird mit der Abtastzeit gibt die Häufigkeit an, mit der das Programm ausgeführt wird. Die Eigenschaft *Phase* kann verwendet werden, um die Abtastungen zu verschieben, bei denen die Programme eines bestimmten Zeitraums ausgeführt werden. Sie sollte auf einen Wert zwischen 0 und dem unter *Period* (Zeitraum) angegebenen Wert eingestellt werden. Ein Programm wird ausgeführt, wenn der Rest der Abtastzahl geteilt durch den Zeitraum gleich der Phase ist. Angenommen, für Programm 1 und Programm 2 wurde „Period“ (Zeitraum) auf 5 gesetzt und „Phase“ auf 0. In diesem Fall werden beide Programme bei den Abtastungen 0, 5, 10 usw. ausgeführt. Wenn die Phase für Programm 2 in 1 geändert wird, wird es jetzt bei den Abtastungen 1, 6, 11 usw. ausgeführt. Die Phase kann also dazu verwendet werden, die Ausführungslast in einem System mit vielen periodischen Programmen auszugleichen, indem diese Programme auf einzelne Abtastungen verteilt werden.

17.4 Verwenden von Variablen

Die IEC-61131-Engine wirkt auf Variablen ein. Diese Variablen können der Kategorie „Control“ (Steuerung) angehören oder anderen Crimson-Datenelementen wie Tags, Ausdrücken oder Verweisen auf Kommunikations- oder E/A-Datenpunkte zugeordnet sein.

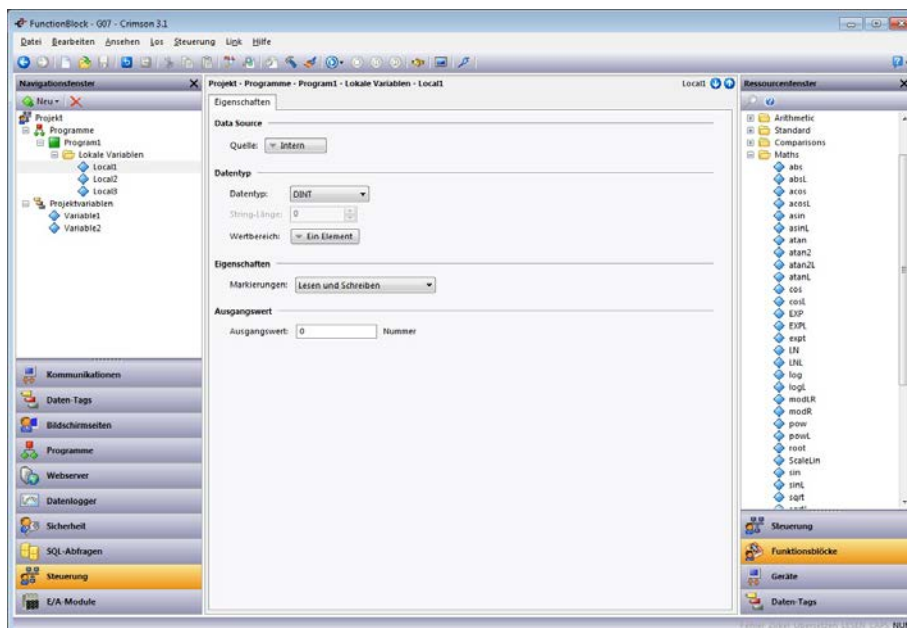
17.4.1 Erstellen von Variablen

Variablen können in der Navigationsliste unter dem Ordner „Local Variables“ (Lokale Variablen) eines Programms oder im Ordner „Project Variables“ (Projektvariablen) des Projekts erstellt werden. Lokale Variablen sind nur für das Programm sichtbar, zu dem sie gehören. Projektvariablen hingegen sind für alle Programme sichtbar und können zum Speichern von Informationen eingesetzt werden, die im Projekt gemeinsam genutzt werden. Wenn eine Variable erstellt wird, werden Sie aufgefordert, einen Namen und den IEC61131-Datentyp anzugeben:



17.4.2 Eigenschaften von Variablen

Jede Variable hat die folgenden Eigenschaften:



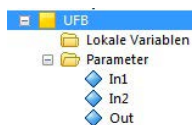
- Mit der Eigenschaft *Data Source* (Datenquelle) wird angegeben, ob die Variable der IEC-61131-Engine intern angehört oder ob sie einem Ausdruck, einem Crimson-Tag oder einem anderen Datenelement zugeordnet werden soll. Sie funktioniert genau wie im Kapitel über Tags beschrieben. Wenn eine Variable einem Crimson-Tag zugeordnet ist, muss der IEC-61131-Datentyp für die direkte Konvertierung in den Datentyp des Tags geeignet sein.

- Mit der Eigenschaft *Data Type* (Datentyp) wird der IEC-61131-Datentyp der Variablen angegeben oder geändert. Wenn der Typ geändert wird, nachdem die Variable in einem Programm referenziert wurde, müssen Sie möglicherweise das Steuerprojekt neu aufbauen und resultierende Fehler korrigieren. Weitere Informationen finden Sie später in diesem Kapitel.
- Die Eigenschaft *String Length* (Zeichenfolgenlänge) wird nur für Variablen des Typs `STRING` verwendet. Sie gibt an, wie viele Zeichen die Zeichenfolge aufnehmen kann.
- Mit der Eigenschaft *Extent* (Umfang) wird angegeben, ob diese Variable ein einzelnes Datenelement oder ein eindimensionales Array ist. Wenn die Einstellung „Array“ ausgewählt wird, wird eine weitere Eigenschaft angezeigt, mit der die Länge des Arrays angegeben werden kann.
- Mit der Eigenschaft *Flags* (Markierungen) wird eine Variable als schreibgeschützt gekennzeichnet und so der Zugriff auf diese Variable eingeschränkt. Sie kann verwendet werden, um zu verhindern, dass Crimson-Tags geändert werden, oder um Daten zu schützen, die über „Initial Values“ (Ausgangswerte) konfiguriert wurden.
- Mit der Eigenschaft *Initial Value* (Ausgangswert) wird angegeben, welche Werte diese Variable bei einem Kaltstart laden soll, wenn keine gepufferten Daten verwendet werden, bzw. bei der ersten Ausführung dieser Datenbank durch ein bestimmtes Zielgerät. Bei komplexen Datentypen wird eine Auswahl Schaltfläche angezeigt, damit die Daten bearbeitet werden können.

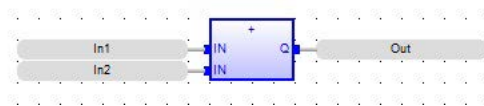
17.4.3 Parameter

Parameter sind vergleichbar mit lokalen Variablen, doch es gibt einen wichtigen Unterschied: Alle Programme können lokale Variablen enthalten, aber Parameter existieren nur für benutzerdefinierte Funktionsbausteine. Sie werden in der Navigationsliste in einem separaten Ordner angezeigt und haben leicht unterschiedliche Eigenschaften. Zum Beispiel kann jeder Parameter als Eingang, Ausgang oder beides festgelegt werden. Parameter können außerdem nicht Crimson-Datenelementen zugeordnet werden. Bei der Ausführung des Funktionsbausteins werden vielmehr die Eingangsparameter mit den Daten geladen, die über das Programm, das sie aufruft, an den Baustein gebunden sind. Nach der Ausführung werden die in den Ausgabeparametern platzierten Werte an die Elemente im aufrufenden Programm übergeben, die diese Daten verbrauchen.

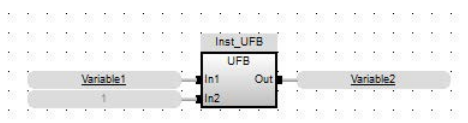
Betrachten Sie den folgenden benutzerdefinierten Funktionsbaustein:



Er enthält einen sehr einfachen Code, der lediglich zwei Eingänge hinzufügt:



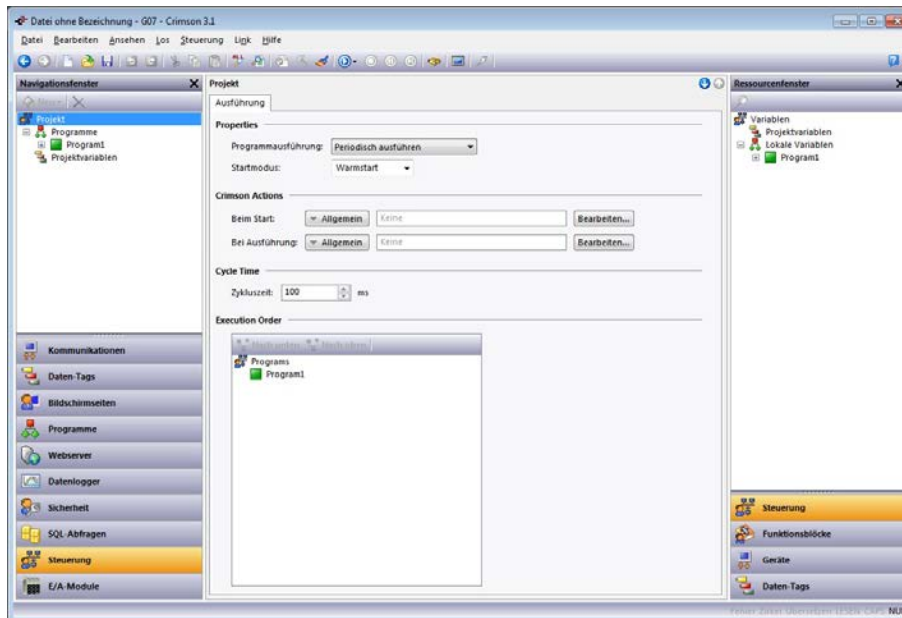
Und so wird er von einem anderen Programm aufgerufen:



Bevor der benutzerdefinierte Funktionsbaustein ausgeführt wird, wird sein Parameter `In1` auf den Inhalt von `Variable2` gesetzt und sein Parameter `IN2` auf den konstanten Wert von 1. Der Funktionsbaustein wird dann ausgeführt und platziert `Variable2+1` in seinem Parameter `Out`. Das aufrufende Programme speichert diesen Ausgang in `Variable3` und schließt den Vorgang ab.

17.5 Projekteigenschaften

Durch Auswählen des Elements „Project“ (Projekt) in der Navigationsliste können Sie auf Eigenschaften zugreifen, die sich auf die Ausführung des gesamten Projekts beziehen. Diese Eigenschaften sehen Sie in der nachstehenden Abbildung:



- Die Eigenschaft *Program Execution* (Programmausführung) wird verwendet, um die Ausführung von IEC-61131 global zu aktivieren bzw. zu deaktivieren. Standardmäßig ist sie deaktiviert, um die Anzeige bestimmter Eingabeaufforderungen zu vermeiden, die beim Herunterladen auf Geräte erforderlich sind, die eine ausgeführte Instanz der IEC-61131-Engine enthalten. **Setzen Sie diese Eigenschaft unbedingt auf „Execute Periodically“ (Periodisch ausführen), sonst werden Ihre Programme nicht ausgeführt.**
- Die Eigenschaft *Start Mode* (Startmodus) wird verwendet, um den Status der gepufferten Variablen beim Einschalten zu definieren. Wenn „Cold Start“ (Kaltstart) ausgewählt wird, werden die Ausgangswerte verwendet und die remanenten Werte verworfen. Wenn „Warm Start“ (Warmstart) ausgewählt wird, werden die remanenten Werte verwendet, sofern sie verfügbar sind; andernfalls die Ausgangswerte.
- Die Eigenschaften *Crimson Action* (Crimson-Aktion) werden verwendet, um Crimson-Programme oder andere Aktionen zu definieren, die von der IEC-61131-Engine beim Start und vor jedem Scan ausgeführt werden. Diese Aktionen werden innerhalb des Hauptscans durchgeführt, und es muss darauf geachtet werden, keinen komplexen Code auszuführen, der die angeforderte Scandauer beeinträchtigen würde. Beachten Sie, dass Sie keine IEC-61131-Lizenz benötigen, um diese Aktionen verwenden zu können. Jedes Gerät, das die Crimson-Steuerung unterstützen kann, führt diese Aktionen auch dann aus, wenn es eine Lizenz für die IEC-61131-Engine zwar benötigt, aber nicht besitzt.
- Mit der *Cycle Time* (Zykluszeit) wird festgelegt, wie oft die IEC-61131-Engine alle Programme mit der Einstellung „Called on Each Cycle“ (Bei jedem Zyklus aufgerufen) ausführen soll. Außerdem wird die Zeitbasis für den oben beschriebenen periodischen Ausführungsmechanismus bereitgestellt.
- Mit der Eigenschaft „Execution Order“ (Ausführungsreihenfolge) wird die Reihenfolge gesteuert, in der Hauptprogramme von der IEC-61131-Engine ausgeführt werden. Die in der Liste angezeigten Programme können mithilfe der Schaltflächen in der angefügten Symbolleiste nach oben und unten verschoben werden.

17.6 Neuerstellen von Projekten

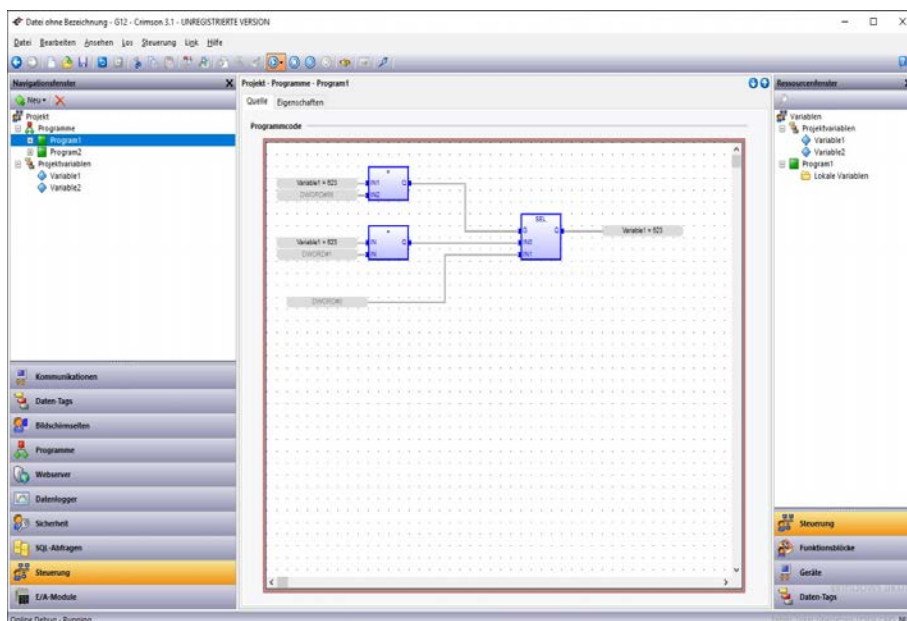
Bei bestimmten Änderungen ist es u. U. erforderlich, dass Sie ihre IEC-61131-Programme neu erstellen, um z. B. der Änderung des Typs einer globalen Variablen Rechnung zu tragen. Hierzu können Sie **STRG+B** drücken, das Hammersymbol in der Symbolleiste verwenden oder die Option „Rebuild“ (Neu erstellen) im Menü „Control“ (Steuerung) wählen. Crimson fordert Sie außerdem vor dem Herunterladen der Datenbank hierzu auf, falls die Datenbank Programme oder andere Steuerelemente enthält, die neu erstellt werden müssen.

17.7 IEC-61131-Debugging

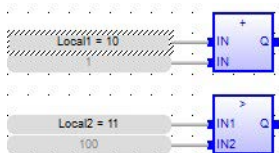
Das IEC-61131-System unterstützt zwei Debugging-Modi, in denen der Ausführungsstatus eines Programms und die zugehörigen Datenelemente in den zugehörigen Editoren angezeigt werden können.

- Der Modus „Offline Emulation“ (Offline-Emulation) führt das Steuerungsprogramm auf dem Windows-basierten Konfigurations-PC aus und emuliert das Verhalten, das zu erwarten ist, wenn die Datenbank auf das Zielgerät heruntergeladen wird.
- Der Modus „Online Debugging“ (Online-Debugging) lädt die Datenbank auf das Zielgerät herunter und ermöglicht Ihnen anschließend, die Steuerungsprogramme im Crimson-Konfigurationstool zu debuggen, während sie auf diesem Gerät ausgeführt werden.

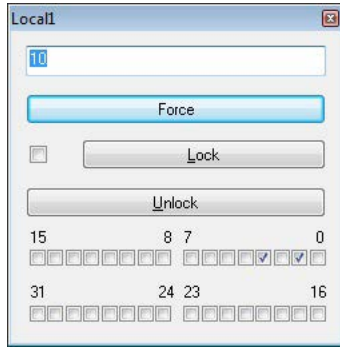
Die Debugging-Modi werden über die Wiedergabe-Schaltfläche in der Symbolleiste aktiviert. Diese Schaltfläche zeigt ein Menü an, in dem der geeignete Modus ausgewählt werden kann. Daraufhin wechselt der aktuelle Editor in den Debugging-Modus:



Der Editor wird mit einem doppelten Rechteck markiert, um daran zu erinnern, dass der Debugging-Modus aktiviert wurde. Im Offline-Modus ist das Rechteck blau und im Online-Modus ist es rot. Der Inhalt des Editors kann nicht mehr geändert werden, dafür werden die aktuellen Werte der Datenelemente an den entsprechenden Stellen angezeigt, wie in der nachfolgenden Abbildung gezeigt:



Während das Programm ausgeführt wird, werden die im Programm gezeigten Werte von Variable2 in Echtzeit aktualisiert. Wenn Sie den Wert einer Variablen bearbeiten möchten, doppelklicken Sie auf die Variable, um das Bearbeitungsfenster anzuzeigen:



Sie können einen neuen Wert in das Feld eingeben und auf die Schaltfläche „Force“ (Erzwingen) klicken. Sie können auch die verschiedenen Kontrollkästchen zum Aktualisieren der einzelnen Bits der Variablen verwenden. Seien Sie stets vorsichtig, wenn Sie Werte von Steuerungssystemen bearbeiten, die mit laufenden Maschinen verbunden sind!

Weitere Schaltflächen auf der Symbolleiste können zur Steuerung des Debugging-Prozesses verwendet werden:



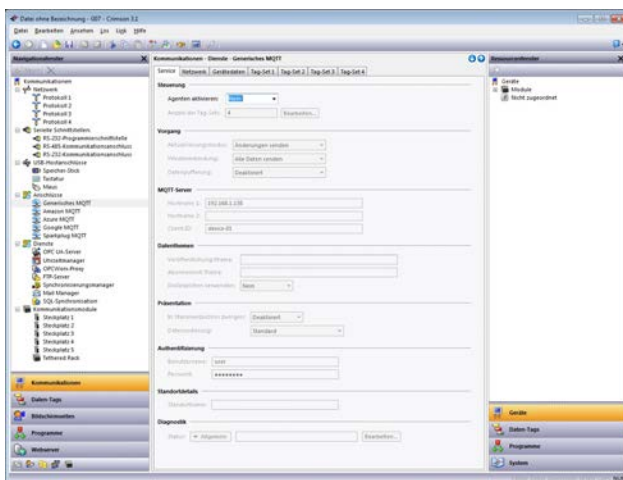
Die Wiedergabe-Schaltfläche und die Stopp-Schaltfläche können zum Anhalten der Emulation bzw. zur Trennung von einer Online-Debugging-Sitzung verwendet werden. Mit der Pause-Schaltfläche kann die Ausführung des Programms angehalten werden. Mit der Schritt-Schaltfläche kann die Ausführung dann Schritt für Schritt wiedergegeben werden. Wenn dieser Modus ausgewählt wird, wird die Pause-Schaltfläche hervorgehoben und die Schritt-Schaltfläche wird verfügbar. Durch erneutes Drücken der Pause-Schaltfläche wird die normale Ausführung fortgesetzt. Einzelschritte sind in den textbasierten Programmiersprachen sehr nützlich, weil der Programmablauf überprüft werden kann. Wenn eine Online-Debugging-Sitzung getrennt wird, während das System angehalten ist, verbleibt das System in diesem Zustand. Eine entsprechende Warnmeldung wird angezeigt.

Kapitel 18 Verwenden von Anschlüssen

Mit den Crimson® 3.1-Anschlüssen ist es möglich, Tag-Daten und Gerätestatusinformationen in die Cloud oder ein SCADA-System im Haus zu pushen. Unterstützung wird bereitgestellt für das JSON-basierte MQTT für Amazon Web Services™ (AWS), Microsoft Azure und den Google Cloud Platform™ Service sowie für das Sparkplug-basierte MQTT für Pakete wie das Ignition® SCADA-System von Inductive Automation®. Ein generischer JSON-basierter MQTT-Treiber wird ebenfalls bereitgestellt, um die Konnektivität mit einer Vielzahl anderer Systeme zu ermöglichen, darunter das Status Enterprise System von B-Scada und der MQTT-Treiber von Kepware®.

18.1 Konfigurieren von Anschlüssen

Die Crimson® 3.1-Anschlüsse werden über den Abschnitt „Communications“ (Kommunikationen) im Navigationsfenster konfiguriert.



18.1.1 Allgemeine Einstellungen

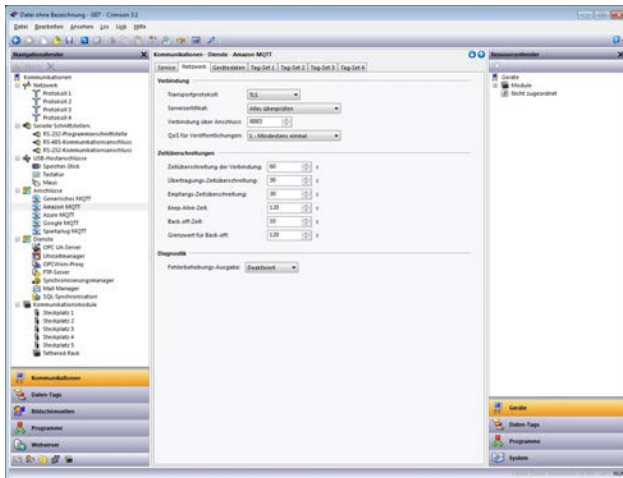
Die Anschlüsse verfügen über sehr ähnliche Eigenschaften, die über die folgenden Registerkarten verteilt sind:

- Auf der Registerkarte *Service* (Dienst) werden die anschlusspezifischen Informationen konfiguriert.
- Auf der Registerkarte *Network* (Netzwerk) werden die zugrunde liegenden Verbindungsoptionen konfiguriert.
- Auf der Registerkarte *Device Data* (Gerätedaten) werden die Gerätestatusdaten angegeben, die an den Server gepusht werden sollen.
- Auf der Registerkarte *Tag Data* (Tag-Daten) werden die Tag-Daten angegeben, die an den Server gepusht werden sollen.

18.1.1.1 Netzwerkooptionen

Auf der Registerkarte „Network“ (Netzwerk) werden die grundlegenden Verbindungsoptionen konfiguriert, die vom Anschluss verwendet werden. Die Standardeinstellungen dürften für den fraglichen Anschluss angemessen sein, müssen jedoch ggf. geändert werden, falls Sie beispielsweise eine Verbindung zu einem nicht standardmäßigen Port herstellen.

Die Netzwerkooptionen sehen Sie in der nachstehenden Abbildung.



- Die Eigenschaft *Transport Protocol* (Transportprotokoll) dient zur Auswahl von TCP bzw. TLS. Die Einstellung „TCP“ verwendet das unverschlüsselte TCP/IP-Protokoll und die Einstellung „TLS“ das verschlüsselte und sichere TLS- bzw. SSL-Protokoll.
- Mit der Eigenschaft *Server Certificate* (Serverzertifikat) wird die Validierungsstufe definiert, die auf das vom Server bereitgestellte Zertifikat angewendet wird. Bei Auswahl von „Ignore“ (Ignorieren) wird keine Prüfung durchgeführt. Bei Auswahl von „Check CA“ (ZS prüfen) wird sichergestellt, dass das Zertifikat zu einem vertrauenswürdigen Zertifikat zurückverfolgt werden kann. Bei Auswahl von „Check CA and Name“ (ZS und Name prüfen) wird zusätzlich sichergestellt, dass das Zertifikat für den richtigen Servernamen gilt. Bei Auswahl von „Check Everything“ (Alles prüfen) wird zusätzlich das Ablaufdatum des Zertifikats geprüft. Idealerweise sollten Sie „Check Everything“ (Alles prüfen) wählen, doch jede Form der Validierung erfordert besondere Vorsicht. Damit die Vertrauenskette validiert werden kann, muss ein geeignetes Stammzertifikat verfügbar sein, entweder über die Anschlusseigenschaften oder über die Registerkarte „TLS-SSL“ in den Netzwerkeinstellungen. Damit der Name validiert werden kann, muss anstelle einer IP-Adresse ein Hostname verwendet werden. Damit das Ablaufdatum validiert werden kann, muss die Uhr des Geräts richtig eingestellt sein, idealerweise über die im Kapitel „Verwenden von Diensten“ beschriebene Netzwerkzeit-Synchronisierung.
- Mit der Eigenschaft *Connect via Port* (Verbindung über Anschluss) wird der TCP/IP-Port definiert, mit dem der Anschluss verbunden wird. Bei den meisten TCP MQTT-Verbindungen lautet der Wert 1883 und bei den meisten TLS MQTT-Verbindungen 8883.
- Mit der Eigenschaft *Publication QoS* (Dienstqualität [Quality of Service, QoS] der Veröffentlichung) wird konfiguriert, auf welche Weise der Konnektor bestätigt, dass die an den Server gesendeten Daten erfolgreich empfangen wurden. Der Standardwert *Level 1* garantiert eine sogenannte *At Least Once*-Operation. Das bedeutet, dass jede Veröffentlichungsnachricht eine Bestätigung vom Server erhält bzw. erneut übertragen wird, falls diese Bestätigung nicht erhalten wird. Mit der Einstellung *Level 0* wird nur eine „Fire and Forget“-Übertragung ausgeführt, wodurch die TCP/IP-Ebene die Zustellung sicherstellen kann. Dies ist weniger zuverlässig, aber wesentlich schneller, wenn große Mengen historischer Daten über Verbindungen mit langen Umlaufzeiten wiedergegeben werden.
- Mit der Option *Connection Timeout* (Verbindungs-Timeout) wird festgelegt, wie lange der Anschluss beim Öffnen einer neuen Verbindung zum Server wartet. Wenn während dieses Zeitraums keine Verbindung

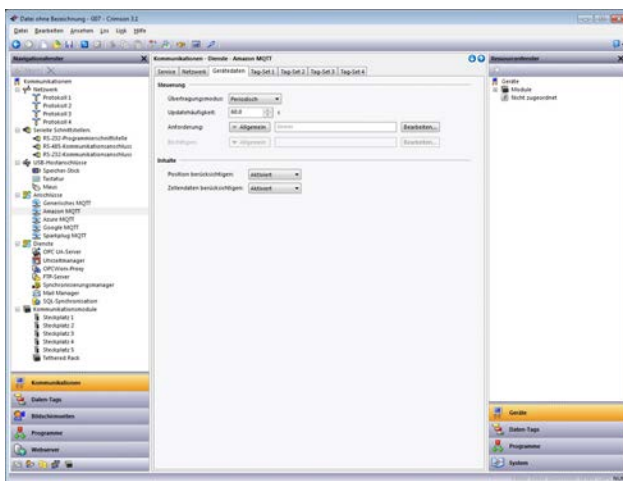
hergestellt wird, wird der Verbindungsversuch abgebrochen und anschließend wiederholt. Die Einstellung muss nicht geändert werden, es sei denn, Sie werden vom technischen Support dazu aufgefordert.

- Mit der Option *Transmit Timeout* (Übertragungs-Timeout) wird festgelegt, wie lange der Anschluss beim Senden einer Anforderung an den Server wartet. Wenn die Anforderung während dieses Zeitraums nicht gesendet werden kann, wird die Verbindung geschlossen und erneut geöffnet. Die Einstellung muss nicht geändert werden, es sei denn, Sie werden vom technischen Support dazu aufgefordert.
- Mit der Option *Receive Timeout* (Empfangs-Timeout) wird definiert, wie lange der Anschluss beim Empfangen einer Antwort vom Server wartet. Wenn die Antwort während dieses Zeitraums nicht empfangen wird, wird die Verbindung geschlossen und erneut geöffnet. Die Einstellung muss nicht geändert werden, es sei denn, Sie werden vom technischen Support dazu aufgefordert.
- Mit der Option *Keep Alive Time* (am-Leben-erhalten-Zeit) wird definiert, wie oft ein MQTT-Keep-Alive-Paket an den Server gesendet werden soll. Mithilfe dieser Pakete wird sichergestellt, dass die Verbindung offen gehalten wird und der Server weiterhin auf Anforderungen reagiert.
- Mit der Option *Back Off Time* (Back-off-Zeit) wird festgelegt, wie lange der Anschluss zunächst warten soll, bevor er versucht, die Kommunikation nach einer fehlgeschlagenen Verbindung zu allen definierten Hosts wiederherzustellen. Die Verzögerung beginnt bei diesem Wert und verdoppelt sich dann nach jedem Zyklus, bis die von der Eigenschaft *Back Off Limit* (Back-off-Limit) angegebene Zeit erreicht ist. Mit kleineren Werten können Verbindungen schneller wiederhergestellt werden, allerdings steigt die Bandbreitennutzung, wenn keine Hosts verfügbar sind.
- Die Eigenschaft *Debug Output* (Debug-Ausgabe) wird verwendet, um die Ausgabe erweiterter Diagnoseinformationen an die verschiedenen Debug-Konsolen des Systems zu aktivieren bzw. zu deaktivieren. Diese Informationen sind technischer Art und nicht vollständig dokumentiert. Sie sind für den technischen Support von Red Lion oder für erfahrenere Benutzer bestimmt. Informationen zum Konfigurieren von Debug-Konsolen finden Sie im Kapitel *Erweitertes Debuggen*.

18.1.1.2 Gerätedaten-Optionen

Mit der Registerkarte „Device Data“ (Gerätedaten) wird gesteuert, wie und wann Gerätestatusinformationen an den Server gepusht werden. Gegenwärtig übergeben die Anschlüsse nichts Nützliches im Gerätedatenblock, aber zukünftige Versionen werden Informationen über die Gerätehardware, Versionen und die Softwarelast bereitstellen. GPS-Positions- oder Mobilfunksignaldaten können optional eingeschlossen werden, falls verfügbar.

Die Gerätedaten-Optionen sehen Sie in der nachstehenden Abbildung.



- Mit der Eigenschaft *Transfer Mode* (Übertragungsmodus) können Sie angeben, wie die Gerätedaten übertragen werden sollen. Bei der Einstellung „Disabled“ (Deaktiviert) wird die Datenübertragung

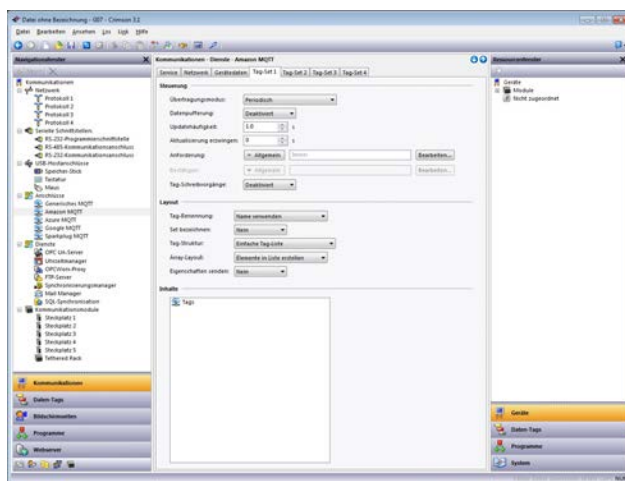
verhindert. Bei der Einstellung „Periodic“ (Regelmäßig) werden Daten regelmäßig übertragen. Bei der Einstellung „Triggered“ (Ausgelöst) werden Daten auf der steigenden Flanke eines auslösenden Ausdrucks übertragen. Gerätedaten werden an den Server übertragen, unabhängig davon, ob sie geändert wurden.

- Mit der Eigenschaft *Update Rate* (Aktualisierungshäufigkeit) wird gesteuert, wie oft die Gerätedaten an den Server gepusht werden, wenn der Modus „Periodic“ (Regelmäßig) ausgewählt ist. Gerätedaten werden in der Regel nicht zu oft gepusht, da sie sich wahrscheinlich nicht allzu stark verändern werden.
- Die Eigenschaft *Request* (Anforderung) hat zwei Verwendungszwecke. Im Modus „Periodic“ (Regelmäßig) fungiert sie als pegelempfindliche Aktivierung der Übertragung, d. h. die Übertragung findet nur statt, wenn der Ausdruck *true* oder nicht definiert ist. Im Modus „Triggered“ (Ausgelöst) fungiert sie als flankenempfindlicher Auslöser für die Übertragung, d. h. der Vorgang wird bei einer steigenden Flanke ausgelöst.
- Mit der Eigenschaft *Acknowledge* (Bestätigen) wird im Modus „Triggered“ (Ausgelöst) Feedback zum Status der Übertragung gegeben. Sie sollte auf einen änderbaren Ausdruck gesetzt werden, der auf *true* gesetzt wird, wenn die Übertragung abgeschlossen ist, und *false*, wenn die Eigenschaft „Request“ (Anforderung) wieder in den Status *false* zurückgekehrt ist.
- Mit der Eigenschaft *Include Location* (Position berücksichtigen) wird angegeben, ob Positionsdaten von einem GPS-Empfänger in den Gerätedatenblock aufgenommen werden sollen. Zum gegenwärtigen Zeitpunkt aktiviert diese Eigenschaft lediglich einen Platzhalter in der JSON-Struktur. Tatsächlich werden keine Daten übertragen. Mit der Eigenschaft *Include Cell Data* (Zellendaten berücksichtigen) wird angegeben, ob die Signalstärke und Zelleninformationen von einem Mobilfunkmodem im Gerätedatenblock enthalten sein sollen. Zum gegenwärtigen Zeitpunkt aktiviert diese Eigenschaft lediglich einen Platzhalter in der JSON-Struktur. Tatsächlich werden keine Daten übertragen.

18.1.1.3 Optionen von Tag-Daten

Mit den Registerkarten „Tag Data“ (Tag-Daten) wird gesteuert, welche Tags an den Server gepusht werden sowie wann und wie die Übertragung stattfindet. Standardmäßig werden vier Blöcke von Tag-Daten unterstützt, es können aber mithilfe der Eigenschaft *Tag Set Count* (Anzahl an Tag-Sätzen) auf der Seite „Service“ (Dienst) zusätzliche Blöcke erstellt werden. Mehrfache Tag-Sätze ermöglichen die Push-Übertragung von Daten bei unterschiedlichen Frequenzen sowie die Beschränkung der Schreibvorgänge vom Server an das Crimson-Gerät auf eine bestimmte Anzahl von Tag-Sätzen.

Die auf der Registerkarte „Tag Data“ (Tag-Daten) verfügbaren Optionen sehen Sie in der nachstehenden Abbildung.



- Mit der Eigenschaft *Transfer Mode* (Übertragungsmodus) können Sie angeben, wie die Tag-Daten übertragen werden sollen. Bei der Einstellung „Disabled“ (Deaktiviert) wird die Datenübertragung

verhindert. Bei der Einstellung „Periodic“ (Regelmäßig) werden Daten regelmäßig übertragen. Bei der Einstellung „Triggered“ (Ausgelöst) werden Daten auf der steigenden Flanke eines auslösenden Ausdrucks übertragen. Sofern Tag-Daten nicht aufgrund der Eigenschaft Force Update (Aktualisierung erzwingen) gesendet werden, werden Tag-Daten nur bei der ersten Verbindung gesendet oder wenn sie von mehr als einer Totzoneneigenschaft eines bestimmten Tags geändert wurden. Die spezifische Einstellung „Write Only“ (Nur Schreibvorgänge) verhindert, dass Updates aus den angegebenen Tags veröffentlicht werden, erlaubt ihnen jedoch weiterhin die Annahme von Schreibvorgängen. Dies kann hilfreich sein, wenn Sie vermeiden möchten, dass Schreibvorgänge sofort auf den Server übernommen werden.

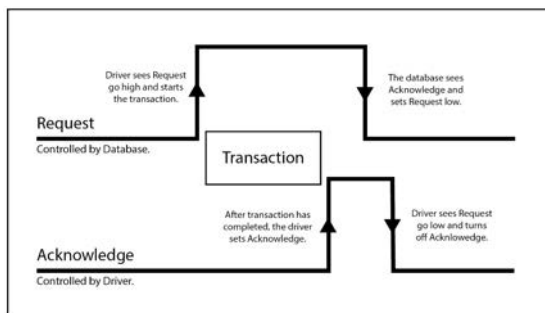
- Die Eigenschaft *Data Buffering* (Datenpufferung) dient der Aktivierung bzw. Deaktivierung der Datenpufferung für diesen Tag-Satz. Weitere Informationen hierzu finden Sie im Abschnitt zur Datenpufferung weiter unten in diesem Kapitel. Beachten Sie, dass die Datenpufferung auch auf der Registerkarte „Services“ (Dienste) aktiviert sein muss, damit diese Eigenschaft wirksam wird.
- Mit der Eigenschaft *Update Rate* (Aktualisierungshäufigkeit) wird gesteuert, wie oft die Tag-Daten an den Server gepusht werden, wenn der Modus „Periodic“ (Regelmäßig) ausgewählt ist. Unterschiedliche Tag-Datensätze werden oft unterschiedliche Einstellungen aufweisen, sodass wichtigere Daten häufiger übertragen werden können.
- Mit der Eigenschaft *Force Update* (Aktualisierung erzwingen) wird gewährleistet, dass der Wert der einzelnen Tags mindestens so häufig wie hier angegeben an den Server gepusht wird. Wenn die Einstellung Null übernommen wird, werden die Tag-Daten nur wie oben unter *Transfer Mode* (Übertragungsmodus) beschrieben gesendet.
- Die Eigenschaft *Request* (Anforderung) hat zwei Verwendungszwecke. Im Modus „Periodic“ (Regelmäßig) fungiert sie als pegelempfindliche Aktivierung der Übertragung, d. h. die Übertragung findet nur statt, wenn der Ausdruck *true* oder nicht definiert ist. Im Modus „Triggered“ (Ausgelöst) fungiert sie als flankenempfindlicher Auslöser für die Übertragung, d. h. der Vorgang wird bei einer steigenden Flanke ausgelöst.
- Mit der Eigenschaft *Acknowledge* (Bestätigen) wird im Modus „Triggered“ (Ausgelöst) Feedback zum Status der Übertragung gegeben. Sie sollte auf einen änderbaren Ausdruck gesetzt werden, der auf *true gesetzt* wird, wenn die Übertragung abgeschlossen ist, und *false*, wenn die Eigenschaft „Request“ (Anforderung) wieder in den Status *false* zurückgekehrt ist.
- Mit der Eigenschaft *Tag Writes* (Tag-Schreibvorgänge) wird angegeben, ob der Anschluss Schreibvorgänge vom Server an die Tags in diesem Satz akzeptieren soll. Die Methode, mit der der Server einen Schreibvorgang sendet, richtet sich nach dem jeweiligen Anschluss und wird im Folgenden noch ausführlicher beschrieben.
- Mit der Eigenschaft *Tag Naming* (Tag-Benennung) wird definiert, auf welche Weise Crimson die Bezeichnung für jedes Tag in der JSON, die es erstellt, generiert. Standardmäßig wird der Name des Tags verwendet, der im Navigationsbereich eingegeben wurde. Sie können jedoch auch die Beschreibung, Bezeichnung oder den Alias des Tags auswählen. Wenn eine lokalisierbare Quelle ausgewählt ist, wird die Standardübersetzung verwendet, unabhängig davon, welche Sprache für die Anzeige in der Benutzeroberfläche ausgewählt wurde.
- Mit der Eigenschaft *Label Set* (Satz bezeichnen) wird angegeben, ob die Tag-Daten aus diesem Satz in einem eindeutigen Objekt innerhalb der JSON-Datenstruktur platziert werden sollen. Weitere Informationen finden Sie in dem Abschnitt zum Thema Datenlayout.
- Mit der Eigenschaft *Tag Structure* (Tag-Struktur) wird angegeben, wie Tags in Crimson-Ordern in der JSON-Datenstruktur dargestellt werden sollen. Bei Auswahl von „Simple Tag List“ (Einfache Tag-Liste) ist jedes Tag als Stammebeneneigenschaft des Objekts *Tags* vorhanden und sein Name entspricht der mit Punkten versehenen Form seines Crimson-Namens. Bei Auswahl von „Folders with Short Names“ (Ordner mit Kurznamen) wird für jeden Crimson-Ordner ein JSON-Objekt erstellt und die entsprechenden Tags werden in diesem Objekt platziert. Der Name der einzelnen Tags entspricht dem letzten Teil seines Crimson-Namens. Bei Auswahl von „Folders with Full Names“ (Ordner mit vollständigen Namen) ist das

Verhalten ähnlich, aber der Name des Tags ist die vollständig mit Punkten versehene Form. Weitere Informationen finden Sie in dem Abschnitt zum Thema Datenlayout. Beachten Sie, dass diese Eigenschaft nicht bei Anschlüssen unterstützt wird, die nicht JSON für die Datendarstellung verwenden.

- Mit der Eigenschaft *Array Layout* (Array-Layout) wird angegeben, wie Array-Elemente in der JSON-Datenstruktur dargestellt werden sollen. Bei Auswahl der Option „Create Elements in List“ (Elemente in Liste erstellen) wird jedes Array-Element als ein eindeutiges Tag angezeigt und ist nach dem Tag und dem Elementindex benannt. Bei Auswahl der Option „Create Elements in Folder“ (Elemente in Ordner erstellen) wird jedes Array-Element als Eigenschaft in einem nach dem Tag benannten JSON-Objekt angezeigt. Weitere Informationen finden Sie in dem Abschnitt zum Thema Datenlayout. Beachten Sie, dass diese Eigenschaft nicht bei Anschlüssen unterstützt wird, die nicht JSON für die Datendarstellung verwenden.
- Mit der Eigenschaft *Send Properties* (Eigenschaften senden) wird angegeben, ob Crimson die verschiedenen mit dem Tag verknüpften Eigenschaften bereitstellen soll. Wenn diese Funktion aktiviert ist, wird ein JSON-Objekt für das Tag erstellt, und eine Teilmenge der Tag-Eigenschaften, die über den Punktoperator zugänglich sind, werden in dieses Objekt eingeschlossen. Außerdem wird eine Eigenschaft namens „Value“ (Wert) erstellt, um die aktuellen Tag-Daten aufzunehmen. Beachten Sie, dass der Anschluss die detaillierten Tag-Eigenschaften als schreibgeschützt behandelt und weder prüft, ob Änderungen vorliegen, noch Änderungen an den Server sendet. Für den Sparkplug-Anschluss ist diese Eigenschaft immer aktiviert und wird daher nicht angezeigt.

18.1.2 Modus Ausgelöst

Beim Betrieb im Modus „Triggered“ (Ausgelöst) wird die Datenübertragung für Gerätedaten oder Tags durch die Eigenschaften *Request* (Anforderung) und *Acknowledge* (Bestätigen) gesteuert. Das Verhalten ist weiter oben beschrieben, doch das folgende Diagramm macht den Sachverhalt deutlicher.



Wie Sie sehen, wird die Übertragung dadurch ausgelöst, dass die Datenbank *Request* (Anforderung) auf *true* setzt. Sobald die Übertragung abgeschlossen ist, setzt der Anschluss *Acknowledge* (Bestätigen) auf *true* und wartet, bis die Datenbank *Request* (Anforderung) auf *false* setzt. Sowie dies geschehen ist, setzt der Anschluss *Acknowledge* (Bestätigen) auf *false*, und der Zyklus kann von vorn beginnen.

18.1.3 Anschlussdiagnose

Sie können für jeden Anschluss ein numerisches Tag angeben, in das ein Wert geschrieben wird, der den Status des Anschlusses widerspiegelt. Dies ist eine einfache Form der Diagnose, anhand der Sie sicherstellen können, dass Sie erfolgreich mit der Cloud verbunden sind. Die Werte, die in das Tag eingefügt werden, lauten wie folgt.

Wert	Anschlusstatus
0	Fährt hoch oder stellt eine unterbrochene Verbindung wieder her.
1	Verbindung mit dem Server wird hergestellt, nachdem die IP-Adresse anhand des Hostnamens identifiziert wurde
2	Verbunden mit dem Server, aber wartet auf die primäre Anwendung
3	Verbunden mit dem Server und bereit, Daten zu übergeben
4	Verbunden mit dem Server und Datenupdates wurden veröffentlicht

Wie Sie sehen, wird die Übertragung dadurch ausgelöst, dass die Datenbank *Request (Anforderung)* auf *true* setzt. Sobald die Übertragung abgeschlossen ist, setzt der Anschluss *Acknowledge (Bestätigen)* auf *true* und wartet, bis die Datenbank *Request (Anforderung)* auf *false* setzt. Sowie dies geschehen ist, setzt der Anschluss *Acknowledge (Bestätigen)* auf *false*, und der Zyklus kann von vorn beginnen.

18.2 Datenpufferung

Die MQTT-Konnektoren von Crimson zur Datenpufferung können so konfiguriert werden, dass Daten im Gerät gepuffert werden, wenn keine Verbindung zum Server verfügbar ist. Ist diese Option nicht konfiguriert, erfolgen keine Abtastungen, wenn der Dienst offline ist. Sobald wieder eine Verbindung hergestellt ist, werden die aktuellsten Werte per Push-Vorgang an den Server gesendet. Die Datenpufferung wird über die gleichnamige Eigenschaft auf der Registerkarte „Service“ (Dienst) des Konnektors gesteuert und kann so eingestellt werden, dass die Daten entweder nur im Speicher gepuffert werden oder dass über die Speicherkarte des Geräts weiterer Speicher bereitgestellt wird, der auch beim Aus- und Einschalten des Geräts erhalten bleibt. Im speicherbasierten Modus speichert der Konnektor bis zu 14.400 Werte, d. h. Daten aus einem Zeitraum von 4 Stunden bei einer Abtastung pro Sekunde. Im festplattenbasierten Modus versucht der Konnektor, so lange zu puffern, bis der Speicherplatz aufgebraucht ist.

Wenn die Datenpufferung aktiviert ist, besitzen Nachrichten, die an den Server gesendet werden, die zusätzlichen Eigenschaften *adhoc* und *historic* (Ad-hoc bzw. historisch). Die Eigenschaft *adhoc* ist *true* (wahr) für Daten, die außerhalb des üblichen Abtastzeitplans erfasst wurden. Die Eigenschaft *historic* ist *true* (wahr) für Daten, die aus dem Datenpuffer wiedergegeben werden.

Im Normalbetrieb ist sowohl die Eigenschaft *adhoc* als auch die Eigenschaft *historic false* (falsch). Es kann aber eine Situation auftreten, in der ein Gerät für die minütliche Datenspeicherung konfiguriert ist, die Verbindung zum Server aber unterbrochen und erst nach einer Pause von zwei Stunden wiederhergestellt wird. Unmittelbar nach der Wiederherstellung der Verbindung gibt der Konnektor alle ungefähr 120 gespeicherten historischen Datensätze wieder, wobei jeder Datensatz mit *historic* markiert wird, damit das Backend beispielsweise weiß, dass es keine Mimics oder anderen Statusanzeigen mit veralteten Daten aktualisieren soll. Sobald die historischen Daten wiedergegeben wurden, sendet der Konnektor umgehend einen einzelnen Messwert mit Live-Daten. Normalerweise werden die Messwerte nur auf die Minute angezeigt, dieser Messwert jedoch wird sofort gesendet und als *Ad-hoc*-Messwert markiert. Wenn Sie Daten in einer Datenbank speichern und nur solche Daten benötigen, die in dem Intervall erfasst wurden, das durch den Abtastzeitraum definiert ist, können Sie mit dem Flag *adhoc* Werte ignorieren, die außerhalb des Zeitplans erfasst werden. Dennoch sollten diese Werte an alle Mimics oder Statusanzeigen weitergegeben werden, da sichergestellt werden muss, dass so schnell wie möglich neue Daten verfügbar sind.

Bei JSON-basierten Konnektoren werden die Eigenschaften *adhoc* und *historic* zusammen mit dem Zeitstempel in der JSON codiert. Für Sparkplug wird die Eigenschaft *adhoc* nicht unterstützt. Die Eigenschaft *historic* hingegen wird mithilfe des Flags „*MetricIsHistorical*“ (Messwert ist historischer Wert) des Protokolls codiert.

18.3 Datenlayout in JSON

Anschlüsse, die JSON für die Datenübertragung verwenden, erstellen ein JSON-Fragment, das die Geräte- und Tag-Daten enthält. Das Fragment wird in der Regel als Objekt in eine größere JSON-Struktur eingebettet, die durch die Anforderungen des Anschlussprotokolls definiert wird. Bei den nachfolgenden Beschreibungen wird davon ausgegangen, dass die Eigenschaft *Tag Naming* (Tag-Benennung) auf „Name“ gesetzt ist. Wenn eine andere Benennungsquelle verwendet wird, wird die JSON entsprechend dieser Einstellung geändert. Die Gruppierung nach Ordnern erfolgt dann nur, wenn die Benennungsquelle Werte mit eingebetteten Punkten enthält.

Die unten gezeigte Abbildung enthält ein typisches JSON-Fragment.

```
"device": {
  "status": "okay"
  "cellular": {
    "valid": "false"
  },
  "location": {
    "valid": "false"
  }
},
"tags": {
  "Tag1": 0,
  "Tag2": 0,
  "Tag3": 0,
  "Tag4": 0
}
```

```
"timestamp" : "2018-12-05T18 : 14 :52 . 000Z"
```

Je nachdem, wie Sie sie anzeigen, befinden sich die Objekte und Eigenschaften möglicherweise in einer anderen Reihenfolge. Die zugrunde liegende Struktur bleibt jedoch gleich. Wie Sie sehen, enthält das Fragment zwei Objekte: Das erste ist das Gerät, das zweite sind die Tags. Hinzu kommt ein Wert, der angibt, dass die Verbindung gültig ist, und ein Zeitstempel für die berichteten Daten.

18.3.1 Anschlussoptionen

Standardmäßig codiert Crimson True-or-False-Werte als Zeichenfolgen, anstatt das spezifische boolesche Format von JSON zu verwenden. Dies kann jedoch mithilfe der Eigenschaft *Data Encoding* (Datencodierung) der Registerkarte „Service“ (Dienst) des Konnektors außer Kraft gesetzt werden. Wenn Sie diese Eigenschaft von „Standard“ in „Use Booleans“ (Boolesche Werte verwenden) ändern, werden die Anführungszeichen aus den True-or-False-Werten entfernt. Wenn Sie sie hingegen in „Numeric Only“ (Nur numerisch) ändern, wird „true“ mit 1 und „false“ mit 0 codiert. Diese letzte Option ist nur im generischen MQTT-Konnektor verfügbar und wird in der Regel für Plattformen wie z. B. Ubidots verwendet, die ausschließlich numerische Datenwerte verarbeiten.

18.3.2 Verbindungsstatus

Wie oben erwähnt, enthält die JSON, die vom Konnektor gesendet wird, eine Eigenschaft namens *connected* (Verbunden), die in normalen Nachrichten immer auf *true* gesetzt ist. MQTT unterstützt einen Mechanismus namens „Last Will and Testament“ (LWT; „Letzter Wille“), mit dem ein Gerät eine Nachricht registrieren kann, die der Broker dann bei einem Verbindungsabbruch in einem bestimmten Thema veröffentlicht. Die JSON-basierten Konnektoren registrieren in der Regel eine LWT-Nachricht, die nur die Eigenschaft *connected* und den Wert *false* enthält. Auf diese Weise wird sichergestellt, dass der Gerätezwilling oder -schatten den Wert „connected“ enthält,

der den Verbindungsstatus des Geräts genau darstellt, und dass telemetriebasierte Anwendungen eine Nachricht erhalten, mit der sie in ähnlicher Weise einen Verbindungsabbruch erkennen können.

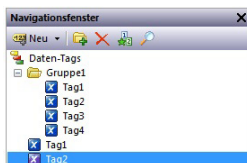
18.3.3 Gerätedaten

Das Objekt `device` enthält eine Eigenschaft namens `status`, die immer auf `okay` gesetzt ist, sowie Objekte namens `cellular` und `location`. Diese beiden Objekte wiederum enthalten eine Eigenschaft namens `valid`, die derzeit auf `false` gesetzt ist, weil die Objekte noch nicht funktionsfähig sind.

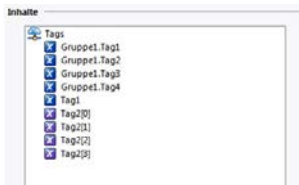
18.3.4 Tagdaten

In diesem Beispiel enthält das Objekt `tags` vier Tags ohne Ordnerstruktur. Dies stellt eine einfache Datenbank dar, bei der diese vier Tags in der obersten Ebene der Tag-Liste konfiguriert sind. Für komplexere Strukturen gibt es verschiedene Möglichkeiten, zu steuern, wie die Tags angezeigt werden.

Betrachten Sie eine Datenbank mit der folgenden Tag-Struktur.



Hier befinden sich vier Tags in einem Ordner und zwei weitere Tags auf der obersten Ebene. Das zweite dieser beiden Tags ist ein Array. In unserer Anschlusskonfiguration ordnen wir all diese Tags wie unten gezeigt den ersten Tag-Datensätzen zu, darunter vier Elemente aus dem Array.



Bei den Standardeinstellungen entsteht hierdurch das unten abgebildete JSON-Objekt `tags`.

```
"tags": {
  "Group1.Tag1": 0,
  "Group1.Tag2": 0,
  "Group1.Tag3": 0,
  "Group1.Tag4": 0,
  "Tag1": 0,
  "Tag2[0]": 0,
  "Tag2[1]": 0,
  "Tag2[2]": 0,
  "Tag2[3]": 0
}
```

Hier sind die vier Tags aus dem Ordner mit ihren mit Punkten versehenen Namen codiert. Das einfache Tag auf der obersten Ebene ist mit seinem einfachen Namen codiert und die vier Array-Elemente sind mit Namen codiert, die das Tag und den Elementindex darstellen.

Beachten Sie, dass Punkte oder eckige Klammern in Tag-Namen von einigen Cloud-Services nicht unterstützt werden. In diesem Fall werden die Punkte und die öffnende eckige Klammer durch Bindestriche ersetzt und die schließende Klammer wird entfernt. Das resultierende JSON-Fragment sieht wie folgt aus.

```
"tags": {
  "Group1-Tag1": 0,
  "Group1-Tag2": 0,
  "Group1-Tag3": 0,
  "Group1-Tag4": 0,
  "Tag1": 0,
  "Tag2-0": 0,
  "Tag2-1": 0,
  "Tag2-2": 0,
  "Tag2-3": 0
}
```

Wie oben erwähnt, können Sie mit der Eigenschaft *Tag Structure* (Tag-Struktur) steuern, wie Tags in Ordnern in JSON dargestellt werden. Das oben gezeigte Beispiel zeigt die Standardeinstellung „Simple Tag List“ (Einfache Tag-Liste). Wenn die Option „Folders with Short Names“ (Ordner mit Kurznamen) ausgewählt ist, sieht JSON wie folgt aus.

```
"tags": {
  "Group1": {
    "Tag1": 0,
    "Tag2": 0,
    "Tag3": 0,
    "Tag4": 0
  },
  "Tag1": 0,
  "Tag2[0]": 0,
  "Tag2[1]": 0,
  "Tag2[2]": 0,
  "Tag2[3]": 0
}
```

Wie Sie sehen, befinden sich die Tags aus dem Ordner jetzt in einem JSON-Objekt, das nach diesem Ordner benannt ist, wobei die Eigenschaften im Objekt die letzten Teile des Tag-Namens bilden. Wenn die Option „Folders with Full Names“ (Ordner mit vollständigen Namen) ausgewählt ist, sieht JSON wie folgt aus.

```
"tags": {
  "Group1": {
    "Group1.Tag1": 0,
    "Group1.Tag2": 0,
    "Group1.Tag3": 0,
    "Group1.Tag4": 0
  },
  "Tag1": 0,
  "Tag2[0]": 0,
  "Tag2[1]": 0,
  "Tag2[2]": 0,
  "Tag2[3]": 0
}
```

Hier ist die Objektstruktur identisch, doch die Eigenschaften im Ordnerobjekt werden jetzt nach dem vollständigen mit Punkten versehenen Namen des entsprechenden Tags benannt. Dies ist weniger effizient als die kurze Namensform, wurde aber von bestimmten Kunden für ihre Anwendungen angefordert.

Wie oben erwähnt, können Sie auch mit der Eigenschaft *Array Layout* (Array-Layout) ändern, wie Array-Elemente in JSON dargestellt werden. Die bisherigen Beispiele wurden alle mit der Einstellung der Eigenschaft auf „Create Elements in List“ (Elemente in Liste erstellen) erstellt. Wenn wir die Eigenschaft *Tag Structure* (Tag-Struktur) auf „Folders with Short Names“ (Ordner mit Kurznamen) zurücksetzen und die Eigenschaft *Array Layout* (Array-Layout) auf „Create Elements in Folder“ (Elemente im Ordner erstellen) setzen, wird JSON wie folgt geändert.

```
"tags": {
  "Group1": {
    "Tag1": 0,
    "Tag2": 0,
    "Tag3": 0,
    "Tag4": 0
  },
  "Tag1": 0,
  "Tag2": {
    "0": 0,
    "1": 0,
    "2": 0,
    "3": 0
  }
}
```

Hier wurde ein JSON-Objekt für das Array erstellt und in diesem Objekt wurden Eigenschaften erstellt, um die einzelnen Elemente darzustellen. Dies kann eine natürlichere Darstellung von Arrays innerhalb Ihrer Anwendung sein.

In allen oben genannten Beispielen war die Einstellung *Send Properties* (Eigenschaften senden) deaktiviert. Wenn sie aktiviert ist, wird unter jedem Tag oder Array-Element eine umfangreichere RAW-Struktur erstellt. Das JSON-Fragment unten zeigt, wie ein Tag unter solchen Umständen dargestellt werden kann. Die Eigenschaften werden im Kapitel „Schreiben von Ausdrücken“ im Abschnitt „Tag-Eigenschaften“ beschrieben. Beachten Sie, wie sich das Tag von einem JSON-Wert in ein Objekt geändert hat, wobei die tatsächlichen Daten in der Eigenschaft „Value“ (Wert) gespeichert sind.

```
"tags": {
  "Group1": {
    "Tag1": {
      "Deadband": 0,
      "Description": "This is Tag1",
      "Label": "Group1.Tag1",
      "Maximum": 1234,
      "Minimum": 0,
      "Name": "Group1.Tag1",
      "Prefix": "",
      "TextOff": "",
      "TextOn": "",
      "Units": "",
      "Value": 22
    }
  }
}
```

Es gibt noch eine letzte Möglichkeit, das Objekt `tags` in der JSON-Struktur zu ändern, nämlich durch das Bezeichnen von Tag-Datensätzen. Standardmäßig werden alle Tag-Daten direkt unter dem Objekt `tags` platziert, selbst wenn diese Daten aus mehreren Tag-Datensätzen stammen. Wenn Sie die Eigenschaft *Label Set* (Satz bezeichnen) auf „Yes“ (Ja) setzen, wird ein Zwischenobjekt erstellt, das nach dem Satz unter dem Objekt `tags` benannt wird. JSON sieht dann ungefähr so aus.


```
"tags": {  
  "set1": {  
    "Group1.Tag1": 0,  
    "Group1.Tag2": 0,  
    "Group1.Tag3": 0,  
    "Group1.Tag4": 0  
  }  
}
```

Hier weist das Objekt `set1` darauf hin, dass all diese Tags aus der Registerkarte „Tag Data 1“ stammen.

18.3.5 Site-Benennung

Bei einigen Anschlüssen ist es möglich, auf der Registerkarte „Service“ (Dienst) eine Eigenschaft *Site Name* (Site-Name) festzulegen. Wenn dieser Wert definiert ist, wird er als Eigenschaft namens `cid` unter jedem Tag-Satz und unter dem Objekt `device` eingefügt. Der Wert wird bei allen Aktualisierungen berücksichtigt, unabhängig davon, ob er geändert wurde. Er ist darauf ausgelegt, dem Back-End zu erleichtern, die einzelnen Meldungen dem Gerät zuzuordnen, von dem sie stammen. Unten sehen Sie ein Beispiel einer JSON-Struktur mit der Eigenschaft `cid`.

```
"device": {  
  "cid": "site-name"  
  "status": "okay"  
  "cellular": {  
    "valid": "false"  
  },  
  "location": {  
    "valid": "false"  
  }  
},  
"tags": {  
  "set1": {  
    "cid": "site-name",  
    "Tag1": 0,  
    "Tag2": 0,  
    "Tag3": 0,  
    "Tag4": 0  
  }  
}
```

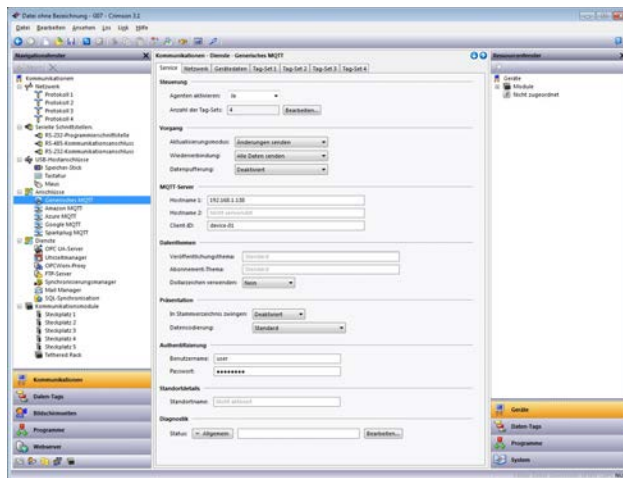
Hier wird der für den Wert *Site Name* (Site-Name) angegebene Wert `site-name` auf der Geräte- und Tag-Satz-Ebene eingeschlossen.

18.4 Der generische MQTT-Anschluss

Der generische MQTT-Anschluss ist darauf ausgelegt, Daten im JSON-Format in einem beliebigen MQTT-Thema zu veröffentlichen und durch Abonnieren eines anderen beliebigen Themas auf Schreibvorgänge zu warten. Es ist nicht für den Betrieb mit einem bestimmten Cloudanbieter optimiert, bietet jedoch verschiedene Funktionen, die die Integration in Status Enterprise von B-Scada unterstützen.

18.4.1 Anschlusseinstellungen

Die Einstellungen für den Anschluss sind unten abgebildet.



- Mit der Eigenschaft *Enable Agent* (Agenten aktivieren) wird dieser Anschluss aktiviert bzw. deaktiviert.
- Mit der Eigenschaft *Tag Set Count* (Anzahl an Tag-Sätzen) kann die Anzahl der vom Konnektor unterstützten Tag-Sätze geändert werden. Es können bis zu 30 Tag-Sätze erstellt werden. Änderungen an diesem Wert können nicht rückgängig gemacht werden. Der Puffer für Rückgängigmachungen wird geleert.
- Mit der Eigenschaft *Update Mode* (Aktualisierungsmodus) wird angegeben, welche Datenwerte in den jeweiligen Aktualisierungsnachrichten gesendet werden sollen. Mit „Send Changes“ (Änderungen senden) werden nur die Werte gesendet, die sich seit dem letzten Senden von Daten um mehr als ihre Totzone geändert haben. Mit „Send All If Any“ (Alle senden, wenn vorhanden) werden alle Datenwerte aus einem Tag-Satz gesendet, sofern sich Werte in diesem Satz um mehr als die Totzone geändert haben. Mit „Send All Data“ (Alle Daten senden) werden alle Daten zu jedem Aktualisierungsereignis gesendet, unabhängig davon, ob Änderungen aufgetreten sind.
- Mit der Eigenschaft *Reconnection* (Wiederverbindung) wird festgelegt, wie sich der Konnektor verhalten soll, wenn eine Verbindung getrennt und wiederhergestellt wird. Mit der Option „Send All Data“ (Alle Daten senden) werden die Verlaufswerte zurückgesetzt, die zum Erkennen von Änderungen verwendet werden, und dadurch werden alle Daten erneut gesendet. Im „Update Mode“ (Aktualisierungsmodus) wird die Aktualisierung der Wiederverbindung wie eine normale Aktualisierung behandelt und daher das gleiche Verhalten angewendet wie von der Eigenschaft *Update Mode* definiert.
- Mit der Eigenschaft *Data Buffering* (Datenpufferung) kann die Datenpufferung aktiviert bzw. inaktiviert werden. Weitere Informationen hierzu finden Sie im entsprechenden Abschnitt weiter oben in diesem Kapitel. Beachten Sie, dass hierfür die Datenpufferung auch auf Ebene des Tag-Satzes aktiviert sein muss.
- Mit den Eigenschaften *Host Name 1* und *Host Name 2* werden die Namen der MQTT-Server angegeben, auf denen der Anschluss Daten veröffentlicht und von denen er Abonnementupdates empfängt. Sie können als IP-Adresse oder als tatsächlicher Hostname angegeben werden, doch Verbindungen mit TLS

erfordern in der Regel einen Namen, wenn das Serverzertifikat validiert werden soll. Der Anschluss versucht zuerst, eine Verbindung mit *Host Name 1* herzustellen. Wenn dies nicht gelingt, versucht er es mit *Host Name 2*. Wenn keiner der Hosts reagiert, wartet der Anschluss bis zum Ende der Back-off-Zeit und beginnt dann den Zyklus von vorn.

- Die Eigenschaft *Client ID* (Client-ID) dient zur Angabe der Client-ID, die in die MQTT-Frames eingefügt wird. Die genaue Interpretation dieses Werts richtet sich nach dem Server, mit dem Sie eine Verbindung herstellen. Der Wert wird auch zum Erstellen der Standardthemenamen verwendet, sofern diese nicht angegeben sind.
- Unter *Publication Topic* (Publikationsthema) wird das MQTT-Thema angegeben, in dem der Anschluss ein JSON-Fragment mit den Geräte- und Tag-Daten veröffentlicht. Wenn Sie keinen Wert angeben, wird `§crimson/generic/device-id/pub` verwendet, wobei der Teil `device-id` durch die Eigenschaft *ClientID* (Client-ID) ersetzt wird. Das Dollarzeichen wird ausgelassen, wenn die Eigenschaft *Drop Dollar Sign* (Dollarzeichen auslassen) auf „Yes“ (Ja) gesetzt wurde.
- Unter *Subscription Topic* (Abonnement-Thema) wird das MQTT-Thema angegeben, das der Anschluss abonniert, um JSON-Fragmente zu empfangen, die Tag-Schreibvorgänge enthalten. Wenn Sie keinen Wert angeben, wird `§crimson/generic/device-id/sub` verwendet, wobei der Teil `device-id` durch die Eigenschaft *ClientID* (Client-ID) ersetzt wird. Das Dollarzeichen wird ausgelassen, wenn die Eigenschaft *Drop Dollar Sign* (Dollarzeichen auslassen) auf „Yes“ (Ja) gesetzt wurde.
- Mit der Eigenschaft *Use Dollar Sign* (Dollarzeichen verwenden) wird der Anschluss angewiesen, das hinzufügen einer Dollarzeichen aus den standardmäßigen Publikations- und Abonnementthemen wegzulassen. Bei bestimmten Brokern wie ActiveMQ ist das anfängliche Dollarzeichen problematisch.
- Mit der Eigenschaft *Force Into Root* (Verschiebung in den Stamm erzwingen) kann die gesamte JSON-Struktur so reduziert werden, dass alle Werte im Stammobjekt enthalten sind. Dies wurde für die Verwendung mit Servern wie z. B. Ubidots konzipiert, die nur sehr einfache JSON-Analysen ausführen.
- Mit der Eigenschaft *Data Encoding* (Datencodierung) wird gesteuert, wie boolesche Werte in der JSON dargestellt werden, die an den Server gesendet wird. Das Verhalten wird weiter oben in diesem Kapitel beschrieben. Der generische Konnektor unterstützt die zusätzliche Einstellung „Numeric Only“ (Nur numerisch), mit der die Umwandlung aller Datenelemente in die numerische Form erzwungen wird. Auch dies wurde für die Verwendung mit Servern wie z. B. Ubidots konzipiert, die nur sehr einfache JSON-Analysen ausführen.
- Mit den Eigenschaften *User Name* (Benutzername) und *Password* (Kennwort) werden die Authentifizierungsinformationen angegeben, die beim Herstellen der MQTT-Verbindung angegeben werden müssen.
- Mit der Eigenschaft *Site Name* (Site-Name) wird eine Zeichenfolge angegeben, die in jedem auf dem Server veröffentlichten Update enthalten ist, sodass die Back-End-Anwendung die Quelle der Aktualisierung leichter identifizieren kann.
- Mit der Eigenschaft *Status* wird das numerische Tag angegeben, in das Informationen zum Anschlussstatus geschrieben werden. Weitere Informationen finden Sie in der Tabelle weiter oben in diesem Kapitel.

18.4.2 Anschlussbetrieb

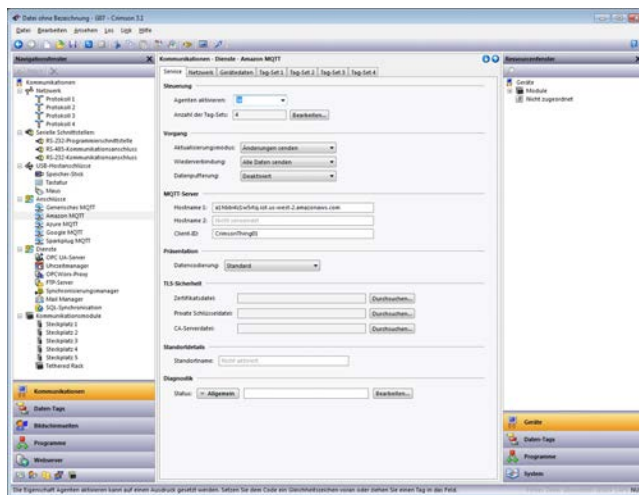
Der Anschluss veröffentlicht JSON-Fragmente mit Geräte- und Tag-Daten in dem *Publication Topic* (Publikationsthema), das bei der Konfiguration festgelegt wurde. Die Fragmente werden wie oben beschrieben formatiert, wobei sich die Objekte `tags` und `device` auf der obersten Ebene des Fragments befinden. Der Anschluss nimmt auch Schreibvorgänge an Crimson-Tags an, sofern dies in der Konfiguration festgelegt wurde. Hierzu abonniert er das in der Konfiguration angegebene *Subscription Topic* (Abonnement-Thema) und erwartet JSON-Fragmente, die auf die gleiche Weise formatiert sind wie die Fragmente, die er veröffentlicht. Zum Beispiel schreibt ein einfaches Fragment von `{ "tags": { "Tag1": 100 } }` den Wert 100 in Tag1, vorausgesetzt, Tag1 ist in einem Tag-Datensatz enthalten, der laut Konfiguration Schreibvorgänge zulässt.

18.5 Der Amazon MQTT-Anschluss

Der Amazon MQTT-Anschluss ist ein spezieller Anschluss für die Veröffentlichung von Tag- und Gerätedaten im Abschnitt *reported* des Schattenobjekts, das für ein Ding in der AWS IoT-Umgebung erstellt wurde. Er abonniert außerdem Änderungen am Abschnitt *desired*, sodass Schreibvorgänge an Tags weitergeleitet werden können, die zur Annahme von Änderungen konfiguriert sind. Der Anschluss implementiert TLS-Sicherheit mithilfe der Zertifikate, die bei der Erstellung des AWS IoT-Dings verfügbar gemacht wurden.

18.5.1 Anschlusseinstellungen

Die Einstellungen für den Anschluss sind unten abgebildet.



- Mit der Eigenschaft *Enable Agent* (Agenten aktivieren) wird dieser Anschluss aktiviert bzw. deaktiviert.
- Mit der Eigenschaft *Tag Set Count* (Anzahl an Tag-Sätzen) kann die Anzahl der vom Konnektor unterstützten Tag-Sätze geändert werden. Es können bis zu 30 Tag-Sätze erstellt werden. Änderungen an diesem Wert können nicht rückgängig gemacht werden. Der Puffer für Rückgängigmachungen wird geleert.
- Mit der Eigenschaft *Update Mode* (Aktualisierungsmodus) wird angegeben, welche Datenwerte in den jeweiligen Aktualisierungsnachrichten gesendet werden sollen. Mit „Send Changes“ (Änderungen senden) werden nur die Werte gesendet, die sich seit dem letzten Senden von Daten um mehr als ihre Totzone geändert haben. Mit „Send All If Any“ (Alle senden, wenn vorhanden) werden alle Datenwerte aus einem Tag-Satz gesendet, sofern sich Werte in diesem Satz um mehr als die Totzone geändert haben. Mit „Send All Data“ (Alle Daten senden) werden alle Daten zu jedem Aktualisierungsereignis gesendet, unabhängig davon, ob Änderungen aufgetreten sind.
- Mit der Eigenschaft *Reconnection* (Wiederverbindung) wird festgelegt, wie sich der Konnektor verhalten soll, wenn eine Verbindung getrennt und wiederhergestellt wird. Mit der Option „Send All Data“ (Alle Daten senden) werden die Verlaufswerte zurückgesetzt, die zum Erkennen von Änderungen verwendet werden, und dadurch werden alle Daten erneut gesendet. Im „Update Mode“ (Aktualisierungsmodus) wird die Aktualisierung der Wiederverbindung wie eine normale Aktualisierung behandelt und daher das gleiche Verhalten angewendet wie von der Eigenschaft „Update Mode“ definiert.
- Mit der Eigenschaft *Data Buffering* (Datenpufferung) kann die Datenpufferung aktiviert bzw. inaktiviert werden. Weitere Informationen hierzu finden Sie im entsprechenden Abschnitt weiter oben in diesem Kapitel. Beachten Sie, dass hierfür die Datenpufferung auch auf Ebene des Tag-Satzes aktiviert sein muss.
- Mit den Eigenschaften *Host Name 1* und *Host Name 2* werden die Namen der MQTT-Server angegeben, auf denen der Anschluss Daten veröffentlicht und von denen er Abonnementupdates

empfängt. *Host Name 1* muss mit dem Namen übereinstimmen, der in der AWS IoT-Konsole auf der Interact-Seite der Ding-Informationen bereitgestellt wurde. *Host Name 2* hingegen wird nur in Topologien mit Redundanz eingesetzt. Der Anschluss versucht zuerst, eine Verbindung mit *Host Name 1* herzustellen. Wenn dies nicht gelingt, versucht er es mit *Host Name 2*. Wenn keiner der Hosts reagiert, wartet der Anschluss bis zum Ende der Back-off-Zeit und beginnt dann den Zyklus von vorn.

- Mit der Eigenschaft *Client ID* (Client-ID) wird der Name des Dings angegeben.
- Mit der Eigenschaft *Data Encoding* (Datencodierung) wird gesteuert, wie boolesche Werte in der JSON dargestellt werden, die an den Server gesendet wird. Das Verhalten wird weiter oben in diesem Kapitel beschrieben.
- Mit den Eigenschaften *Certificate File* (Zertifikatdatei), *Private Key File* (Private Schlüsseldatei) und *Server CA File* (ZS-Serverdatei) werden die drei Dateien angegeben, die von der AWS IoT-Konsole heruntergeladen wurden, als das Ding erstellt wurde. Die private Schlüsseldatei ist nicht erforderlich.
- Mit der Eigenschaft *Site Name* (Site-Name) wird eine Zeichenfolge angegeben, die in jedem auf dem Server veröffentlichten Update enthalten ist, sodass die Back-End-Anwendung die Quelle der Aktualisierung leichter identifizieren kann.
- Mit der Eigenschaft *Status* wird das numerische Tag angegeben, in das Informationen zum Anschlussstatus geschrieben werden. Weitere Informationen finden Sie in der Tabelle weiter oben in diesem Kapitel.

18.5.2 Anschlussbetrieb

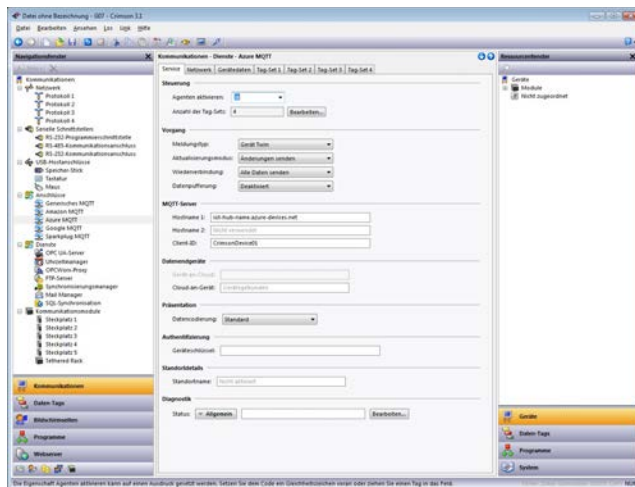
Der Anschluss veröffentlicht JSON-Fragmente mit Geräte- und Tag-Daten im Abschnitt `reported` innerhalb des Schattens des angegebenen Dings. Wenn im Abschnitt `desired` des Schattens ein entsprechender Tag-Wert erstellt wird, aktualisiert der Anschluss das Tag, wenn dieses Tag in einem Datensatz enthalten ist, der Schreibvorgänge zulässt. Dann entfernt er den Wert „desired“ aus dem Schatten.

18.6 Der Azure MQTT-Anschluss

Der Azure MQTT-Konnektor ist ein spezieller Konnektor für die Veröffentlichung von Tag- und Gerätedaten im berichteten Abschnitt des Zwillingobjekts, das für ein Gerät in der Azure IoT-Umgebung erstellt wurde, oder im Thema „Events“ (Ereignisse) des Geräts. Beim Betrieb im Zwillingmodus liest der Konnektor den gewünschten Abschnitt des Zwilling bei der ersten Verbindungsherstellung und überwacht ihn auf Änderungen, wobei Schreibvorgänge an Tags gesendet werden können, die für die Annahme von Änderungen konfiguriert sind. Im Telemetriemodus werden die Daten stattdessen auf dem Ereignisendpunkt des Geräts veröffentlicht. Unabhängig vom ausgewählten Modus abonniert der Konnektor den gerätegebundenen Endpunkt des Geräts und verarbeitet erneut Schreibvorgänge in Tags, die für die Annahme konfiguriert sind. Jeder Name eines Endpunkts kann geändert werden, wenn es die Anwendung erfordert.

18.6.1 Anschlusseinstellungen

Die Einstellungen für den Anschluss sind unten abgebildet.



- Mit der Eigenschaft *Enable Agent* (Agenten aktivieren) wird dieser Anschluss aktiviert bzw. deaktiviert.
- Mit der Eigenschaft *Tag Set Count* (Anzahl an Tag-Sätzen) kann die Anzahl der vom Konnektor unterstützten Tag-Sätze geändert werden. Es können bis zu 30 Tag-Sätze erstellt werden. Änderungen an diesem Wert können nicht rückgängig gemacht werden. Der Puffer für Rückgängigmachungen wird geleert.
- Mit der Eigenschaft *Message Type* (Nachrichtentyp) wird der Typ für die Nachrichten angegeben, die der Konnektor senden soll. Im Modus „Device Twin“ (Gerätezwillingmodus) wird der berichtete Status an den Azure-Zwilling gesendet und es wird nach Änderungen am gewünschten Status gesucht. Im Telemetriemodus werden Telemetriemessages gesendet und empfangen; eine Zwillingunterstützung durch Azure ist nicht erforderlich.
- Mit der Eigenschaft *Update Mode* (Aktualisierungsmodus) wird angegeben, welche Datenwerte in den jeweiligen Aktualisierungsnachrichten gesendet werden sollen. Mit „Send Changes“ (Änderungen senden) werden nur die Werte gesendet, die sich seit dem letzten Senden von Daten um mehr als ihre Totzone geändert haben. Mit „Send All If Any“ (Alle senden, wenn vorhanden) werden alle Datenwerte aus einem Tag-Satz gesendet, sofern sich Werte in diesem Satz um mehr als die Totzone geändert haben. Mit „Send All Data“ (Alle Daten senden) werden alle Daten zu jedem Aktualisierungsereignis gesendet, unabhängig davon, ob Änderungen aufgetreten sind.
- Mit der Eigenschaft *Reconnection* (Wiederverbindung) wird festgelegt, wie der Anschluss sich verhalten soll, wenn eine Verbindung getrennt und wiederhergestellt wird. Mit der Option „Send All Data“ (Alle Daten senden) werden die Verlaufswerte zurückgesetzt, die zum Erkennen von Änderun-

gen verwendet werden, und dadurch werden alle Daten erneut gesendet. Im „Update Mode“ (Aktualisierungsmodus) wird die Aktualisierung der Wiederverbindung wie eine normale Aktualisierung behandelt und daher das gleiche Verhalten angewendet wie von der Eigenschaft Update Mode (Aktualisierungsmodus) definiert.

- Mit der Eigenschaft *Data Buffering* (Datenpufferung) kann die Datenpufferung aktiviert bzw. inaktiviert werden. Weitere Informationen hierzu finden Sie im entsprechenden Abschnitt weiter oben in diesem Kapitel. Beachten Sie, dass hierfür die Datenpufferung auch auf Ebene des Tag-Satzes aktiviert sein muss.
- Mit den Eigenschaften *Host Name 1* und *Host Name 2* werden die Namen der MQTT-Server angegeben, auf denen der Anschluss Daten veröffentlicht und von denen er Abonnementupdates empfängt. *Host Name 1* muss mit dem Hostnamen des IoT-Hubs übereinstimmen, der in Azure erstellt wurde. *Host Name 2* hingegen wird nur in Topologien mit Redundanz eingesetzt. Der Anschluss versucht zuerst, eine Verbindung mit *Host Name 1* herzustellen. Wenn dies nicht gelingt, versucht er es mit *Host Name 2*. Wenn keiner der Hosts reagiert, wartet der Anschluss bis zum Ende der Back-off-Zeit und beginnt dann den Zyklus von vorn.
- Mit der Eigenschaft *Host Name* (Hostname) wird der Name des MQTT-Servers angegeben, auf dem der Konnektor Daten veröffentlicht und von dem er Abonnementupdates empfängt. Er sollte mit dem Hostnamen des in Azure erstellten IoT-Hubs übereinstimmen.
- Mit der Eigenschaft *Client ID* (Client-ID) wird der Name des IoT-Geräts angegeben.
- Mit der Eigenschaft *Device-to-Cloud* (Gerät-zu-Cloud) wird der Name des Endpunkts angegeben, an den Daten im Telemetriemodus gesendet werden. Der vollständige MQTT-Themenname lautet „device/device-id/messages“ (gerät/geräte-id/nachrichten), gefolgt vom angegebenen Endpunktnamen, wobei die Geräte-ID durch die oben angegebene Eigenschaft „Client ID“ (Client-ID) ersetzt wird. Wird kein Wert eingegeben, so wird der Standardendpunktname „events“ (Ereignisse) verwendet.
- Mit der Eigenschaft *Cloud-to-Device* (Cloud-zu-Gerät) wird der Name des Endpunkts angegeben, bei dem der Konnektor Benachrichtigungen über Schreibvorgänge abonniert. Im Zwillingsmodus wird auch der Gerätezwilling auf Änderungen überwacht. Der vollständige MQTT-Themenname lautet „device/device-id/messages“ (gerät/geräte-id/nachrichten), gefolgt vom angegebenen Endpunktnamen, wobei die Geräte-ID durch die oben angegebene Eigenschaft „Client ID“ (Client-ID) ersetzt wird. Wird kein Wert eingegeben, so wird der Standardendpunktname „devicebound“ (gerätegebunden) verwendet.
- Mit der Eigenschaft *Data Encoding* (Datencodierung) wird gesteuert, wie boolesche Werte in der JSON dargestellt werden, die an den Server gesendet wird. Das Verhalten wird weiter oben in diesem Kapitel beschrieben.
- Die *Eigenschaft Device Key* (Geräteschlüssel) wird zur Authentifizierung bei Azure verwendet und muss auf den Primary Key (Primärschlüssel) gesetzt werden, der auf der Seite „Device Details“ des zugehörigen Geräts in der Azure Management Console angezeigt wird. Der Anschluss erstellt mit diesem Schlüssel jedes Mal, wenn er eine Verbindung herstellt, ein entsprechendes Zugriffstoken.
- Mit der Eigenschaft *Site Name* (Site-Name) wird eine Zeichenfolge angegeben, die in jedem auf dem Server veröffentlichten Update enthalten ist, sodass die Back-End-Anwendung die Quelle der Aktualisierung leichter identifizieren kann.
- Mit der Eigenschaft *Status* wird das numerische Tag angegeben, in das Informationen zum Anschlussstatus geschrieben werden. Weitere Informationen finden Sie in der Tabelle weiter oben in diesem Kapitel.

18.6.2 Anschlussbetrieb

Im Zwillingsmodus, der Anschluss veröffentlicht JSON-Fragmente mit Geräte- und Tag-Daten im Abschnitt `reported` innerhalb des Zwillings des angegebenen Dings. Wenn im Abschnitt `desired` des Zwillings ein entsprechender Tag-Wert erstellt wird, aktualisiert der Anschluss das Tag, wenn dieses Tag in einem Datensatz enthalten ist, der Schreibvorgänge zulässt. Der Anschluss entfernt den Wert „desired“ nicht, weil das Azure Framework eine Änderung des Abschnitts `desired` durch das Gerät nicht zulässt. Dies kann die bidirektionale

Datenübertragung erschweren, weil ein ausstehender Cloud-Schreibvorgang eine lokale Änderung unerwartet außer Kraft setzen kann.

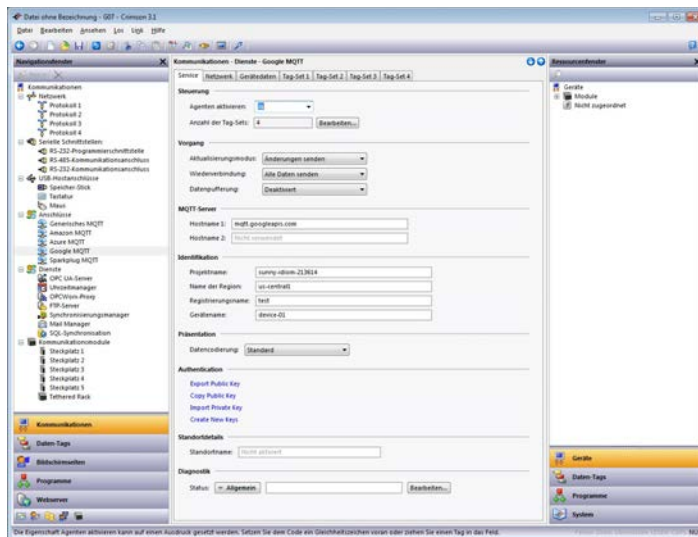
Im Telemetriemodus veröffentlicht der Konnektor JSON-Fragmente, die Geräte- und Tag-Daten enthalten, auf dem Ereignisendpunkt des Geräts. Das Fragment enthält zusätzlich zu den üblichen Daten einen Zeitstempel und den Azure-Gerätenamen des Quellgeräts. Der Konnektor abonniert den gerätegebundenen Endpunkt des Geräts und überprüft alle Nachrichten auf einen Tag-Abschnitt, der Informationen enthält, die Tags mit aktivierten Schreibvorgängen entsprechen. Ein Schreibvorgang auf ein Tag kann eine Ereignisnachricht über die Änderung erzeugen. Wenn dieses Verhalten nicht gewünscht ist, platzieren Sie Ihre beschreibbaren Tags in einem Tag-Satz, der für „Write Only“ (Nur Schreibvorgänge) konfiguriert ist.

18.7 Der Google MQTT-Konnektor

Der Google MQTT-Konnektor ist ein spezieller Konnektor, der darauf ausgelegt ist, Tag- und Gerätedaten in der Google Cloud Platform (GCP) zu veröffentlichen. Aktualisierungen werden im Ereignisthema veröffentlicht. Schreibvorgänge auf Tags, die als beschreibbar konfiguriert sind, werden vom Konfigurationsthema angenommen. Der Konnektor implementiert die Sicherheit mithilfe lokal generierter RS256-Schlüsselpaare.

18.7.1 Anschlusseinstellungen

Die Einstellungen für den Anschluss sind unten abgebildet.



- Mit der Eigenschaft *Enable Agent* (Agenten aktivieren) wird dieser Anschluss aktiviert bzw. deaktiviert.
- Mit der Eigenschaft *Tag Set Count* (Anzahl an Tag-Sätzen) kann die Anzahl der vom Konnektor unterstützten Tag-Sätze geändert werden. Es können bis zu 30 Tag-Sätze erstellt werden. Änderungen an diesem Wert können nicht rückgängig gemacht werden. Der Puffer für Rückgängigmachungen wird geleert.
- Mit der Eigenschaft *Update Mode* (Aktualisierungsmodus) wird angegeben, welche Datenwerte in den jeweiligen Aktualisierungsnachrichten gesendet werden sollen. Mit „Send Changes“ (Änderungen senden) werden nur die Werte gesendet, die sich seit dem letzten Senden von Daten um mehr als ihre Totzone geändert haben. Mit „Send All If Any“ (Alle senden, wenn vorhanden) werden alle Datenwerte aus einem Tag-Satz gesendet, sofern sich Werte in diesem Satz um mehr als die Totzone geändert haben. Mit „Send All Data“ (Alle Daten senden) werden alle Daten zu jedem Aktualisierungsereignis gesendet, unabhängig davon, ob Änderungen aufgetreten sind.
- Mit der Eigenschaft *Reconnection* (Wiederverbindung) wird festgelegt, wie sich der Konnektor verhalten soll, wenn eine Verbindung getrennt und wiederhergestellt wird. Mit der Option „Send All Data“ (Alle Daten senden) werden die Verlaufswerte zurückgesetzt, die zum Erkennen von Änderungen verwendet werden, und dadurch werden alle Daten erneut gesendet. Im „Update Mode“ (Aktualisierungsmodus) wird die Aktualisierung der Wiederverbindung wie eine normale Aktualisierung behandelt und daher das gleiche Verhalten angewendet wie von der Eigenschaft *Update Mode* definiert.
- Mit der Eigenschaft *Data Buffering* (Datenpufferung) kann die Datenpufferung aktiviert bzw. inaktiviert werden. Weitere Informationen hierzu finden Sie im entsprechenden Abschnitt weiter oben in diesem Kapitel. Beachten Sie, dass hierfür die Datenpufferung auch auf Ebene des Tag-Satzes aktiviert sein muss.
- Mit den Eigenschaften *Host Name 1* und *Host Name 2* werden die Namen der MQTT-Server angegeben, auf denen der Anschluss Daten veröffentlicht und von denen er Abonnementupdates empfängt.

Host Name 1 muss mit dem Hostnamen des IoT-Hubs übereinstimmen, der in Azure erstellt wurde. *Host Name 2* hingegen wird nur in Topologien mit Redundanz eingesetzt. Der Konnektor versucht zuerst, eine Verbindung zu *Host Name 1* herzustellen. Wenn dies nicht gelingt, versucht er es mit *Host Name 2*. Wenn keiner der Hosts reagiert, wartet der Anschluss bis zum Ende der Back-off-Zeit und beginnt dann den Zyklus von vorn.

- Mit der Eigenschaft *Project Name* (Projektname) wird der Name des GCP-Projekts angegeben. Mit der Eigenschaft *Region Name* (Regionsname) wird die Region angegeben, in der die Geräteregistrierung erstellt wurde. Mit der Eigenschaft *Registry Name* (Registry-Name) wird der Name dieser Registry angegeben. Mit der Eigenschaft *Device Name* (Gerätename) wird der Name des Geräts in der Registry angegeben.
- Mit der Eigenschaft *Data Encoding* (Datencodierung) wird gesteuert, wie boolesche Werte in der JSON dargestellt werden, die an den Server gesendet wird. Das Verhalten wird weiter oben in diesem Kapitel beschrieben.
- Der Abschnitt *Authentication* (Authentifizierung) enthält Vorgänge zur Verwaltung des kryptografischen Schlüsselpaars, das zur Identifizierung eines Geräts in der GCP verwendet wird. Wenn der Konnektor zum ersten Mal aktiviert wird, wird ein öffentlicher und ein privater Schlüssel erstellt. Der öffentliche Schlüssel kann in eine Datei exportiert oder in der Zwischenablage abgelegt werden. Er muss bei GCP mit einem RS256-Typ angegeben werden, wenn das entsprechende Gerät erstellt wird. Wenn Sie eine Verbindung zu einem Gerät herstellen, das über ein vorhandenes Schlüsselpaar verfügt, können Sie den privaten Schlüssel importieren und die Werte von Crimson überschreiben. Es steht auch eine Option zur Verfügung, um ein neues Schlüsselpaar zu generieren. Damit der Konnektor Daten erfolgreich weitergeben kann, muss der neue öffentliche Schlüssel bei GCP angegeben werden.
- Mit der Eigenschaft *Site Name* (Site-Name) wird eine Zeichenfolge angegeben, die in jedem auf dem Server veröffentlichten Update enthalten ist, sodass die Backend-Anwendung die Quelle der Aktualisierung leichter identifizieren kann.
- Mit der Eigenschaft *Status* wird das numerische Tag angegeben, in das Informationen zum Konnektorstatus geschrieben werden. Weitere Informationen finden Sie in der Tabelle weiter oben in diesem Kapitel.

18.7.2 Bedienung des Steckverbinders

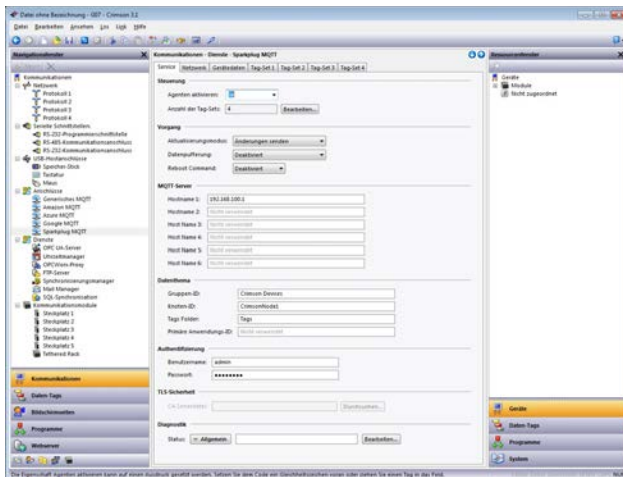
Der Konnektor veröffentlicht JSON-Fragmente, die Geräte- und Tag-Daten enthalten, im Ereignisthema und überwacht eingehende Nachrichten im *Konfigurationsthema*. Alle Tag-Daten in den eingehenden Nachrichten werden an die entsprechenden Tags weitergeleitet, vorausgesetzt, diese sind für die Annahme von Schreibvorgängen konfiguriert.

18.8 Der Sparkplug MQTT-Konnektor

Der Sparkplug MQTT-Konnektor ist ein spezieller Konnektor, der darauf ausgelegt ist, Tag- und Gerätedaten im Ignition SCADA-System von Inductive Automation zu veröffentlichen. Im Gegensatz zu der von den meisten anderen Anschlüssen verwendeten textlichen JSON-Darstellung verwendet Sparkplug eine binäre Darstellung, die etwas effizienter ist. Dies bedeutet auch, dass die meisten Optionen für Tag-Daten hinsichtlich der JSON-Formatierung nicht für diesen Anschluss verfügbar sind.

18.8.1 Anschlusseinstellungen

Die Einstellungen für den Anschluss sind unten abgebildet.



- Mit der Eigenschaft *Enable Agent* (Agent aktivieren) wird dieser Konnektor aktiviert bzw. deaktiviert.
- Mit der Eigenschaft *Tag Set Count* (Anzahl an Tag-Sätzen) kann die Anzahl der vom Konnektor unterstützten Tag-Sätze geändert werden. Es können bis zu 30 Tag-Sätze erstellt werden. Änderungen an diesem Wert können nicht rückgängig gemacht werden. Der Puffer für Rückgängigmachungen wird geleert.
- Mit der Eigenschaft *Update Mode* (Aktualisierungsmodus) wird angegeben, welche Datenwerte in den jeweiligen Aktualisierungsnachrichten gesendet werden sollen. Mit „Send Changes“ (Änderungen senden) werden nur die Werte gesendet, die sich seit dem letzten Senden von Daten um mehr als ihre Totzone geändert haben. Mit „Send All If Any“ (Alle senden, wenn vorhanden) werden alle Datenwerte aus einem Tag-Satz gesendet, sofern sich Werte in diesem Satz um mehr als die Totzone geändert haben. Mit „Send All Data“ (Alle Daten senden) werden alle Daten zu jedem Aktualisierungsereignis gesendet, unabhängig davon, ob Änderungen aufgetreten sind.
- Mit der Eigenschaft *Data Buffering* (Datenpufferung) kann die Datenpufferung aktiviert bzw. inaktiviert werden. Weitere Informationen hierzu finden Sie im entsprechenden Abschnitt weiter oben in diesem Kapitel. Beachten Sie, dass hierfür die Datenpufferung auch auf Ebene des Tag-Satzes aktiviert sein muss.
- Mit den Eigenschaften *Host Name 1* und *Host Name 2* werden die Namen der MQTT-Server angegeben, auf denen der Konnektor Daten veröffentlicht und von denen er Abonnementupdates empfängt. Sie können als IP-Adresse oder als tatsächlicher Hostname angegeben werden, doch Verbindungen mit TLS erfordern in der Regel einen Namen, wenn das Serverzertifikat validiert werden soll. Der Konnektor versucht zuerst, eine Verbindung mit *Host Name 1* herzustellen. Wenn dies nicht gelingt oder wenn die Eigenschaft *Primary App ID* (ID der Primäranwendung) definiert ist und der Server meldet, dass die Primäranwendung offline ist, versucht es der Konnektor mit *Host Name 2*. Wenn keiner der Hosts reagiert, wartet der Anschluss bis zum Ende der Back-off-Zeit und beginnt dann den Zyklus von vorn.

- Mit den Eigenschaften *Group ID* (Gruppen-ID) und *Node ID* (Knoten-ID) wird angegeben, wie dieses Gerät seine Daten im Tag-Browser von Ignition darstellen soll. Die Zeichenfolgen sind willkürlich, sollten aber zusammen einen eindeutigen Wert ergeben, der dieses Crimson-Gerät identifiziert.
- Mit der Eigenschaft *Primary App ID* (Primäre Anwendungs-ID) wird die Kennung der primären Anwendung definiert, die mit dem Server verbunden werden muss, bevor der Anschluss mit der Übergabe von Daten beginnt. Wenn die primäre Anwendung als offline gemeldet wird oder wenn innerhalb von 75 % der Keep-Alive-Zeit kein Bericht über einen Online-Status empfangen wird, wechselt der Anschluss zu einem alternativen Server, falls verfügbar. Wenn Sie dieses Feld leer lassen, wird diese Funktionalität deaktiviert. Wenn Sie einen Wert für dieses Feld eingeben, muss er mit dem Feld „Primary Host ID“ (Primäre Host-ID) in der Konfiguration der Ignition MQTT Engine bzw. mit dem entsprechenden Feld in der verwendeten primären Anwendung übereinstimmen. Wenn die Felder nicht übereinstimmen, übergibt der Anschluss keine Daten an den Server.
- Die Eigenschaften *User Name* (Benutzername) und *Password* (Kennwort) dienen zur Authentifizierung des Anschlusses gegenüber dem MQTT-Server. Die Standardwerte von Ignition lauten `admin` und `changeme` und sind gleichermaßen die Standardwerte für diesen Anschluss.
- Die Eigenschaft *Server CA File* (ZS-Serverdatei) ist für die künftige Nutzung reserviert.
- Mit der Eigenschaft *Status* wird ein numerisches Tag angegeben, in das Statusinformationen geschrieben werden. Weitere Informationen finden Sie in der Tabelle weiter oben in diesem Kapitel.

18.8.2 Anschlussbetrieb

Der Anschluss veröffentlicht Geräte- und Tag-Daten an Ignition und reagiert auf Tag-Schreibvorgänge, sofern entsprechend konfiguriert. Tags innerhalb der Ordner werden in den entsprechenden Ordner im Tag-Browser von Ignition platziert und Array-Elemente werden in Ordnern platziert, die dem zugrunde liegenden Tag entsprechen. Wenn die primäre Anwendungs-ID definiert ist, gewährleistet der Anschluss, dass die entsprechende Anwendung mit dem Server verbunden ist und als online gemeldet wird, bevor Daten übertragen werden. Wenn die Anwendung nicht verbunden ist, versucht der Server, einen alternativen Host zu kontaktieren, sofern definiert. Wenn die Anwendung auf keinem der Hosts gefunden werden kann, wartet der Anschluss bis zum Ende der Back-off-Zeit und beginnt den Vorgang von vorn, bis die primäre Anwendung gefunden wird.

Kapitel 19 Verwenden von Diensten

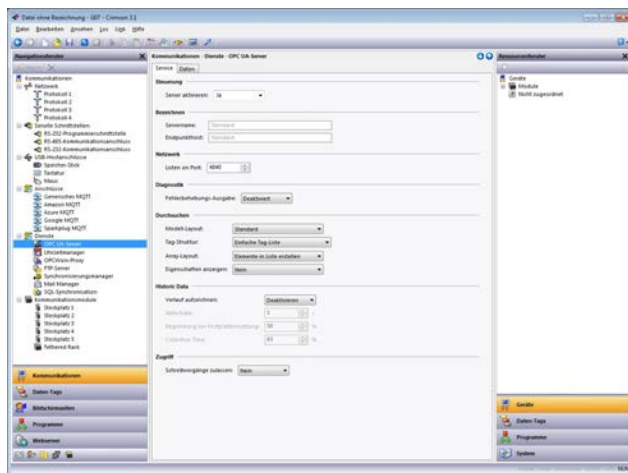
Zusätzlich zu den in diesem Dokument bereits beschriebenen Kernfunktionen bietet die Kategorie „Communications“ (Kommunikationen) auch die Möglichkeit, verschiedene Dienste zu konfigurieren. Diese Dienste werden im Navigationsfenster unter dem Symbol „Dienste“ angezeigt und im Folgenden einzeln beschrieben.

19.1 Verwenden des OPC UA-Servers

Crimson® 3.1 enthält einen OPC UA-Server, der es ermöglicht, die von Ihrer Datenbank definierten Tags mithilfe des OPC UA-Protokolls zugänglich zu machen. Der Server unterstützt sowohl Live-Daten als auch die Pufferung von historischen Daten, damit Daten nach einem Verbindungsverlust wiederhergestellt werden können. Der Server unterstützt Abonnements und Veröffentlichungen und kann somit Updates an Clients bereitstellen, ohne dass kontinuierlich Datenwerte gelesen werden müssten. Beachten Sie, dass es bei der Konfiguration des OPC UA-Servers wichtig ist, einen Mechanismus für die Zeitsynchronisierung bereitzustellen, da das Protokoll von genauen Zeitstempeln abhängig ist. Weitere Informationen zum Konfigurieren dieser Funktionalität finden Sie im Abschnitt „Zeitmanagement“ dieses Kapitels.

19.1.1 Konfigurieren des Dienstes

Der OPC UA-Server wird über das zugehörige Symbol im Navigationsfenster konfiguriert:



19.1.1.1 Diensteigenschaften

Die folgenden Eigenschaften können über die Registerkarte „Service“ (Dienst) konfiguriert werden:

- Mit der Einstellung *Enable Server* (Server aktivieren) wird der OPC UA-Server aktiviert bzw. deaktiviert.
- Mit der Eigenschaft *Server Name* (Servername) wird der Servername definiert, der während des Verbindungsprozesses an den OPC UA-Client zurückgegeben wird. Der Name muss in der Regel für jedes OPC UA-Gerät, das mit einem bestimmten Netzwerk verbunden ist, eindeutig sein. Wenn Sie den Namen leer lassen, verwendet der Server den Namen des Endpunkthosts.
- Mit der Eigenschaft *Endpoint Host* (Endpunkthost) wird der Host-Teil des Endpunktnamens definiert, der an den OPC UA-Client zurückgegeben wird. Dieser entspricht in der Regel dem DNS-Namen, der

zur Identifizierung des Geräts verwendet wird. Der vollständige Endpunktname ist dieser Name mit dem Präfix `opc.tcp://`, gefolgt von einem Doppelpunkt und der TCP-Portnummer. Wenn Sie den Namen leer lassen, verwendet der Server den Namen des Geräts, der auf der Registerkarte „Zero Config“ definiert wurde. Hier wird standardmäßig die Form `red-xx-yy-zz.local` verwendet, die bereits an anderer Stelle in diesem Handbuch erörtert wurde.

- Mit der Eigenschaft *Listen on Port* (*Port überwachen*) wird der TCP-Port definiert, den der Server überwachen soll. Der Standardwert lautet 4840 und entspricht dem Standardwert, der von den meisten OPC UA-Clients verwendet wird.
- Die Eigenschaft *Debug Output* (*Debug-Ausgabe*) wird verwendet, um die Ausgabe erweiterter Diagnoseinformationen an die verschiedenen Debug-Konsolen des Systems zu aktivieren bzw. zu deaktivieren. Diese Informationen sind technischer Art und nicht vollständig dokumentiert. Sie sind für den technischen Support von Red Lion oder für erfahrenere Benutzer bestimmt. Informationen zum Konfigurieren von Debug-Konsolen finden Sie im Kapitel *Erweitertes Debuggen*.
- Mit der Eigenschaft *Model Layout* (*Modell-Layout*) wird entweder das Datenmodell-Standardlayout ausgewählt oder aber ein vereinfachtes Layout, in dem die Tags direkt im Ordner „Objects“ (Objekte) des Servers platziert werden. Das vereinfachte Layout führt zwar zu kürzeren Pfadnamen für Datenelemente, doch wird ein Mechanismus entfernt, der speziell für die Bereitstellung der Abwärtskompatibilität bei Softwareupdates vorgesehen ist. Es wird empfohlen, das Standardmodell zu verwenden, es sei denn, es gibt triftige Gründe, die dagegen sprechen.
- Mit der Eigenschaft *Tag Structure* (*Tag-Struktur*) wird definiert, wie der Server Daten-Tags innerhalb des OPC UA-Modells präsentiert. Bei der Einstellung „Simple Tag List“ (Einfache Tag-Liste) werden alle ausgewählten Tags unter einem einzelnen Ordner aufgelistet, wobei die in Crimson erstellte Ordnerstruktur anhand mit Punkten versehenen Namen dargestellt wird. Bei der Einstellung „Folders with Short Names“ (Ordner mit Kurznamen) wird die in Crimson erstellte Ordnerstruktur dupliziert, und die Kurznamen der Tags werden zur Benennung der Tag-Knoten verwendet. Bei der Einstellung „Folders with Full Names“ (Ordner mit vollständigen Namen) wird ebenfalls die Ordnerstruktur dupliziert, doch die einzelnen Tag-Knoten werden mit dem vollständigen mit Punkten versehenen Namen benannt.
- Mit der Eigenschaft *Array Layout* (*Array-Layout*) wird definiert, wie Crimson Array-Elemente im Datenmodell präsentiert. Bei der Einstellung „Create Elements in List“ (Elemente in Liste erstellen) wird für jedes Array-Element ein Element namens `Tag[x]` erstellt und diese Elemente werden direkt in dem Ordner platziert, der das Tag enthält. Bei der Einstellung „Create Elements in Folder“ (Elemente im Ordner erstellen) wird ein Ordner namens `Tag` am gleichen Speicherort erstellt und in diesem Ordner werden Elemente mit numerischen Namen erstellt.
- Mit der Eigenschaft *Show Properties* (*Eigenschaften anzeigen*) wird angegeben, ob der Server die verschiedenen Tag-Eigenschaften zugänglich machen soll, die im Kapitel „Schreiben von Ausdrücken“ beschrieben sind, um den Zugriff auf zusätzliche Informationen zu ermöglichen. Wenn Sie „StateText“ aktivieren, verfügen alle Tags, die Mehrfachstatus-Objekte verwenden, auch über eine Eigenschaft, die die Anzahl der Status definiert, die sie darstellen, sowie eine weitere Eigenschaft, die ein Array von Zeichenfolgen darstellt, mit dem der numerische Wert des Tags in Zeichenfolgen konvertiert werden kann.
- Mit der Eigenschaft *Record History* (*Datensatzverlauf*) wird angegeben, ob historische Daten auf der Speicherkarte aufgezeichnet und über das OPC UA-Protokoll zur Verfügung gestellt werden sollen. Die Daten werden mit der in der Eigenschaft *Sample Rate* (*Abtastrate*) angegebenen Rate aufgezeichnet, wobei der Anteil der Speicherkarte, der durch das *Festplattennutzungslimit* festgelegt ist, genutzt wird. Weitere Informationen hierzu finden Sie im Abschnitt zu historischen Daten weiter unten.
- Mit der Eigenschaft *Allow Writes* (*Schreibvorgänge zulassen*) werden Schreibvorgänge in Tags aktiviert bzw. deaktiviert.

19.1.1.2 Dateneigenschaften

Auf der Registerkarte „Data“ (Daten) werden die Tags definiert, die vom Server verfügbar gemacht werden sollen. Die Tags in Ordnern werden nach ihren vollständig mit Punkten versehenen Namen aufgeführt. Diese flache Liste

kann je nach Einstellung der Eigenschaft „Tag Structure“ (Tag-Struktur) von Crimson wieder in eine Ordnerhierarchie konvertiert werden.

19.1.2 Darstellung von Daten

In der Standardkonfiguration macht der OPC UA-Server das konfigurierte Tag auf zweierlei Art verfügbar:

- Der Ordner `Arrays` enthält drei Arrays, die jeweils Elemente enthalten. Die Anzahl dieser Elemente entspricht dabei der Anzahl der Tags, die vom Gerät verfügbar gemacht werden. Das Array `Integer` enthält die Ganzzahl-Darstellung von numerischen Tags sowie Nullen für String-Tags. Das Array `Double` enthält die Gleitkomma-Darstellung von numerischen Tags sowie Nullen für String-Tags. Das Array `String` enthält die String-Darstellung jedes Tags, wobei jedes Tag mit dem geeigneten Formatobjekt konvertiert wurde.
- Der Ordner `Tags` enthält einen Eintrag für jedes verfügbar gemachte Tag. Die Einträge sind optional in Ordnern angeordnet, die mit der Tag-Struktur der Datenbank übereinstimmen, und jedes Tag enthält optional zusätzliche Eigenschaften in Bezug auf dieses Tag.

Beide Ordner befinden sich im Ordner `v1` unter dem Ordner `Objects` im OPC UA-Datenmodell des Servers. Der Ordner `v1` enthält einen Versionierungsmechanismus, damit vorhandene Bereitstellungen nicht durch später bereitgestellte Versionen untauglich gemacht werden. Wenn Red Lion das Datenmodell in einer Weise ändert, die nicht mit vorhandenen Versionen kompatibel ist, wird das alte Modell unter dem Ordner `v1` beibehalten, und das neue Modell wird unter einem Ordner namens `v2` eingefügt. Alle zukünftig unter `v1` durchgeführten Änderungen sind daher garantiert mit vorhandenen Bereitstellungen kompatibel.

Das Standardmodell kann wie oben beschrieben mit der Eigenschaft `Model Layout` (Modell-Layout) außer Kraft gesetzt werden.

19.1.3 Historische Daten

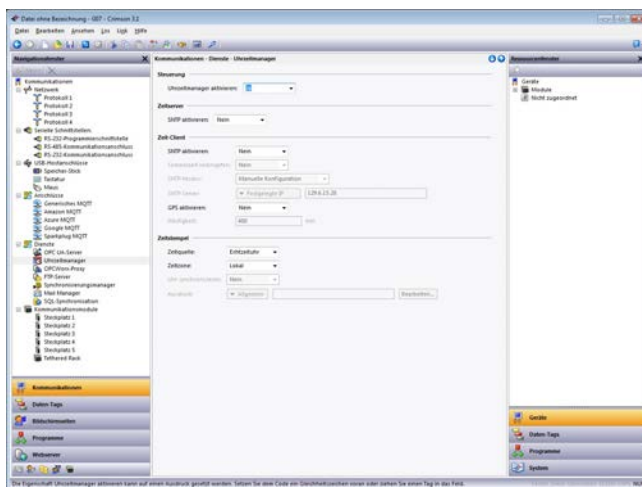
Der OPC UA-Server kann für die Speicherung historischer Daten konfiguriert werden, sodass Messwerte bei der Wiederherstellung einer unterbrochenen Verbindung zum Server wiederhergestellt werden können. Obwohl Daten aus beliebigen Punkten im Verlaufspuffer extrahiert werden können, ist die Leistung begrenzt. Die Funktion unterstützt nicht das zufällige Durchsuchen von Daten. Vielmehr wird erwartet, dass ein Client wie OSIsoft PI Historian mit dieser Funktion alle fehlenden Datenelemente liest und dann zur Annahme veröffentlichter Daten zurückkehrt. Außerdem sollte beachtet werden, dass die Funktion für historische Daten eine erhebliche Speicherkarten-Bandbreite genutzt hat, insbesondere bei hohen Abtastraten mit hohen Tag-Anzahlen. Sie wurde zwar mit 500 Tags und einer Abtastrate von einer Sekunde getestet, jedoch wird nicht empfohlen, diese Zahlen zu überschreiten oder andere speicherkartenintensive Vorgänge wie das MQTT-Teilstreckenverfahren (Store-and-Forward) gleichzeitig zu verwenden. Auf der Website von Red Lion finden Sie technische Hinweise zur Integration mit verschiedenen Historian-Clients über diese Funktion.

19.2 Verwenden von Zeitmanagement

Crimson enthält Funktionen, mit denen Sie die Uhrzeit und das Datum des Zielgeräts mit einer Vielzahl von Quellen synchronisieren können. Der Time Manager (Zeitmanager) kann auch Informationen über die Zeitzone des Geräts und über die Aktivierung der Sommerzeit verwalten. Genaue Zeitoneninformationen sind für die korrekte Synchronisierung wichtig, da die verschiedenen Synchronisierungsmethoden alle für die Verwendung der „Universal Coordinated Time“ entwickelt wurden, die auch als UTC oder Greenwich Mean Time bezeichnet wird. Eine genaue UTC-Zeit kann auch erforderlich sein, wenn Sie TLS oder SSL verwenden, um eine Verbindung zu einem externen Gerät herzustellen, da bei der Validierung des Serverzertifikats optional das Ablaufdatum des Zertifikats geprüft werden kann. Crimson kann sowohl als Client als auch als Server fungieren und entweder die Zeit von anderen Crimson-Geräten anfordern oder diesen bereitstellen. Beachten Sie, dass die Server-Implementierung derzeit keine Clients von Dritten unterstützt.

19.2.1 Konfigurieren des Dienstes

Der Time Manager (Zeitmanager) wird über das zugehörige Symbol im Navigationsfenster konfiguriert:



Über die Eigenschaft *Enable Time Manager* (Zeitmanager aktivieren) wird der Zugriff auf die anderen Funktionen gesteuert. Wenn diese Eigenschaft nicht aktiviert ist, arbeitet Crimson nur in der lokalen Zeitzone. Andere Zeitonen sind dann unbekannt und es liegen auch keine weiteren Informationen zum Zeitmanagement vor.

19.2.1.1 Zeitserver

Durch die entsprechende Konfiguration der Eigenschaft *Enable SNTP* (SNTP aktivieren) unter „Time Server“ (Zeitserver) wird Crimson angewiesen, als SNTP-Server zu fungieren. Andere Crimson-Geräte können dann ihre eigenen Uhren mit der Uhr dieses Geräts synchronisieren. Beachten Sie, dass die SNTP-Implementierung von Crimson nicht ganz RFC-konform ist und nicht als Synchronisationsquelle für Clients von Drittanbietern unterstützt wird.

19.2.1.2 Time Client

Wenn Sie im Abschnitt „Time Client“ für die Eigenschaft *Enable SNTP* (SNTP aktivieren) die Einstellung „Yes“ (Ja) wählen, synchronisiert Crimson seine Uhr mit einem anderen Crimson-basierten Gerät oder mit einer anderen SNTP-basierten Zeitquelle, die über das lokale Netzwerk oder das Internet zugänglich ist. Unter „Time Client“ sind außerdem die folgenden Eigenschaften verfügbar:

- Mit der Eigenschaft *Linked DST* (Verknüpfte SZ) wird der SNTP-Client angewiesen, zu versuchen, die aktuelle Einstellung der Sommerzeit auf dem SNTP-Server zu lesen. Da diese Funktion kein

Bestandteil des SNTP-Standardprotokolls ist, funktioniert sie nur, wenn ein anderes Crimson-Gerät als Server angegeben wird. Diese Funktion ist nützlich, weil die Sommerzeit auf einem einzelnen Gerät im Werksnetz eingestellt werden kann und die anderen Geräte dann der zentralen Einstellung folgen.

- Die Eigenschaften *SNTP Mode* (SNTP-Modus) und *SNTP Server* (SNTP-Server) werden zum Konfigurieren der IP-Adresse des SNTP-Servers (Simple Network Time Service) verwendet. Wenn „Configured via DHCP“ (Konfiguriert über DHCP) ausgewählt wird, muss mindestens ein Ethernet-Port für die Verwendung von DHCP konfiguriert werden, und der Server muss so konfiguriert werden, dass er einen Server über die Option 42 bestimmt. Die Servereinstellung für „Manual Configuration“ (Manuelle Konfiguration) ist standardmäßig ein öffentlich zugänglicher Server, der vom National Institute of Standards and Technology (NIST) in den USA verwaltet wird. Dies ist nur dann geeignet, wenn das Gerät über Internetzugang verfügt.
- Die Eigenschaft *Enable GPS* (GPS aktivieren) wird verwendet, um den Time Client anzuweisen, ein über NMEA-0183 verbundenes GPS-Gerät als alternative Methode zum Abrufen der aktuellen Uhrzeit zu verwenden. Das Gerät kann über den entsprechenden Treiber an einen beliebigen seriellen Anschluss angeschlossen sein.
- Die Eigenschaft *Frequency* (Häufigkeit) gibt an, wie oft Crimson versuchen soll, seine Uhrzeit mit den oben aktivierten Methoden zu synchronisieren. Crimson wird immer versuchen, zwanzig Sekunden nach dem Einschalten zu synchronisieren. Danach erfolgt die Synchronisierung so oft, wie von dieser Eigenschaft angegeben. Wenn ein bestimmter Synchronisierungsversuch fehlschlägt, macht das Gerät alle 30 Sekunden einen weiteren Versuch, bis es eine geeignete Zeitquelle gefunden hat.

19.2.1.3 Zeitstempel

Crimson kann eine Vielzahl von Protokolldateien auf der Speicherkarte des Zielgeräts aufzeichnen. Dabei bekommt jeder Protokolleintrag einen Zeitstempel. Standardmäßig wird der Zeitstempel von der lokalen Echtzeituhr übernommen und befindet sich in der lokalen Zeitzone. Das Verhalten kann über die folgenden Eigenschaften geändert werden:

- Mit der Eigenschaft *Time Source* (Zeitquelle) wird angegeben, wo die Zeitstempel abgerufen werden sollen. Bei der Standardeinstellung wird die Zeit von der Echtzeituhr des Geräts selbst abgerufen. Bei der alternativen Einstellung kann die aktuelle Uhrzeit mit einem Ausdruck definiert werden. Dieser Ausdruck ist in der Regel ein Verweis auf ein Datenelement in einem angeschlossenen Gerät und ermöglicht es, die Uhr dieses Geräts für die Datenprotokollierung zu verwenden. Der Ausdruck muss unter der Eigenschaft *Expression* (Ausdruck) eingegeben werden.
- Mit der Eigenschaft *Sync Clock* (Uhr synchronisieren) wird angegeben, ob die lokale Echtzeituhr mit der oben angegebenen alternativen Zeitquelle synchronisiert werden soll. Wenn diese Option aktiviert ist, wird die lokale Uhr beim Start und danach in regelmäßigen Abständen synchronisiert. Die lokale Uhr wird als Zeitstempelquelle verwendet, wenn die alternative Quelle aufgrund von Kommunikationsproblemen nicht verfügbar ist.
- Mit der Eigenschaft *Time Zone* (Zeitzone) wird die Zeitzone angegeben, die für Zeitstempel verwendet werden soll. Sie ist nur anwendbar, wenn die lokale Echtzeituhr als Quelle für Zeitstempel konfiguriert ist. Bei Auswahl von „Local“ (Lokal) wird die lokale Zeitzone verwendet und bei Auswahl von „UTC“ die „Universal Coordinated Time“. Die letztgenannte Einstellung erzeugt Protokolldateien, die sich leichter zwischen Zeitzonen übertragen lassen und keine Unstetigkeiten wegen des Wechsels zwischen Sommer- und Winterzeit aufweisen.

19.2.2 Auswählen eines SNTP-Servers

Beim Konfigurieren des SNTP-Clients haben Sie bei der Auswahl eines Servers verschiedene Möglichkeiten.

Wenn Sie einen Windows- oder Unix-basierten Zeitserver als Bestandteil ihrer Netzwerkinfrastruktur verwenden, sollten Sie letztendlich mit dieser Quelle synchronisieren, um eine unternehmensweite Synchronisierung zu gewährleisten. Wenn Sie jedoch mehrere Crimson-Geräte in dem selben Netzwerk verwenden, ist es besser, eines der Geräte zwecks Einstellung der Sommerzeit als Master-Gerät festzulegen und dann nur dieses Gerät mit der Unternehmens-Zeitquelle zu synchronisieren. Anschließend können Sie die anderen Geräte so konfigurieren, dass sie sich mit dem Master-Gerät synchronisieren, und die Funktion „Linked DST“ (Verknüpfte SZ) aktivieren, um die Sommerzeit-Einstellung auf das ganze Werk zu übertragen.

Wenn Sie keine Unternehmens-Zeitquelle haben, können Sie ein einzelnes Crimson-Gerät als den Punkt festlegen, an dem ein Bediener die Uhrzeit einstellt, und dann veranlassen, dass andere Geräte sich mit dieser Quelle synchronisieren. Wenn Ihre Installation den TCP/IP-Zugang zum Internet über eine Ethernet- oder Modemverbindung bereitstellt, können Sie den SNTP-Client so konfigurieren, dass er mit einem öffentlichen Zeitserver synchronisiert wird. Ein Beispiel hierfür wäre 192.6.15.28; dies ist die aktuelle IP-Adresse eines öffentlichen Zeitservers, der von NIST bereitgestellt wird.

Eine Liste weiterer Server finden Sie unter:

<http://support.microsoft.com/kb/262680>

Wenn auf Ihrem Gerät DNS aktiviert ist, sollten Sie idealerweise mit einem DNS-Namen auf den SNTP-Server verweisen. Für die oben erwähnten NIST-Server wird mit dem Namen `time.nist.gov` nach dem Zufallsprinzip einer der aktiven Server im Serverpool ausgewählt. Dieser Mechanismus bietet einen besseren Lastenausgleich und eine bessere Redundanz und ist gleichzeitig immun gegen Änderungen an IP-Adressen.

19.2.3 Konfigurieren von Zeitzonen

Wie bereits erwähnt, muss ein Crimson-Gerät die aktuelle Zeitzone kennen, wenn es das erweiterte Zeitmanagement verwenden soll. Diese Informationen können auf zwei Arten bereitgestellt werden. Die einfachste Methode ist die Verwendung des Befehls „Send Time“ (Zeit senden) im Menü „Link“ der Crimson-Konfigurationssoftware. Dieser Befehl stellt nicht nur die Uhr ein, sondern sendet auch die aktuelle Zeitzone des PCs und den Status der Sommerzeit. Crimson speichert diese Daten im nichtflüchtigen Speicher und verwendet sie von diesem Zeitpunkt an. Wichtig ist hierbei, dass der PC gültige Uhrzeit- und Datumsinformationen enthält, bevor diese Angaben an das Gerät gesendet werden.

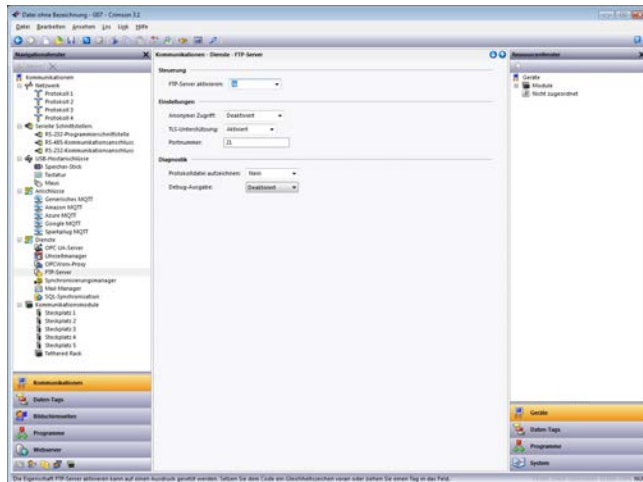
Die alternative Methode besteht darin, die Systemvariablen `TimeZone` und `UseDST` zu verwenden. Die erste Variable enthält die Anzahl der Stunden, um die die lokale Zeitzone von der UTC abweicht, und kann negativ oder positiv sein. Die Einstellung `-5` entspricht z. B. der Eastern Standard Time in den USA. Die zweite Variable enthält entweder 0 oder 1, je nachdem, ob die Sommerzeit aktiv ist. Die Bearbeitung einer dieser Variablen über die Benutzeroberfläche führt dazu, dass die Uhr des Geräts geändert wird, um die neuen Einstellungen zu berücksichtigen. Wenn Sie beispielsweise die Sommerzeit aktivieren, wird die Uhrzeit um eine Stunde nach vorn verschoben, und wenn Sie sie deaktivieren, um eine Stunde zurück. Eine typische Datenbank muss lediglich `UseDST` für die Bearbeitung durch den Benutzer zugänglich machen, und selbst dies ist möglicherweise nicht notwendig, wenn das oben beschriebene Feature „Linked DST“ (Verknüpfte SZ) verwendet wird.

19.3 Verwenden des FTP-Servers

Der FTP-Server von Crimson bietet eine Methode zum Austausch von Dateien zwischen einem Crimson-Gerät und einem Remotecomputer, auf dem eine FTP-Client-Anwendung ausgeführt wird. Das Crimson-Gerät fungiert als Server und wartet darauf, dass Clientanwendungen eine Verbindung herstellen und Daten herunter- oder hochladen.

19.3.1 Konfigurieren des Dienstes

Der FTP-Server wird über das zugehörige Symbol im Navigationsfenster konfiguriert:



Folgende Eigenschaften können konfiguriert werden:

- Die Eigenschaft *Anonymous Access* (Anonymer Zugriff) definiert die Rechte, falls vorhanden, die einem Benutzer gewährt werden, der über anonymes FTP auf den Server zugreift. Die Einstellung „Disabled“ (Deaktiviert) verhindert den anonymen Zugriff. Bei der Einstellung „Read-Only“ (Schreibgeschützt) kann der Benutzer zwar Dateien von der Speicherkarte herunter- aber nicht hochladen. Bei der Einstellung „Read-Write“ (Lese-/Schreibzugriff) sind sowohl Uploads als auch Downloads zulässig.
- Aktivieren Sie *Record Log File* (Protokolldatei aufzeichnen), um im Stammverzeichnis der Speicherkarte ein Protokoll aller FTP-Interaktionen zu führen. Diese Datei kann beim Debuggen von FTP-Vorgängen nützlich sein, beeinträchtigt die Leistung jedoch geringfügig.
- Die Eigenschaft *Debug Output* (Debug-Ausgabe) wird verwendet, um die Ausgabe erweiterter Diagnoseinformationen an die verschiedenen Debug-Konsolen des Systems zu aktivieren bzw. zu deaktivieren. Diese Informationen sind technischer Art und nicht vollständig dokumentiert. Sie sind für den technischen Support von Red Lion oder für erfahrenere Benutzer bestimmt. Informationen zum Konfigurieren von Debug-Konsolen finden Sie im Kapitel *Erweitertes Debuggen*.

19.3.2 FTP-Sicherheit

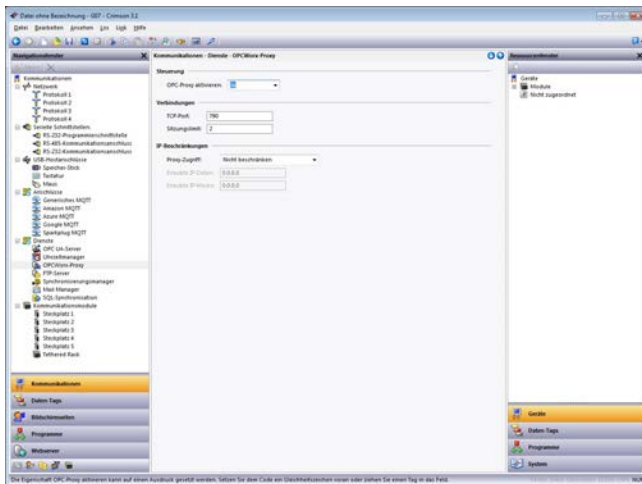
Da der FTP-Server vollen Zugriff auf die Speicherkarte bieten kann, wird dringend empfohlen, mit dem Sicherheitsmanager bestimmte Benutzername/Kennwort-Kombinationen festzulegen und diesen Benutzern entsprechende Zugriffsrechte zu erteilen. Generell sollten Sie es vermeiden, anonymen Zugriff zu gewähren. Insbesondere sollten Sie vermeiden, anonyme Schreibzugriffe zuzulassen.

19.4 Verwenden von OPCWorx Proxy

OPCWorx Proxy ist ein Service, der der OPCWorx-Software von Red Lion Zugriff auf die Tag-Datenbank des Geräts gibt. Er implementiert OPC-artige Kommunikation nicht von sich aus, sondern bietet vielmehr einen Mechanismus, mit dem OPCWorx einen PC-basierten OPC-Server bereitstellt, der Tags von einem oder mehreren Crimson-Geräten zugänglich machen kann. Darüber hinaus kann der OPC Proxy eine Verbindung mit dem Red Lion OPC-Treiber annehmen. Dieser Treiber ermöglicht einem Gerät den Zugriff auf die Tags in einem anderen Gerät, ohne dass Modbus oder ein anderes registerbasiertes Protokoll eingerichtet werden muss.

19.4.1 Konfigurieren des Dienstes

Der OPCWorx Proxy-Server wird über das zugehörige Symbol im Navigationsfenster konfiguriert:



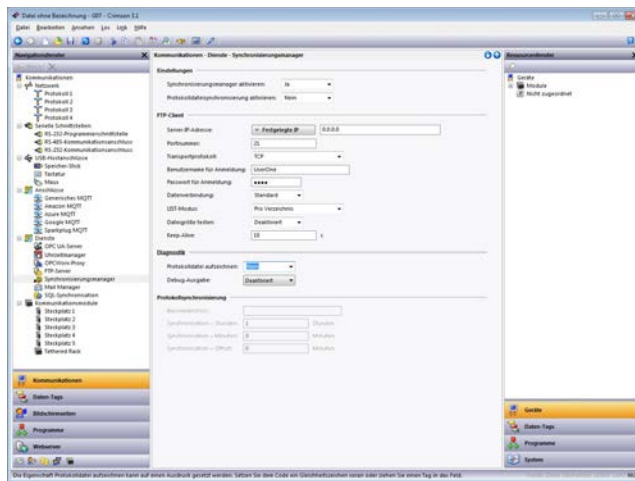
- Die Einstellung *Enable OPC Proxy* (OPC Proxy aktivieren) aktiviert bzw. deaktiviert den Dienst.
- Die Option *TCP Port* (TCP-Port) legt fest, welchen Port der Dienst überwachen soll. OPCWorx und der Red Lion OPC-Treiber erwarten standardmäßig eine Verbindung mit Port 790 und der Standardwert müsste für die meisten Anwendungen geeignet sein.
- Die Eigenschaft *Session Limit* (Sitzungslimit) steuert, wie viele Sitzungen gleichzeitig unterstützt werden. Die Anzahl sollte mindestens eins mehr als die maximale Anzahl von Geräten betragen (entweder PCs, die OPCWorx ausführen, oder andere Crimson-Geräte, die den OPC-Treiber ausführen), die eine Verbindung zu diesem Gerät herstellen.
- Mit der Eigenschaftsgruppe „IP Restrictions“ (IP-Beschränkungen) kann der Zugriff auf den Server auf bestimmte IP-Adressen beschränkt werden. Dies gilt entweder für den gesamten Zugriff oder für Schreibvorgänge in Tags. Mit der Eigenschaftsgruppe können Sie sicherstellen, dass nur autorisierte Geräte Zugriff auf die Tag-Datenbank haben.

19.5 Verwenden der Dateisynchronisierung

Mit dem Synchronisierungs-Manager können Dateien zwischen einem Crimson-Gerät und einem FTP-Server ausgetauscht werden. Diese Funktion kann zum Synchronisieren von Protokolldateien mit einem Servercomputer eingesetzt werden. Dabei kann dieser Vorgang automatisch oder auf Anforderung erfolgen. Die Funktion bietet somit eine Alternative zum Zugriff auf die Protokolldatei über den Webserver und ermöglicht die unbeaufsichtigte Übertragung von Dateien von mehreren Geräten an einen zentralen Sammelpunkt. Beachten Sie, dass der Dienst aus historischen Gründen als Synchronisierungs-Manager bezeichnet wird; es handelt sich jedoch um einen universellen FTP-Client, der auch für andere FTP-Vorgänge verwendet werden kann. Informationen zur manuellen Initiierung von Dateiübertragungen finden Sie in den Ausführungen zu den Funktionen `FtpGetFile()` und `FtpPutFile()` im Referenzhandbuch.

19.5.1 Konfigurieren des Dienstes

Der Synchronisierungs-Manager wird über das zugehörige Symbol im Navigationsfenster konfiguriert:



19.5.1.1 FTP-Client

Die folgenden Eigenschaften beziehen sich auf den FTP-Client:

- Mit der Eigenschaft *Enable Sync Manager* (Sync Manager aktivieren) wird der FTP-Client aktiviert. Der Client kann aktiviert werden, ohne die Synchronisierung zu aktivieren, und kann somit für die manuelle Dateiübertragung über die oben genannten FTP-Funktionen eingesetzt werden.
- Die Eigenschaft *Enable Log File Sync* (Synchronisierung der Protokolldatei aktivieren) dient zum Aktivieren der automatischen Protokollsynchronisierung. Im nächsten Abschnitt werden die übrigen Einstellungen für diese Funktion näher erörtert und Sie erfahren, wie sich diese Einstellungen auf den Dateispeicherort und den Zeitpunkt der Synchronisierung auswirken.
- Mit der Eigenschaft *Server IP address* (Server IP-Adresse) wird die IP-Adresse des Servers angegeben. Wie bei vielen Adressfeldern in Crimson kann es sich hier um eine feste IP-Adresse handeln; eine Zeichenfolge, die einen Hostnamen darstellt; oder ein Tag, das einen solchen Namen enthält. Sie kann auch auf eine 32-Bit-Ganzzahl gesetzt werden, die die gepackte Version der Adresse enthält.
- Mit der Eigenschaft *Port Number* (Portnummer) wird der TCP-Port angegeben, mit dem der FTP-Client eine Verbindung herzustellen versucht. Der Standardwert ist für die meisten Anwendungen geeignet, da die meisten Server Port 21 überwachen.
- Logon Username (Anmeldebenutzername) und Logon Password (Anmeldekennwort) sind die Anmeldedaten, die beim Herstellen der Verbindung an den Server gesendet werden. Bei beiden wird in der

Regel die Groß-/Kleinschreibung beachtet, obwohl dies von der jeweiligen Server-Implementierung abhängt. Für eine anonyme Anmeldung übernehmen Sie für den Benutzernamen den Standardwert. Lassen Sie das Kennwort-Feld entweder leer oder geben Sie aus Höflichkeit dem Anbieter gegenüber Ihre E-Mail-Adresse ein.

- Unter *Data Connection* (Datenverbindung) können Sie den Standardmodus oder den PASV-Modus auswählen. Sie können den PASV-Modus aktivieren, um den FTP-Client zu zwingen, alle Datenverbindungen zu initiieren, anstatt auf eingehende Verbindungen vom Server zu warten. Dieser Modus ist manchmal erforderlich, wenn Sie hinter nicht-FTP-fähigen Firewalls arbeiten oder wenn Sie bestimmte Formen der Übersetzung von Netzwerkadressen verwenden. Er wird auch beim Arbeiten über Mobilfunkmodems bevorzugt.
- Mit der *Keep Alive Time* (am-Leben-erhalten-Zeit) wird angegeben, wie lange die FTP-Verbindung aufrecht erhalten werden soll, falls weitere Übertragungen erforderlich sind. Bei einem Wert von Null wird die Verbindung geschlossen, sobald der aktuelle Transfer abgeschlossen ist. Werte von nicht Null sorgen bei der Übertragung von mehreren Dateien für einen effizienteren Betrieb.
- Mit der Eigenschaft *Record Log File* (Protokolldatei aufzeichnen) kann im Stammverzeichnis der Speicherkarte ein Protokoll aller FTP-Interaktionen geführt werden. Diese Datei kann beim Debuggen von FTP-Vorgängen nützlich sein, beeinträchtigt die Leistung jedoch geringfügig.
- Die Eigenschaft *Debug Output* (Debug-Ausgabe) wird verwendet, um die Ausgabe erweiterter Diagnoseinformationen an die verschiedenen Debug-Konsolen des Systems zu aktivieren bzw. zu deaktivieren. Diese Informationen sind technischer Art und nicht vollständig dokumentiert. Sie sind für den technischen Support von Red Lion oder für erfahrenere Benutzer bestimmt. Informationen zum Konfigurieren von Debug-Konsolen finden Sie im Kapitel *Erweitertes Debuggen*.

19.5.1.2 Synchronisierung von Protokolldateien

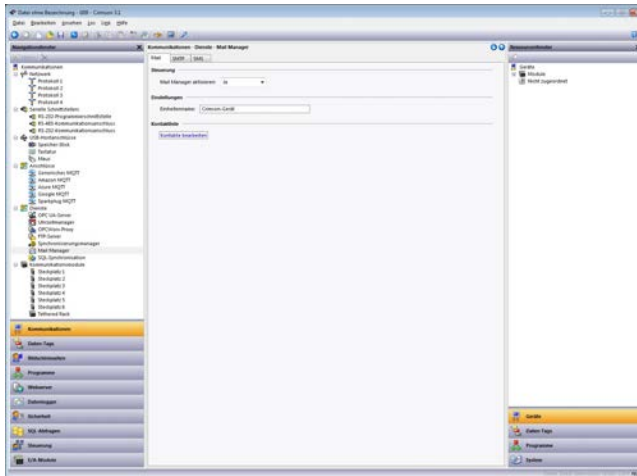
Die folgenden Eigenschaften beziehen sich speziell auf die Synchronisierung von Protokolldateien:

- Die Eigenschaft *Base Directory* (Basisverzeichnis) definiert das Verzeichnis auf dem Server, in dem die Protokolldateien platziert werden. Dieses Verzeichnis ist relativ zum Ordnerbereich des FTP-Servers und nicht relativ zu der zugrunde liegenden Verzeichnisstruktur des Server-Dateisystems. In der Regel geben Sie für jedes Crimson-Gerät, das mit einem bestimmten Server synchronisiert wird, ein anderes Basisverzeichnis an.
- Die Eigenschaft *Sync Period* (Synchronisierung – Zeitraum) gibt an, wie oft der FTP-Client eine Verbindung zum Server herstellt und seine Dateien überträgt. Sie wird in Stunden gemessen hat immer Mitternacht als Grundlage. Wenn Sie beispielsweise einen Wert von drei auswählen, finden Übertragungen um Mitternacht, um 03:00 Uhr, um 06:00 Uhr usw. statt.
- Die Eigenschaft *Sync Delay* (Synchronisierung – Verzögerung) gibt eine Abweichung in Minuten von der Standardzeit an, zu der Dateiübertragungen stattfinden. Mithilfe dieser Eigenschaft können mehrere Terminals mit einem Server kommunizieren, ohne dass alle Dateiübertragungen gleichzeitig stattfinden und dadurch die Fähigkeiten des Ziels überlasten.

19.6 Verwenden von E-Mail

Crimson kann so konfiguriert werden, dass E-Mail-Nachrichten gesendet werden, wenn Alarmbedingungen vorliegen oder wenn Benachrichtigungen über andere Ereignisse im System bereitgestellt werden müssen. Diese Nachrichten können per SMS an Mobilgeräte oder per SMTP-E-Mail an jedes Gerät gesendet werden, das E-Mail über Standardinternetprotokolle annimmt.

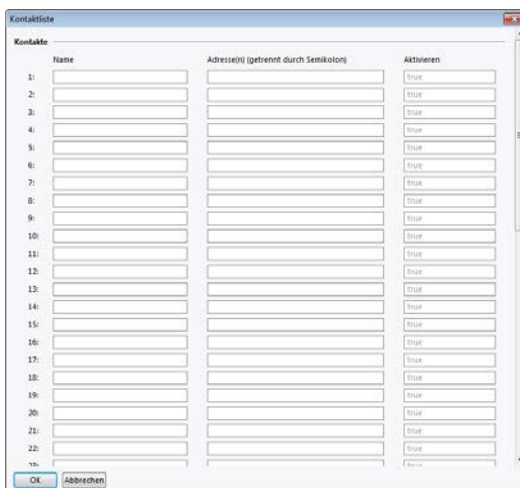
Die E-Mail-Übertragung und das E-Mail-Adressbuch werden über den Mail Manager konfiguriert:



Mit den Eigenschaften auf der Registerkarte „Mail“ können Sie den Mail Manager aktivieren bzw. deaktivieren und einen Namen für das Gerät angeben, auf dem Crimson ausgeführt wird. Dieser Name wird in E-Mail-Nachrichten verwendet, um den Absender der Nachricht zu identifizieren. Anwendungen verwenden in der Regel den Namen der Maschine, an die das Gerät angeschlossen ist, oder den Namen des Standorts, der überwacht wird.

19.6.1 Hinzufügen von Kontakten

Über die Schaltfläche „Contacts“ (Kontakte) kann auf das Adressbuch von Crimson zugegriffen werden:



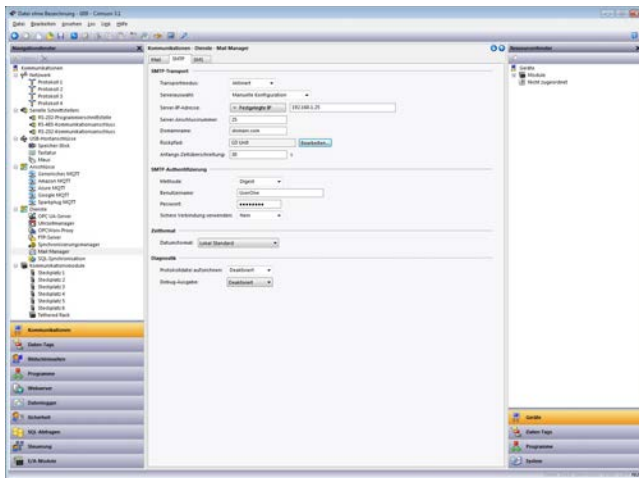
Für jeden Eintrag können Sie einen Anzeigenamen und eine Adresse eingeben sowie einen optionalen Ausdruck, der E-Mails an diesen Kontakt aktiviert bzw. deaktiviert. Die Adresse muss in einem Format angegeben werden, das für die jeweilige Übertragungsform geeignet ist. Bei SMTP-Namen müssen Sie beispielsweise das übliche Format `Name@Domäne` verwenden. SMS-Namen hingegen müssen als internationale Telefonnummern ohne das

führende Pluszeichen eingegeben werden. Sie können mehrere E-Mail-Adressen eingeben, indem Sie sie durch Semikolons voneinander trennen. Dies ermöglicht die Erstellung einfacher Verteilerlisten.

19.6.2 Konfigurieren von SMTP

Die Registerkarte „SMTP“ dient zum Konfigurieren des „Simple Mail Transport Protocol“. Dies ist das Standardprotokoll für das Senden von E-Mails über das Internet oder über andere TCP/IP-Netzwerke. SMTP-Adressen folgen dem bekannten Standard `Name@Domäne`.

Die Konfigurationsoptionen für den SMTP-Transport sehen Sie in der unten gezeigten Abbildung:



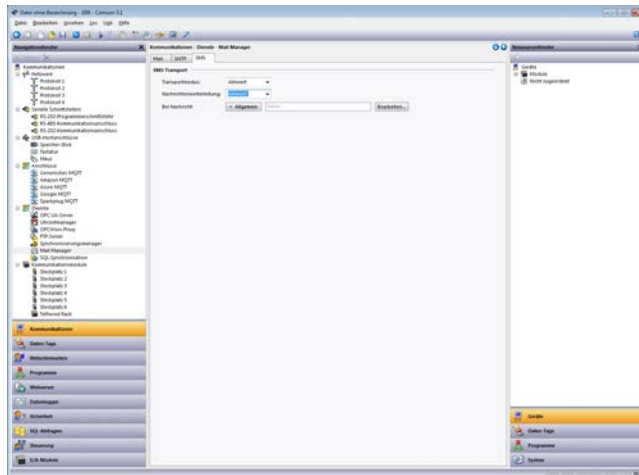
- Mit der Eigenschaft *Transport Mode* (Transportmodus) wird der Transport aktiviert bzw. deaktiviert. Beachten Sie, dass der Mail Manager über die Registerkarte „Mail“ aktiviert werden muss, damit der SMTP-Transport aktiviert werden kann. Beachten Sie auch, dass mindestens ein Transport aktiviert sein muss, damit der Mail Manager Nachrichten übermitteln kann.
- Die Eigenschaft *Server Selection* (Serverauswahl) gibt an, wie der Transport einen SMTP-Server ausfindig macht. Bei Auswahl von „Manual Selection“ (Manuelle Auswahl) muss manuell ein Server mit der Eigenschaft *Server IP Address* (Server IP-Adresse) bestimmt werden. Wenn „Configured via DHCP“ (Konfiguriert über DHCP) ausgewählt wird, muss mindestens ein Ethernet-Port für die Verwendung von DHCP konfiguriert werden, und der DHCP-Server des Netzwerks muss so konfiguriert werden, dass er über die DHCP-Option 69 einen SMTP-Server bestimmt.
- Mit der Eigenschaft *Server IP Address* (Server IP-Adresse) wird ein SMTP-Server bestimmt, wenn die manuelle Serverauswahl aktiviert ist. Der Server muss so konfiguriert sein, dass E-Mails vom Panel angenommen und Meldungen weitergeleitet werden, sofern für die Anwendung erforderlich.
- Die Eigenschaft *Server Port Number* (Server-Portnummer) definiert die TCP-Portnummer, die für SMTP-Sitzungen verwendet wird. Der Standardwert lautet 25. Dieser Wert ist für die meisten Anwendungen geeignet und muss nur angepasst werden, wenn der SMTP-Server für die Verwendung eines anderen Ports konfiguriert wurde.
- Die Eigenschaft *Domain Name* (Domänenname) gibt den Domännennamen an, der in dem Befehl HELO oder EHLO an den SMTP-Server übergeben wird. Die allermeisten SMTP-Server ignorieren diese Zeichenfolge. In dem unwahrscheinlichen Fall, dass Ihr SMTP-Server versucht, eine DNS-Suche durchzuführen, um die Identität des Clients zu bestätigen, müssen Sie ggf. etwas eingeben, das für Ihre DNS-Konfiguration geeignet ist.
- Die Eigenschaft *Reverse Path* (Antwortpfad) gibt die E-Mail-Adresse an, die als Absender der Nachrichten angegeben wird, die vom Zielgerät gesendet werden. Die Eigenschaft umfasst einen Anzeigenamen und eine E-Mail-Adresse. Da Crimson nicht zum Empfang von Nachrichten fähig ist, wird die E-Mail-Adresse häufig auf etwas gesetzt, bei Senden einer Antwort die Nachricht „Unzustellbar“ zurücksendet.

- Die Eigenschaft *Initial Timeout* (Anfangs-Timeout) gibt an, wie viele Sekunden der Mailclient darauf wartet, dass der SMTP-Server sein Begrüßungsbanner sendet. Einige Microsoft-Server versuchen, eine Microsoft-spezifische Authentifizierung mit Mailclients auszuhandeln, wodurch sich die Anzeige des Banners verzögert. Sie können diesen Zeitraum auf bis zu 2 Minuten oder mehr verlängern, wenn Sie mit solchen Servern arbeiten.
- Die Eigenschaft *Record Log File* (Protokolldatei aufzeichnen) kann aktiviert werden, um im Stammverzeichnis der Speicherkarte ein Protokoll aller SMTP-Interaktionen zu führen. Diese Datei kann beim Debuggen von SMTP-Vorgängen nützlich sein. Wenn sie jedoch ständig aktiviert ist, beeinträchtigt sie die Leistung ein wenig. Eine Protokolldatei kann nur erstellt werden, wenn eine Verbindung mit dem Mailserver eingerichtet wurde.
- Die Eigenschaft *Debug Output* (Debug-Ausgabe) wird verwendet, um die Ausgabe erweiterter Diagnoseinformationen an die verschiedenen Debug-Konsolen des Systems zu aktivieren bzw. zu deaktivieren. Diese Informationen sind technischer Art und nicht vollständig dokumentiert. Sie sind für den technischen Support von Red Lion oder für erfahrenere Benutzer bestimmt. Informationen zum Konfigurieren von Debug-Konsolen finden Sie im Kapitel *Erweitertes Debuggen*.
- Die Eigenschaft *Method* (Methode) gibt den Authentifizierungstyp an, der vom Client versucht werden soll. Die Auswahl „Digest“ besteht auf einer Authentifizierungsmethode, die das Kennwort in verschlüsselter Form sendet. Die Authentifizierung wird übersprungen, wenn der Server eine solche Methode nicht unterstützt. Bei Auswahl von „Basic“ (Grundlegend) wird versucht, die sicherere Technik zu verwenden, aber bei Bedarf wird auf eine gewöhnlich codierte Übertragung zurückgegriffen. Bei Auswahl von „None“ (Keine) wird keine Authentifizierung angestrebt. Ihr Server wird unter Umständen eine Authentifizierung erfordern. Wenden Sie sich an Ihren Netzwerkadministrator oder den Mail-Anbieter, um weitere Informationen über die Einstellung zu erhalten, die für Ihren Server verwendet werden sollte.
- Die Eigenschaften *Username* (Benutzername) und *Password* (Kennwort) stellen die optionalen Anmeldedaten für den oben beschriebenen Authentifizierungsprozess bereit.
- Mit der Option *Use Secure Connection* (Sichere Verbindung verwenden) wird SSL für die verschlüsselte E-Mail-Übertragung aktiviert. Viele öffentliche Server bestehen jetzt auf SSL, daher muss diese Option bei Verwendung dieser Dienste aktiviert werden. Die Option „TLS-SSL“ startet die Verbindung im sicheren Modus. Die Option „STARTTLS“ startet die Verbindung wie gewohnt, wechselt jedoch manuell in den sicheren Betrieb, sobald die Verbindung hergestellt ist. Wenden Sie sich an Ihren Netzwerkadministrator oder den Mail-Anbieter, um weitere Informationen über die Einstellung zu erhalten, die für Ihren Server verwendet werden sollte.

19.6.3 Konfigurieren von SMS

Die Registerkarte „SMS“ dient zum Konfigurieren des Short Message Service-Transports, der unterstützt wird, wenn in Verbindung mit dem Zielgerät ein GPRS-Modem verwendet wird. Die E-Mail-Adressen für den SMS-Transport sind internationale Telefonnummern ohne das Pluszeichen. Mit der Adresse 17175551111 würde beispielsweise eine Nachricht an ein Mobiltelefon oder ein anderes GSM-Gerät mit der Telefonnummer (717) 555-1111 in den USA gesendet.

Die Konfigurationsoptionen für den SMS-Transport sehen Sie in der nachfolgenden Abbildung:



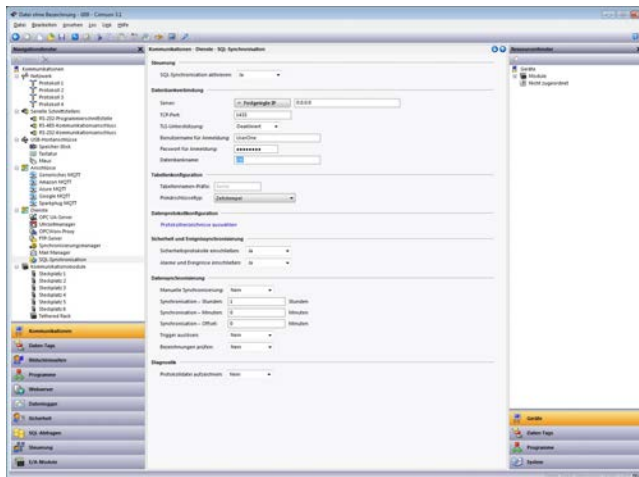
- Mit der Eigenschaft *Transport Mode* (Transportmodus) wird der Transport aktiviert bzw. deaktiviert.
- Die Eigenschaft *Message Relay* (Nachrichtenweiterleitung) dient zum Aktivieren bzw. Deaktivieren der SMS-Weiterleitungsfunktion von Crimson. Wenn diese Funktion aktiviert ist, kann ein Benutzer, der eine SMS-Nachricht empfängt, die an mehrere Empfänger gesendet wurde, auf diese Nachricht antworten und seine Antwort vom Crimson-Gerät an die anderen Empfänger übermitteln lassen. Das ermöglicht den Nachrichtenempfängern eine einfache Konferenzeinrichtung untereinander.
- Mit der Eigenschaft *On Message* (Bei Nachricht) wird eine Aktion definiert, die jedes Mal ausgeführt wird, wenn eine Nachricht empfangen wird. In der Aktion wird eine lokale Systemvariable namens *Data* definiert, die Zugriff auf die Nachricht selbst ermöglicht. Die Quellnummer der SMS wird der Nachricht vorangestellt und durch einen Doppelpunkt vom Nachrichtentext getrennt.

19.7 Verwenden der SQL-Synchronisierung

Mit der SQL-Synchronisierung kann eine Verbindung zu einer Microsoft SQL Server-Datenbank eingerichtet werden, um regelmäßig alle Protokolldateien mit einer externen Datenbank zu synchronisieren. Dieser Dienst ermöglicht eine direkte Integration in Systeme auf Unternehmensebene, ohne dass CSV-Dateien per FTP übertragen und dann von einem Serverskript erfasst werden müssen.

19.7.1 Konfigurieren des Dienstes

Die SQL-Synchronisierung wird über das zugehörige Symbol im Navigationsfenster konfiguriert:



- Mit der Eigenschaft *Server* wird die IP-Adresse des SQL-Servers angegeben.
- Mit der Eigenschaft *TCP Port* (TCP-Port) wird der TCP/IP-Port definiert, mit dem die SQL-Synchronisierung eine Verbindung herstellt. Dieser Port muss mit dem Port übereinstimmen, der auf dem Server für den TCP/IP-Zugriff konfiguriert ist. Weitere Auskünfte hierzu erteilt Ihr Datenbankadministrator.
- *Logon Username* (Anmeldebenutzername) und *Logon Password* (Anmeldekennwort) sind die Anmeldedaten, die beim Herstellen der Verbindung an den Server gesendet werden. Beim Kennwort wird immer die Groß-/Kleinschreibung beachtet. Ob die Groß-/Kleinschreibung auch beim Benutzernamen beachtet wird, hängt vom Server ab. Der Benutzer muss über die erforderlichen Rechte verfügen, um Tabellen zu erstellen und Daten zu schreiben.
- Mit der Eigenschaft *Database Name* (Datenbankname) wird der Name der Datenbank auf dem Server angegeben, mit der die SQL-Synchronisierung eine Synchronisierung durchführen wird.
- Mit der Eigenschaft *Table Name Prefix* (Tabellennamen-Präfix) wird ein Namespace für alle Tabellen angegeben, die von der SQL-Synchronisierung erstellt werden. Dieses Präfix wird mit einem Unterstrich dem Namen des Protokolls vorangestellt, um den Tabellennamen zu erstellen. Mithilfe dieser Eigenschaft können Namenskonflikte vermieden werden, wenn andere Geräte mit demselben Server synchronisiert werden.
- Mit der Einstellung *Primary Key Type* (Primärschlüsseltyp) wird gesteuert, wie die SQL-Synchronisierung Datenzeilen voneinander unterscheidet. Standardmäßig wird der Zeitstempel des Ereignis- oder Datenprotokolleintrags als Primärschlüssel für die einzelnen Tabellen verwendet. Doch bei Ereignis- und Datenprotokollen, die mehr als einmal pro Sekunde aktualisiert werden, ist der resultierende Schlüssel nicht eindeutig. Wählen Sie für diese Fälle *Auto-Increment* (Automatische Erhöhung) aus, um anzugeben, dass eine automatisch inkrementierende Zahl als Primärschlüssel der Tabelle verwendet werden soll. Dies kann auch erforderlich sein, wenn eine Nicht-UTC-Protokollierung verwendet wird und Anpassungen bezüglich der Sommerzeit vorgenommen werden, weil das dazu führen kann, dass zwei Einträge mit derselben Kombination von Uhrzeit und Datum vorliegen.

- Mithilfe der Option *Select Log Directories* (Protokollverzeichnisse auswählen) wird ein Dialogfeld angezeigt, in dem die Synchronisierung für jedes Datenprotokoll aktiviert bzw. deaktiviert werden kann. Dadurch ist es möglich, nur bestimmte Protokolle an den Server zu pushen und die Übertragung von Informationen zu vermeiden, die nur auf dem lokalen Gerät benötigt werden.
- Die Eigenschaften *Include Security Logs* (Sicherheitsprotokolle einbeziehen) und *Include Alarm and Events* (Alarmer und Ereignisse einbeziehen) steuern, ob die entsprechenden Ereignisprotokolle an den Server gepusht werden. Aktivieren Sie die Eigenschaften, wenn diese Informationen zentral benötigt werden.
- Die Eigenschaft *Sync Period* (Synchronisierung – Zeitraum) gibt an, wie oft die SQL-Synchronisierung eine Verbindung zum Server herstellt und Dateien überträgt. Sie wird in Stunden gemessen hat immer Mitternacht als Grundlage. Wenn Sie beispielsweise einen Wert von drei auswählen, finden Übertragungen um Mitternacht, um 03:00 Uhr, um 06:00 Uhr usw. statt.
- Die Eigenschaft *Sync Delay* (Synchronisierung – Verzögerung) gibt eine Abweichung in Minuten von der Standardzeit an, zu der Datenübertragungen stattfinden. Diese Eigenschaft kann verwendet werden, damit mehrere Terminals mit einem Server kommunizieren können, ohne dass alle Übertragungen gleichzeitig stattfinden und dadurch die Fähigkeiten des Ziels überlastet werden.
- Mit der Eigenschaft *Record Log File* (Protokolldatei aufzeichnen) wird bestimmt, ob die Aktivität der SQL-Synchronisierung auf der Compact Flash Karte protokolliert werden soll. Die protokollierten Informationen umfassen Befehle, die an den SQL-Server gesendet wurden, sowie Meldungen über aufgetretene Fehler. Diese Funktion kann beim Debuggen fehlgeschlagener Synchronisierungsversuche nützlich sein.

Kapitel 20 Freigeben von Ports

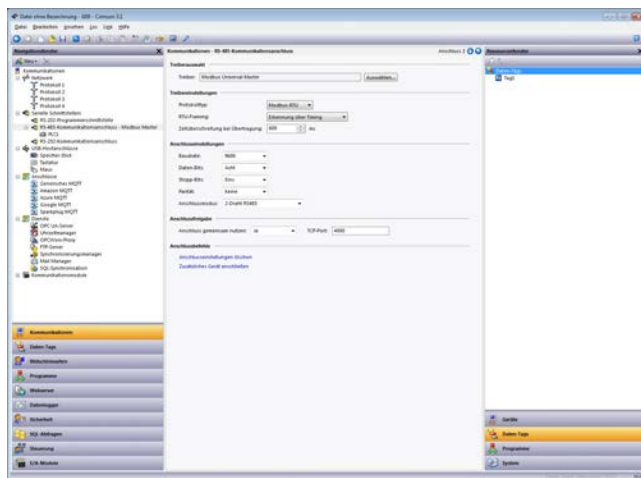
Crimson® 3.1 bietet eine Funktion zur Freigabe von Ports, mit der entweder physische oder virtuelle serielle Verbindungen mit jedem seriell angeschlossenen Gerät hergestellt werden können. Angenommen, Sie verwenden ein Bedienfeld mit einem programmierbaren Controller, doch da die SPS über nur einen seriellen Port verfügt, müssen Sie bei einer Änderung des Kontaktplanprogramms möglicherweise Kabel austauschen. Durch die Freigabe des Kommunikationsports zwecks Verbindung mit der SPS können Sie Daten direkt an den Controller senden, entweder von einem anderen seriellen Port oder mittels einer Verbindung über eine TCP/IP-Verknüpfung.

20.1 Aktivieren von TCP/IP

Der erste Konfigurationsschritt bei der Portfreigabe besteht darin, den Ethernet-Port wie an anderer Stelle in diesem Handbuch beschrieben zu aktivieren. Sie können sich zwar gegen die Verwendung des virtuellen seriellen Ports entscheiden, doch selbst die lokale Freigabe von Ports basiert auf dem TCP/IP-Protokoll, das erst verfügbar ist, wenn mindestens eine Netzwerkschnittstelle aktiviert ist.

20.2 Freigeben des erforderlichen Ports

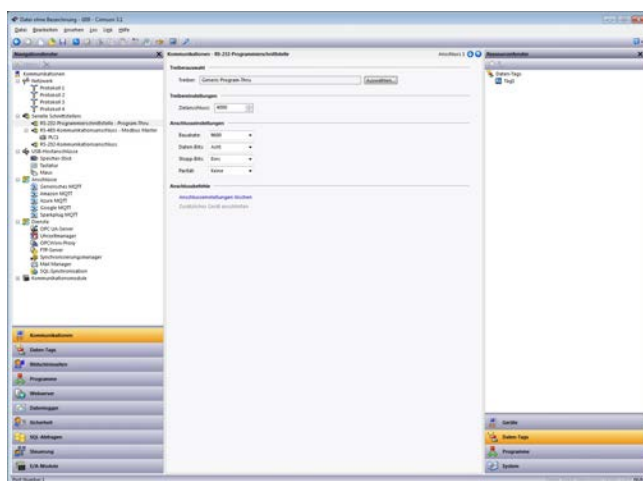
Der nächste Schritt besteht darin, den erforderlichen Port freizugeben. Hierzu wählen Sie für die Eigenschaft „Share Port“ (Port freigeben) die Einstellung „Yes“ (Ja) aus und geben dann eine geeignete TCP/IP-Portnummer ein, um genau anzugeben, wie der virtuelle Port angesprochen werden soll:



Wenn Sie die Porteinstellung 0 übernehmen, wird eine Zahl verwendet, die sich aus 4000 plus dem logischen Index des Ports zusammensetzt. Sie können eine beliebige Zahl verwenden, die nicht bereits von einem anderen TCP/IP-Protokoll verwendet wird. Wenn Sie keine Idee haben, empfehlen wir eine Zahl zwischen 4000 und 4099.

20.3 Herstellen einer Verbindung über einen anderen Port

Wenn Sie einen anderen Port auf dem Zielgerät verwenden möchten, um Daten an den freigegebenen Port weiterzuleiten, müssen Sie den Treiber „Generic Program Thru“ für diesen Port auswählen. Anschließend müssen Sie diesen Treiber mit der TCP/IP-Portnummer des Ports konfigurieren, den Sie freigegeben haben. Im unten gezeigten Beispiel werden Daten vom Programmierport an eine SPS übertragen, die über den Kommunikationsport RS485 verbunden ist:



Beachten Sie, dass die Baudrate und andere Porteeinstellungen in den meisten Fällen nicht mit den Einstellungen des freigegebenen Ports übereinstimmen müssen, da Crimson die Konvertierung durchführt. Hier gibt es jedoch eine Ausnahme, nämlich wenn ein Gerät große Datenmengen überträgt, ohne Antworten von dem anderen Gerät zu erhalten. In diesem Fall darf das Gerät, das die größeren Übertragungen durchführt, keine höhere Baudrate verwenden als das empfangende Gerät. Andernfalls verfügt Crimson möglicherweise nicht über genügend Speicher, um die Daten zu puffern, während auf die erneute Übertragung gewartet wird.

Im oben gezeigten Beispiel müssten Sie, um den freigegebenen Port zu verwenden, einen freien seriellen Port am PC mit dem Programmierport des Zielgeräts verbinden und die SPS-Programmiersoftware so konfigurieren, dass sie mit diesem COM-Port kommuniziert. Sobald der PC mit der Übermittlung an die SPS beginnt, wird die gesamte Kommunikation zwischen Crimson und der SPS unterbrochen, und die beiden Ports des Zielgeräts werden in der Software so verbunden, dass der PC direkt mit der SPS zu kommunizieren scheint. Wenn mehr als eine Minute lang keine Daten übertragen werden, wird die Kommunikation zwischen Crimson und der SPS wieder aufgenommen.

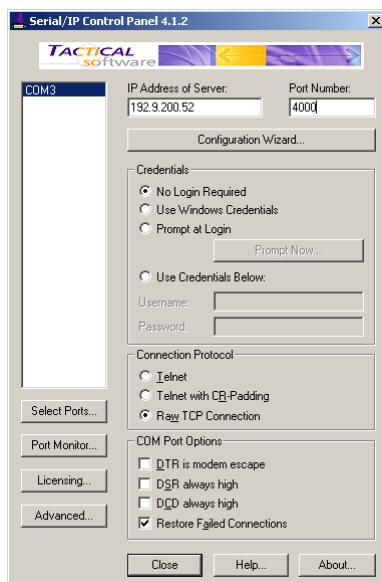
20.4 Herstellen einer Verbindung über Ethernet

Anstatt einen zusätzlichen seriellen Port auf Ihrem PC und auf dem Crimson-basierten Gerät zu verwenden, können Sie ein Dienstprogramm eines Drittanbieters verwenden und auf Ihrem PC sogenannte virtuelle serielle Ports erstellen. Diese werden von Anwendungen als physische COM-Ports wahrgenommen, doch in Wahrheit werden Daten über TCP/IP an ein externes Gerät gesendet bzw. von diesem empfangen. Wenn Sie eines dieser Dienstprogramme installieren und so konfigurieren, dass es das Crimson-basierte Gerät anspricht, haben Sie einen seriellen Zugang zu jedem mit diesem Gerät verbundenen Gerät, ohne zusätzliche Kabel zu benötigen. Tatsächlich müssen auf dem PC gar keine physischen seriellen Ports verfügbar sind. Dies ist sehr nützlich, wenn Sie mit modernen Laptops arbeiten, bei denen ein COM-Port häufig ein kostspieliges Extra ist.

Für virtuelle serielle Schnittstellen stehen verschiedene Dienstprogramme von Drittanbietern zur Verfügung. Im Freeware-Bereich bietet das Unternehmen HW Group (<http://www.hw-group.com>) das Dienstprogramm HW Virtual Serial Port an. Es werden noch andere Freeware-Porttreiber angeboten, die meist von derselben Quellbasis abgeleitet zu sein scheinen. Im kommerziellen Bereich bietet das Unternehmen Tactical Software (<http://www.tacticalsoftware.com>) die Software Serial/IP für etwa 100 \$ pro Port an.

Auch wenn die verschiedenen Freeware-Treiber sicherlich viele zufriedene Benutzer für sich gewinnen konnten, haben wir festgestellt, dass diese Treiber auf bestimmten PCs zu Stabilitätsproblemen führen können. Serial/IP von Tactical Software ist daher das einzige Paket, für das wir Unterstützung anbieten können. Die nachstehenden Informationen beruhen auf der Annahme, dass Sie dieses Paket verwenden.

Um einen virtuellen seriellen Port zu erstellen, öffnen Sie das Konfigurationsdialogfeld von Serial/IP und wählen den Namen des COM-Ports aus, den Sie definieren möchten. Dabei handelt es sich in der Regel um den ersten freien COM-Port nach den Ports, die bereits den auf dem PC installierten physischen Ports und Modems zugeordnet sind. Geben Sie als Nächstes die IP-Adresse des Crimson-basierten Geräts ein und dann die TCP/IP-Portnummer, die Sie beim Freigeben des Ports zugewiesen haben. Die Konfiguration im unten gezeigten Beispiel ist auf die vorherigen Beispiele in diesem Dokument abgestimmt. Achten Sie abschließend darauf, dass „Raw TCP Connection“ (Raw-TCP-Verbindung) ausgewählt ist, und schließen Sie das Dialogfeld „Serial/IP“.



Sie können jetzt jede Windows-basierte Software so konfigurieren, dass der neu erstellte COM-Port für Downloads verwendet wird. Wenn die Software die Verbindung öffnet, unterbricht Crimson die Kommunikation am freigegebenen Port. Dann werden Daten zwischen der PC-Software und der externen SPS ausgetauscht, als wären sie direkt miteinander verbunden. Wenn der Port geschlossen wird oder wenn eine Minute lang keine Daten übertragen werden, wird die Kommunikation fortgesetzt.

Sofern Sie die entsprechende Anzahl an Lizenzen für Serial/IP erworben haben, können Sie so viele virtuelle Ports erstellen, wie Sie benötigen. Das bedeutet, dass Sie von ein und demselben PC mit mehreren Geräten verbunden sein und über das jeweilige Programmierpaket Daten auf diese Geräte herunterladen können, ohne ein einziges Kabel anschließen oder trennen zu müssen. Diese Funktion ist sehr wertvoll, wenn Sie ein komplexes System mit vielen Geräten verwenden.

20.5 Rein virtuelle Ports

Gelegentlich möchten Sie vielleicht einen freien seriellen Port auf einem Crimson-basierten Gerät verwenden, um den Zugriff auf ein externes Gerät zu ermöglichen, das in der Datenbank nicht anderweitig referenziert ist. Dann wird der freie Port effektiv als externer serieller Server eingesetzt. Konfigurieren Sie dazu den Port auf die übliche Weise und wählen Sie für diesen Port den Treiber „Virtual Serial Port“ aus. Geben Sie dann den Port wie oben beschrieben frei und machen Sie ihn über TCP/IP zugänglich. Der Treiber „Virtual Serial Port“ führt keine eigenen Kommunikationsaktivitäten durch, lässt aber dennoch zu, dass das Gerät für den externen Zugriff freigegeben wird.

20.6 Einschränkungen

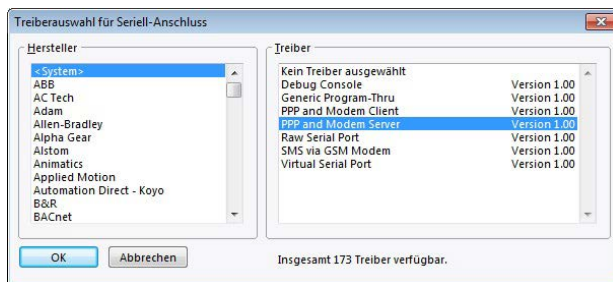
Beachten Sie, dass einige SPS-Programmierpakete möglicherweise nicht mit virtuell oder physisch freigegebenen Ports funktionieren. Die Probleme reichen von engen Timeouts, die Crimson keine Zeit für die Übermittlung der Daten an die SPS lassen, und einer Abhängigkeit von der Übertragung von Pausensignalen oder der Manipulation von Hardware-Handshaking-Lines bis hin zu einem DOS-artigen Portzugriff, bei dem das Paket den virtuellen seriellen Port nicht sehen kann. Glücklicherweise sind diese Probleme selten und die meisten Pakete werden eine reibungslose Kommunikation bieten, als wären sie direkt mit der betreffenden SPS verbunden.

Kapitel 21 Verwenden von Modems

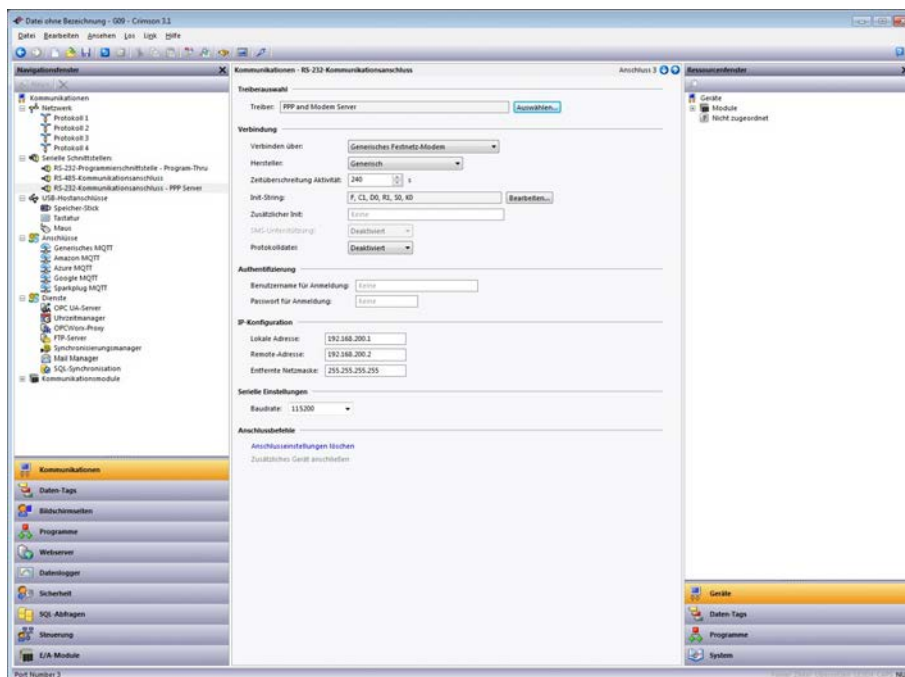
In diesem Kapitel wird erklärt, wie Sie Crimson® 3.1 für die Verwendung von Modems bzw. für die Verwendung direkter serieller Verbindungen mit Windows-Computern konfigurieren können. Die Modem-Unterstützung von Crimson beruht gänzlich auf der Ausführung von TCP/IP über das Point-to-Point-Protokoll, das auch als PPP bezeichnet wird. Während Protokolle wie Modbus eine einzelne Konversation zwischen zwei beliebigen Geräten gestatten, hat PPP mehr Ähnlichkeit mit einer Ethernet-Verbindung, insofern als es eine unbegrenzte Anzahl von logischen Verbindungen über einen einzigen physischen Link zulässt. Eine einzelne PPP-Verbindung kann daher einen gleichzeitigen Zugriff auf die TCP/IP-Downloadfunktion des Bedienfelds, dessen Webserver, dessen freigegebene seriellen Ports und auf sämtliche TCP/IP-Protokolle gestatten, die definiert wurden.

21.1 Hinzufügen einer Dial-In-Verbindung

Um einer Datenbank eine Dial-In-Verbindung (Einwählverbindung) hinzuzufügen, wählen Sie die Kategorie „Communications“ (Kommunikationen) aus und navigieren zu dem seriellen Port, über den die Verbindung hergestellt werden soll. Klicken Sie neben der Eigenschaft „Driver“ (Treiber) auf die Schaltfläche „Pick“ (Auswählen) und wählen Sie im Abschnitt „System“ den Treiber „PPP and Modem Server“ (PPP- und Modem-Server) aus.



Im Bearbeitungsfenster wird nun die Modemkonfiguration angezeigt:



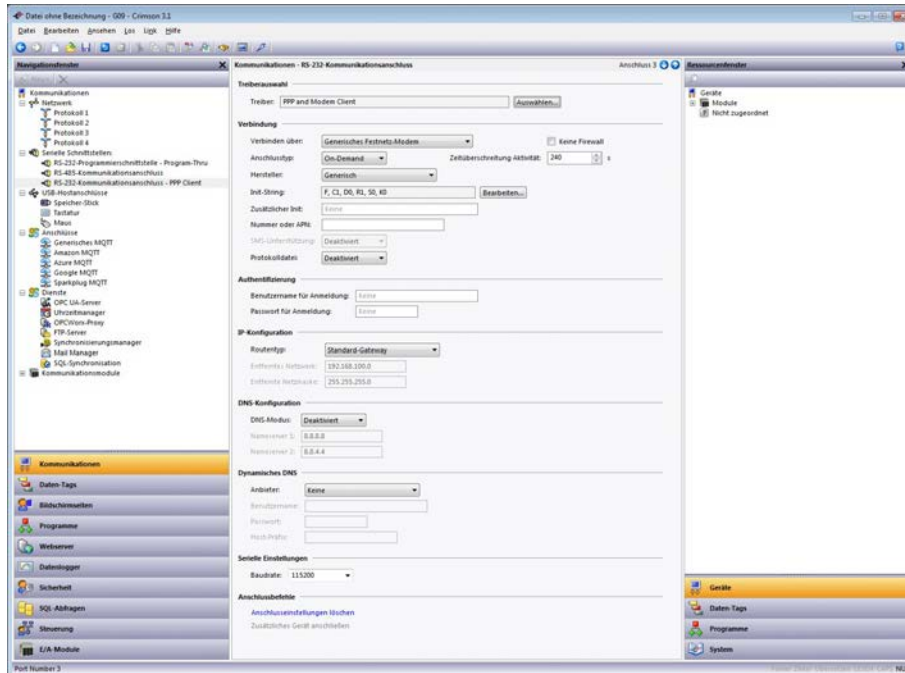
Für das Modem gibt es die folgenden Konfigurationsoptionen:

- Mit der Eigenschaft *Connect Using* (Verbinden über) wird das physische Gerät zur Herstellung der Verbindung ausgewählt. Die Geräte, die über serielle Ports unterstützt werden, sind direkte serielle Verbindungen zu Computern mit dem Betriebssystem Microsoft Windows; generische Festnetzmodems, die den Hayes-Befehlssatz implementieren; sowie GSM-Modems, die branchenübliche GSM-Befehle implementieren. Wenn GSM-Geräte für die Einwahl verwendet werden, müssen sie im leitungsvermittelten Datenmodus konfiguriert werden.
- Mithilfe der Eigenschaft *Manufacturer* (Hersteller) wird unter den Herstellern oder Modellen ausgewählt, für die spezielle Konfigurationen entwickelt und in Crimson gespeichert wurden. Wenn Sie hier die Einstellung „Generic“ (Generisch) übernehmen, können Sie die Einstellungen für Initialisierungszeichenfolgen und dergleichen anpassen. Bei Fragen zu den Einstellungen, die für ein bestimmtes Modem erforderlich sind, wenden Sie sich an den technischen Support.
- Mit der Eigenschaft *Activity Timeout* (Zeitüberschreitung Aktivität) wird angegeben, wie viel Zeit vergehen muss, in der das Crimson-Gerät kein Paket über den PPP-Link sendet, damit die Verbindung beendet wird. Bei Dial-In-Verbindungen wird davon ausgegangen, dass das verbindende Gerät vertrauenswürdig ist. Daher werden keinerlei optionale Pakete herausgefiltert, die dazu führen könnten, dass die Verbindung über lange Zeiträume aktiv bleibt. Selbst wenn Sie eine permanente Verbindung wünschen, müssen Sie dennoch einen Timeout-Wert eingeben, damit tote Links erkannt werden können. Dies bedeutet, dass so genannte permanente Verbindungen dennoch gelegentlich ausfallen können, aber auf jeden Fall sofort wiederhergestellt werden.
- Mit der Eigenschaft *Init String* (Init-String) werden während der Initialisierungssequenz bestimmte Befehle aktiviert bzw. deaktiviert. Sie wird automatisch konfiguriert, wenn unter „Manufacturer“ (Hersteller) eine bestimmte Einstellung eingegeben wird.
- Die Eigenschaft *Additional Init* (Zusätzlicher Init) wird bei nicht-direkten Links verwendet und bietet eine Reihe von AT-Befehlen zur Initialisierung des Modems. Das Präfix AT ist nicht erforderlich. Es können mehrere Befehle kombiniert werden, indem Sie sie einfach nacheinander auflisten. Die genaue Zeichenfolge, die für Ihr Modem benötigt wird, richtet sich nach dessen interner Software. Wenn Sie den technischen Support um Hilfe bitten, sollten Sie daher Details zum Hersteller und Modell griffbereit haben.
- Die Eigenschaft *SMS Support* (SMS-Unterstützung) dient zur Aktivierung des Short Message Service für GSM-Modems. Damit SMS ordnungsgemäß funktioniert, müssen Sie auch den SMS-Transport im Mail Manager aktivieren. Dies ist an anderer Stelle in diesem Handbuch beschrieben.
- Mit der Eigenschaft *Log File* (Protokolldatei) wird die Protokollierung des Datenaustauschs mit dem Modem auf der Speicherkarte aktiviert. Diese Datei kann bei der ersten Modem-Einrichtung für Debugging-Zwecke oder bei dem Versuch, die richtigen Konfigurationsoptionen zu finden, eingesetzt werden. Achten Sie darauf, diese Funktion zu deaktivieren, nachdem die richtige Modem-Konfigurationssequenz eingerichtet wurde. Andernfalls wird die Leistung erheblich beeinträchtigt.
- Mit den Eigenschaften *Logon Username* (Anmeldebenutzername) und *Logon Password* (Anmeldekennwort) werden die Anmeldedaten definiert, die der Remote-Client bereitstellen muss, damit er eine Verbindung mit diesem Gerät herstellen darf. Beim Benutzernamen wird die Groß- und Kleinschreibung nicht beachtet, wohl aber beim Kennwort. Die PPP-Implementierung von Crimson fordert ihren Peer zur Verwendung der CHAP-Authentifizierung auf, damit Kennwörter nicht in Klartext gesendet oder empfangen werden. Wenn der Remote-Client CHAP nicht unterstützt, wird auf PAP zurückgegriffen.
- Mit der Eigenschaft *Local Address* (Lokale Adresse) wird die IP-Adresse definiert, die dem lokalen Ende der Verbindung zugeordnet werden soll. Dies ist daher die IP-Adresse des Crimson-Geräts für diesen Link. Beachten Sie, dass diese nicht mit der IP-Adresse des Ethernet-Ports des Geräts identisch sein darf, da jede physische Schnittstelle, die IP verwendet, eine eindeutige Adresse aufweisen muss. Der Standardwert funktioniert in den meisten Situationen, es sei denn, Ihr Netzwerkdesign erfordert eine andere Einstellung.

- Die Eigenschaft *Remote Address* (Remote-Adresse) wird verwendet, um die IP-Adresse zu definieren, die dem fernen Ende der Verbindung zugeordnet werden soll. Sie wird zusammen mit der Eigenschaft *Remote Mask* (Remote-Maske) verwendet, um zu bestimmen, welche Pakete an diese Verbindung weitergeleitet werden. Bei den meisten Anwendungen wird die Maske 255.255.255.255 verwendet. Hierdurch wird Crimson angewiesen, über diese Schnittstelle nur die Pakete zu senden, die direkt an den Remote-Client gehen. Bei der Maske 0.0.0.0 hingegen können alle Pakete, die nicht speziell mit einer anderen Schnittstelle übereinstimmen, an den Remote-Client und von dort wahrscheinlich an den gewünschten Host weitergeleitet werden. Dazwischen liegende Masken können verwendet werden, um genau zu steuern, welche Pakete gesendet werden.

21.2 Hinzufügen einer Dial-Out-Verbindung

Dial-Out-Verbindungen werden genau wie oben beschrieben hinzugefügt, mit der Ausnahme, dass für den erforderlichen Kommunikationsport der Treiber *PPP and Modem Client* ausgewählt werden sollte. Die Konfigurationsoptionen für dieses Modem sind unten dargestellt:



- Für die Eigenschaft *Connect Using* (Verbinden über) gilt das gleiche wie bei Dial-In-Verbindungen. Zusätzlich werden GPRS- und HSPA+-Verbindungen über ein GSM-Modem unterstützt. Diese Verbindungen unterscheiden sich von CSD-Verbindungen insofern, als dass sie wesentlich höhere Geschwindigkeiten erreichen und Gebühren in der Regel nach der Datenmenge berechnet werden und nicht nach der Dauer der Verbindung. GPRS- und HSPA+ Verbindungen können daher für permanente Verbindungen konfiguriert werden, es sei denn, es muss eine Ausfallzeit bereitgestellt werden, damit SMS-Nachrichten übertragen werden können.
- Mit der Eigenschaft *No Firewall* (Keine Firewall) wird der Firewallschutz deaktiviert, der andernfalls für Dial-Out-Verbindungen bereitgestellt wird. Dieser Schutz verhindert, dass mit dieser Schnittstelle eingehende Verbindungen hergestellt werden. Ferner verhindert er, dass das Gerät bestimmte Diagnosepakete sendet, die einem Hacker Anschluss über das System geben würden oder aber von einem Angreifer genutzt werden könnten, um eine Verbindung aktiv zu halten, obwohl keine Datenübertragung stattfindet. Wenn Sie über diese Verbindung eine direkte Verbindung mit dem Internet herstellen, sollten Sie die Firewall normalerweise aktiviert lassen. Die Firewall sollte nur für Verbindungen zu Unternehmensnetzwerken oder anderen kontrollierten Umgebungen deaktiviert werden.
- Mit der Eigenschaft *Connection Type* (Verbindungstyp) können Sie angeben, ob diese Verbindung dauerhaft sein soll oder ob sie bei dem Versuch, Daten an Hosts zu übertragen, die über diese Schnittstelle erreichbar sind, automatisch hergestellt werden soll. Wenn Sie den Verbindungstyp „On-Demand“ (Auf Anforderung) auswählen, müssen Sie die Zeitüberschreitung festlegen, nach deren Ablauf die Verbindung beendet wird, wenn keine Pakete vom Crimson-Gerät übertragen wurden.
- Mit der Eigenschaft *Number or APN* (Nummer oder APN) wird die Telefonnummer konfiguriert, die ein herkömmliches Modem wählt, bzw. die APN, die für den Zugriff auf den Dienst erforderlich ist, der mit der SIM verknüpft ist, die in einem GSM-Modem installiert ist. Diese Informationen stellt in der Regel der GSM-Dienstanbieter bereit.

- Mit den Eigenschaften *Logon Username* (Anmeldebenutzername) und *Logon Password* (Anmelde-kennwort) werden die Anmeldedaten definiert, die bei dem Versuch, eine Verbindung zu initialisieren, an den Remote-Server übergeben werden. Beim Benutzernamen wird die Groß- und Kleinschreibung nicht beachtet, wohl aber beim Kennwort. Die PPP-Implementierung von Crimson fordert ihren Peer zur Verwendung der CHAP-Authentifizierung auf, damit Kennwörter nicht in Klartext gesendet oder empfangen werden. Wenn der Remote-Server CHAP nicht unterstützt, wird auf PAP zurückgegriffen.
- Mit der Eigenschaft *Route Type* (Routentyp) werden die Daten festgelegt, die über diese Schnittstelle übertragen werden. Bei On-Demand-Verbindungen bestimmt dies quasi, wann die Verbindung akti-viert wird. Bei Auswahl von *Default Gateway* (Standard-Gateway) werden alle Pakete, die nicht mit der Adresse und Netzwerkmaske der Ethernet-Verbindung übereinstimmen, an diese Schnittstelle gesen-det. Beachten Sie, dass der Ethernet-Port in diesem Modus die Gateway-Einstellung 0.0.0.0 haben muss. Andernfalls nimmt er alle Pakete an sich und es bleiben keine zum Aktivieren des Modems übrig. Wenn *Specific Network* (Bestimmtes Netzwerk) ausgewählt ist, müssen Sie die Adresse und Netzmaske angeben, die das Netzwerk definieren, an das Pakete weitergeleitet werden.
- Die Optionen unter *DNS Configuration* (DNS-Konfiguration) dienen zum Konfigurieren der DNS-Ser-ber, die über diese Schnittstelle erreichbar sind. Im Modus „Manual“ (Manuell) können die Server ein-zeln angegeben werden. Im Modus „Automatic“ (Automatisch) werden die Server verwendet, die während des PPP-Verbindungsprozesses ausgehandelt werden. Die Standardserver, die im manuel-len Modus für neue Datenbanken bereitstehen, sind die öffentlichen DNS-Server 8.8.8.8 und 8.8.4.4 von Google.
- Mit den Optionen unter *Dynamic DNS* (Dynamisches DNS) wird gesteuert, ob die öffentliche IP-Adresse dieser Schnittstelle mit einem dynamischen DNS-Dienst von Drittanbietern registriert werden soll. Die Eigenschaft *Provider* (Anbieter) wählt den dynamischen DNS-Anbieter aus, der verwendet werden soll. Aktuelle Versionen unterstützen nur DynDNS mit einem von zwei Domänensuffixen. Die Eigenschaften *Username* (Benutzername) und *Password* (Kennwort) geben die mit dem DynDNS-Konto verknüpften Anmeldedaten an und „Host Prefix“ (Host-Präfix) gibt den Namen an, der der Domäne „dnsalias.com“ oder „dyndns.org“ vorangestellt wird, wenn dieses Gerät registriert wird.
- Die übrigen Eigenschaften sind genau wie bei den Dial-In-Verbindungen.

21.3 Hinzufügen einer SMS-Verbindung

SMS-Verbindungen werden verwendet, wenn die Text-Messaging-Funktionalität benötigt wird, aber weder Dial-In- noch Dial-Out-PPP-Verbindungen hergestellt werden. Sie werden wie oben beschrieben konfiguriert, mit der Ausnahme, dass das Gerät *SMS via GSM Modem* (SMS über GSM-Modem) für den erforderlichen Port ausgewählt werden muss. Die Eigenschaften für diesen Treiber stellen eine Untergruppe der Eigenschaften für Dial-In-Verbindungen dar. Die SMS-Unterstützung ist bei diesem Treiber immer aktiviert, aber beachten Sie, dass SMS-Messaging nur funktioniert, wenn Sie den SMS-Transport im Mail Manager aktiviert haben.

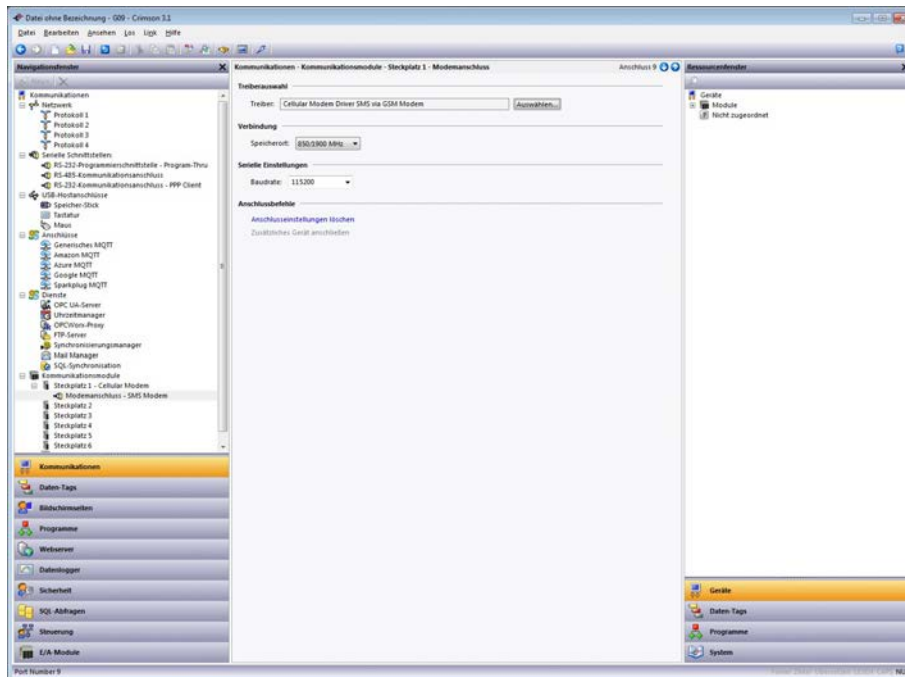
21.3.1 Verarbeitung von SMS-Nachrichten

Wenn das SMS-Messaging aktiviert ist, weist Crimson das GSM-Modem an, alle fünf Sekunden nach neuen eingehenden oder ausgehenden Nachrichten zu suchen. Eingehende Nachrichten werden an den Mail Manager weitergeleitet, der sie je nach Konfiguration an andere Benutzer weiterleiten kann. Beachten Sie, dass es nicht möglich ist, nach Meldungen zu suchen, während das Modem mit einer CSD- oder GSM-Sitzung verbunden ist. Daher sollten Sie beim Arbeiten mit SMS keine permanenten Verbindungen verwenden. Beachten Sie auch Folgendes: Wenn mehr als ein GSM-Modem konfiguriert ist, können zwar alle Modems Nachrichten empfangen, aber nur das letzte Modem wird für das Senden verwendet.

21.4 Erweiterungskarten für Modems

Die Ausführungen weiter oben beziehen sich auf das Verfahren, mit dem ein Modem über eine der seriellen Schnittstellen des Geräts angeschlossen wird. Für viele Geräte gibt es aber auch Modem-Erweiterungsmodule und -Erweiterungskarten. Diese werden konfiguriert, indem Sie die Erweiterungsoption wie im Kapitel „Verwenden von Kommunikationen“ beschrieben hinzufügen und anschließend den entsprechenden Treiber für den Port auswählen, der dann im Navigationsfenster angezeigt wird.

In diesem Beispiel wird ein Graphite-Erweiterungsmodul verwendet, um SMS-Konnektivität bereitzustellen:



21.5 Überprüfen des Modemstatus

Um die Fehlersuche bei Modemverbindungen zu erleichtern, stellt Crimson die Funktion `GetInterfaceStatus()` bereit. Diese Funktion arbeitet mit einem einzelnen Argument, bei dem es sich um den numerischen Index der erforderlichen Schnittstelle handelt. Die Schnittstelle Null ist die interne Loopback-Schnittstelle. Als Nächstes werden alle aktivierten Ethernet-Schnittstellen aufgeführt, gefolgt von den PPP-Schnittstellen. Bei Systemen mit einem einzelnen Ethernet-Port z. B. trägt die erste PPP-Schnittstelle den Index 1.

Die Funktion gibt eine Zeichenfolge zurück, die anhand der folgenden Tabelle interpretiert werden kann:

Status	Bedeutung
CLOSED	Die Schnittstelle wurde noch nicht initialisiert. Dieser Status tritt nur kurz während des Systemstarts auf.
INIT	Das Modem wird initialisiert. Wenn die Verbindung in diesem Status verbleibt, enthalten die an das Modem gesendeten Init-Strings wahrscheinlich Fehler.
IDLE	Die Verbindung ist im Leerlauf. GSM-Modems geben am Ende der Zeichenfolge eine Zahl zurück, die die Signalstärke anzeigt. In der nächsten Tabelle wird erläutert, wie Sie diese Werte interpretieren.
SMS	Das Modem sendet SMS-Nachrichten, oder das Modem wird abgefragt, um festzustellen, ob neue SMS-Nachrichten verfügbar sind. Wenn SMS-Messaging für ein Modem aktiviert ist, wird dieser Status alle fünf Sekunden kurz angezeigt.
CONNECTING	Das Modem baut eine Verbindung auf. Dieser Status wird in der Regel nur für Client-Verbindungen angezeigt und bedeutet, dass ein Anruf getätigt wird.
LISTENING	Das Modem wartet auf einen Anruf. Dieser Status wird nur für Serververbindungen angezeigt. Beachten Sie, dass GSM-Modems auch den Status IDLE zurückgeben, während sie auf einen Anruf warten, um die Signalstärke anzuzeigen.
ANSWER	Das Modem nimmt einen Anruf entgegen und versucht, die Baudrate für die Verbindung auszuhandeln. Dieser Status wird nur für Serververbindungen angezeigt. Wenn die Verbindung aufgebaut wurde, wechselt das Modem in den Status CONNECTED.
CONNECTED	Das Modem hat eine Verbindung aufgebaut. Dieser Status besteht nur kurze Zeit, denn nach einer kurzen Verzögerung beginnt die LCP-Aushandlung.
NEG LCP	Die Verbindung handelt LCP-Optionen aus. Dieser Vorgang entscheidet über eine Reihe von Verbindungsprotokoll-Einstellungen, die sowohl für den Client als auch für den Server akzeptabel sind.
AUTH	Die Verbindung führt die Authentifizierung durch, um sicherzustellen, dass die richtigen Benutzeranmeldedaten verwendet werden.
NEG IPCP	Die Verbindung handelt IPCP-Optionen aus. Dieser Vorgang entscheidet über eine Reihe von Netzwerkprotokoll-Einstellungen, die sowohl für den Client als auch für den Server akzeptabel sind.
UP	Die Verbindung ist aktiv und IP-Daten können ausgetauscht werden.
HANGING UP	Das Modem trennt die Verbindung. Dieser Status besteht nur kurze Zeit, bevor das Modem in den Status IDLE zurückkehrt.

Die von GSM-Modems zurückgegebenen Signalstärkewerte haben folgende Bedeutung:

Wert	Signalstärke
0,00	-113 dBm oder weniger
1,0	-111 dBm
2-30	-109 dBm bis -52 dBm in Schritten von 2 dBm
31	-51 dBm oder mehr
99	Die Signalstärke kann nicht ermittelt werden.

Mobiltelefone interpretieren diese Werte in der Regel wie folgt, wenn die Signalstärke angezeigt wird:

Wert	Stärke	Anzahl der Balken
5 oder weniger	-103 dBm oder weniger	Eins
6 bis 9	-101 dBm bis -95 dBm	Zwei
10 bis 14	-93 dBm bis -85 dBm	Drei
15 oder mehr	-83dBm oder stärker	Vier

21.6 Verwenden von mehreren Schnittstellen

Crimson unterstützt bis zu zwei unabhängige Modemverbindungen. In Kombination mit den ein oder zwei Ethernet-Ports, die vom Zielgerät bereitgestellt werden, ergibt dies bis zu vier verschiedene IP-Schnittstellen, die jeweils entsprechend der Konfigurationsparameter arbeiten, die für die einzelnen Verbindungen definiert wurden. In diesem Abschnitt wird beschrieben, wie diese verschiedenen Schnittstellen interagieren und wie Crimson entscheidet, wohin die einzelnen Datenpakete gesendet werden.

21.6.1 Schnittstellenauswahl

Jede Schnittstelle hat eine IP-Adresse und eine Netzwerkmaske, anhand denen entschieden wird, ob Pakete an diese Schnittstelle weitergeleitet werden. Wenn eine Ethernet-Schnittstelle z. B. mit der IP-Adresse 192.168.1.0 und der Netzwerkmaske 255.255.255.0 konfiguriert ist, werden alle Pakete an IP-Adressen, die mit 192.168.1 beginnen, an diese Schnittstelle gesendet. Wenn eine On-Demand-Modemverbindung die Remote-IP-Adresse 192.168.2.2 und die Netzwerkmaske 255.255.255.255 hat, führt das Senden eines Pakets an die Adresse 192.168.2.2 dazu, dass die Verbindung hergestellt wird.

Beachten Sie, dass dieser Mechanismus nur ein Paket an eine einzelne Schnittstelle sendet. Dies bedeutet, dass Schnittstellen unterschiedliche Netzwerkadressen aufweisen müssen, die durch ihre IP-Adresse und Netzwerkmaske definiert sind. Wenn Sie gegen diesen Grundsatz verstoßen, werden Pakete nicht an die zweite Schnittstelle mit dieser Netzwerkadresse weitergeleitet, und die Kommunikation an diesem Port schlägt fehl. Zum Beispiel wird davon abzuraten, einen Ethernet-Port mit der Adresse 192.168.100.1 zu konfigurieren und den anderen mit 192.168.100.2, da Pakete für das Netzwerk 192.168.100.0 nur an den ersten Port gesendet werden.

21.6.2 Standardroute

Darüber hinaus kann eine einzelne Schnittstelle auch eine Standardroute festlegen. Diese Standardroute wird für die Verarbeitung von Paketen verwendet, die nicht eindeutig mit einer anderen Schnittstelle übereinstimmen. Die zum Konfigurieren der Route verwendete Methode richtet sich nach dem Schnittstellentyp. Dies ist in der folgenden Tabelle dargestellt:

schnittstelle	So legen Sie Standardroute fest
Ethernet	Geben Sie für die Eigenschaft <i>Gateway</i> einen Wert von nicht Null ein.
Dial-In	Geben Sie für <i>Remote Mask</i> (Remote-Maske) 0.0.0.0 ein.
Dial-Out	Wählen Sie für die Eigenschaft <i>Route Type</i> (Routentyp) die Einstellung „Default Gateway“ (Standard-Gateway) aus.

Auch hier gilt, dass nur eine einzelne Schnittstelle eine Standardroute festlegen kann. Ein Bedienfeld kann beispielsweise mit einer Reihe von Ethernet-Geräten mit der IP-Adresse 192.168.1.0 und der Netzwerkmaske 255.255.255.0 verbunden werden, ohne dass ein Gateway festgelegt wird. Eine On-Demand-Modemverbindung kann so konfiguriert werden, dass bei Auftreten eines Alarms auf einen Internetdienstanbieter zugreifen kann, um E-Mails zu senden. Deren *Route Type* (Routentyp) wird auf „Default Gateway“ (Standard-Gateway) eingestellt, sodass bei ihr alle Pakete an IP-Adressen eingehen, die nicht mit dem für den Ethernet-Port definierten Netzwerk übereinstimmen. Der SMTP-Server wird als 24.104.0.39 konfiguriert. Das bedeutet, dass bei dem Versuch, eine Nachricht zu senden, eine Dial-Out-Verbindung hergestellt wird.

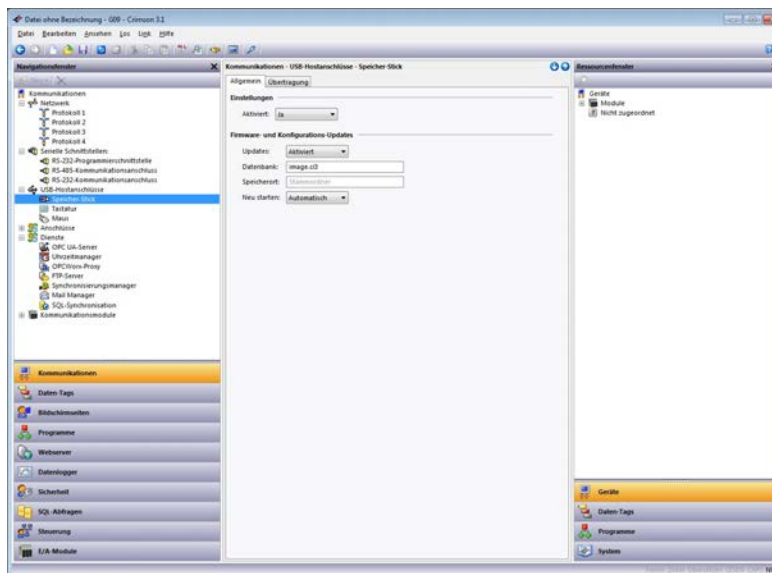
Kapitel 22 Verwenden des USB-Hosts

Wenn Ihr Zielgerät über einen oder mehrere USB-Hostanschlüsse verfügt, können Sie die unterstützten Geräte über das entsprechende Symbol in der Kategorie „Communications“ (Kommunikationen) konfigurieren. Aktuelle Builds von Crimson® 3.1 unterstützen USB-Speichergeräte, -Tastaturen und -Mäuse. Der Tastaturreiber unterstützt die vielen USB-Barcode-Lesegeräte, die Tastaturemulation bieten, und der Maustreiber unterstützt auch einige Arten von Touchscreens oder Touchpads.

22.1 Unterstützung für Speicher-Sticks

USB-Speichergeräte werden über das Symbol „Speicher-Stick“ konfiguriert:

22.1.1 Allgemeine Eigenschaften



- Mit der Eigenschaft *Enabled* (Aktiviert) wird die Unterstützung für Speicher-Sticks global aktiviert bzw. deaktiviert.
- Mit der Eigenschaft *Updates* wird die automatische Übertragung einer Update-Datei in das Stammverzeichnis der Speicherkarte des Zielgeräts konfiguriert. Wenn die Einstellung aktiviert ist, wird die angegebene Datei kopiert, wenn das Speichergerät zum ersten Mal in den USB-Anschluss eingesetzt wird. Wenn das Zielgerät neu gestartet wird, wird die Datei zur Aktualisierung der Firmware und der Konfigurationsdatenbank des Geräts verwendet.
- Die Eigenschaft *Database* (Datenbank) legt den Namen des Datenbank-Images fest, das in die Datei `image.ci3` auf der Speicherkarte kopiert werden soll. Mit dieser Einstellung können mehrere Dateien auf einem einzelnen Stick platziert werden, wobei jedes Crimson-Gerät die Datei kopiert, die für das Gerät geeignet ist, in das der Stick eingesetzt wird.
- Die Eigenschaft *Location* (Speicherort) gibt den Speicherort des oben angegebenen Datenbank-Images auf dem Speicher-Stick an.

- Mit der Eigenschaft *Restart* (Neustart) wird angegeben, ob ein automatischer Neustart durchgeführt werden soll, sobald die Datei kopiert wurde. Wenn Sie diese Eigenschaft aktivieren, können die Informationen aus dem Datenbank-Image sofort von Crimson geladen werden.

22.1.2 Übertragungseigenschaften

The screenshot shows a configuration window titled "Dateisynchronisation 1". It contains five rows of settings, each with a label and a control element:

- Übertragung:** A dropdown menu with "Dateien kopieren" selected.
- Richtung:** A dropdown menu with "Speicherkarte zu Speicher-Stick" selected.
- Update:** A dropdown menu with "Nur neue und geänderte Dateien" selected.
- Quelle:** A text input field containing "Stammordner".
- Ziel:** A text input field containing "Stammordner".

- Die Eigenschaft *Transfer* (Übertragung) für die einzelnen Synchronisierungsgruppen legt die Funktion fest, die ausgeführt werden soll. Informationen können entweder kopiert oder verschoben werden und der Vorgang kann entweder auf die Dateien im angegebenen Ordner oder zusätzlich auf rekursiver Basis auf die Unterordner und deren Inhalte angewendet werden.
- Die Eigenschaft *Direction* (Richtung) gibt die Richtung der Übertragung an.
- Mit der Eigenschaft *Update* können Sie angeben, ob Dateien, die bereits auf dem Zielgerät vorhanden sind, trotzdem kopiert werden sollen, oder ob nur neue und geänderte Dateien übertragen werden sollen. Crimson entscheidet anhand des Zeitstempels und der Größe der Datei, ob die Datei verarbeitet werden muss.
- Mit den Eigenschaften *Source* (Quelle) und *Destination* (Ziel) werden die Ordner auf den Quell- und Zielgeräten angegeben, in denen sich die Dateien befinden sollten.

22.2 Tastaturunterstützung

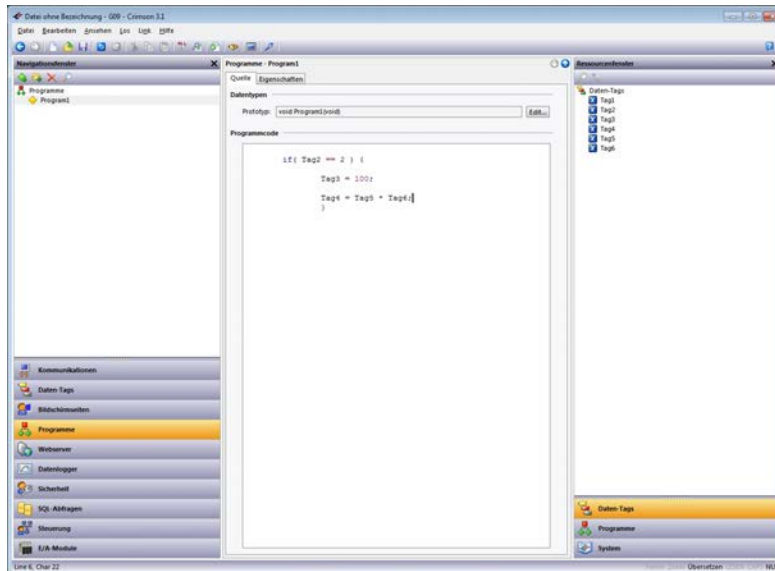
An das Zielgerät kann eine USB-Tastatur angeschlossen werden, um eine weitere Methode der Dateneingabe bereitzustellen. Crimson-Primitive akzeptieren die über die Tastatur eingegebenen Zeichen, als würden sie über das Popup-Tastenfeld eingegeben. Die Tastatur wird unter „USB Host Ports“ über den Abschnitt „Keyboard“ konfiguriert. Sie können lediglich das Tastaturlayout auswählen, damit Crimson die Scancodes in die entsprechenden Zeichen konvertieren kann. Derzeit werden nur US-amerikanische und britische Layouts unterstützt. Beachten Sie, dass USB-Barcode-Lesegeräte, die Tastaturen der Klasse HID implementieren, ebenfalls unterstützt werden. Barcodes können also mithilfe eines einfachen Dateneingabe-Primitivs direkt in Crimson-String-Tags eingegeben werden.

22.3 Mausunterstützung

An das Zielgerät kann eine USB-Maus oder ein ähnliches Zeigergerät angeschlossen werden, um eine weitere Eingabemethode bereitzustellen. Crimson-Primitive akzeptieren Klicks von der Maus, als würden sie vom Touchscreen des Geräts stammen. Diese Maus wird unter „USB Host Ports“ über den Abschnitt „Mouse“ konfiguriert. Sie können lediglich einen Skalierungsfaktor angeben, um die Maus weniger empfindlich zu machen, sowie eine Zeitspanne, nach deren Ablauf der Mauszeiger ausgeblendet wird.

Kapitel 23 Verwenden von Programmen

In den vorherigen Kapiteln dieses Handbuchs wurde erläutert, wie Sie Aktionen einsetzen, um als Reaktion auf die Betätigung von Tasten, die Berührung des Touchscreens oder Änderungen an Daten-Tags bestimmte Vorgänge durchzuführen. Wenn Sie eine Aktion ausführen müssen, die so komplex ist, dass sie nicht in eine einzelne Zeile passt, oder die eine komplexere Entscheidungslogik erfordert, können Sie mit der Kategorie „Programms“ (Programme) Programme erstellen und bearbeiten.



23.1 Die Programmliste

Die Programmliste im Navigationsfenster ist eine konventionelle Navigationsliste, die zum Erstellen, Löschen, Umbenennen und sonstigen Gestalten von Programmen verwendet werden kann. Beachten Sie, dass die Programme in Ordnern gruppiert werden können und dass jedes Programmsymbol drei Status anzeigen kann. Grün kennzeichnet ein Programm, das übersetzt und validiert wurde. Gelb kennzeichnet ein Programm, das bearbeitet, aber noch nicht übersetzt wurde. Rot kennzeichnet ein Programm, das einen oder mehrere Fehler enthält.

23.2 Ermitteln der Programmnutzung

Wenn Sie eine Liste aller Elemente anzeigen möchten, die sich auf ein bestimmtes Programm beziehen, klicken Sie im Navigationsfenster mit der rechten Maustaste auf das Element und wählen Sie den Befehl „Find Usage“ (Nutzung suchen) aus. Die gefundenen Elemente werden in der globalen Suchergebnisliste aufgeführt, und Sie können über die Taste F4 bzw. die Tastenkombination UMSCHALT+F4 darauf zugreifen. Die Liste selbst kann durch Drücken der Taste F8 ein- oder ausgeblendet werden.

23.3 Bearbeiten von Programmen

Zum Bearbeiten eines Programm bearbeiten Sie einfach den Programmtext auf der Registerkarte „Source“ (Quelle) im Bearbeitungsfenster. Sie werden feststellen, dass das Programmsymbol gelb wird, sobald Sie mit der Eingabe beginnen. Dies weist darauf hin, dass Sie Änderungen vorgenommen haben, die noch übersetzt werden müssen. Sie werden außerdem feststellen, dass der Programmeditor von Crimson® 3.1 Syntaxfärbung, automatische Einrückung und eine Reihe anderer Funktionen durchführt, die für einen Code-Editor selbstverständlich sind. Zum Konfigurieren der Editor-Optionen klicken Sie mit der rechten Maustaste auf das Bearbeitungsfenster und wählen im Kontextmenü den gewünschten Befehl aus.

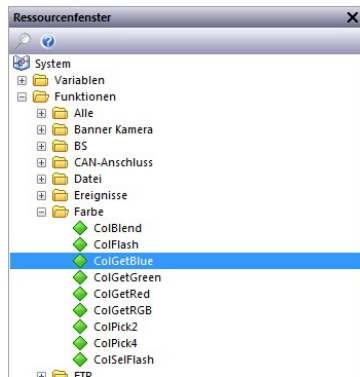
Wenn Sie mit dem Schreiben eines Programms fertig sind, drücken Sie die Tastenkombination `STRG+T` oder wählen in der Symbolleiste die Schaltfläche „Translate“ (Übersetzen) aus. Das Programm wird daraufhin auf Fehler überprüft. Wenn ein Fehler gefunden wird, wird ein Dialogfeld angezeigt, und das Programmsymbol wird rot. Außerdem wird der Cursor an die Fehlerstelle gesetzt. Wenn keine Fehler vorhanden sind, ertönt ein akustisches Signal, und das Programmsymbol wird grün. Dies bedeutet, dass das Programm in eine Form übersetzt wurde, die für die Ausführung auf dem Zielgerät geeignet ist.

23.4 Anzeigen von Hilfetext

Bei der Arbeit im Bearbeitungsfenster können Sie über ein Tastaturkürzel Hilfetext zu Systemfunktionen anzeigen. Setzen Sie den Cursor in einen Funktionsnamen oder an dessen Ende und drücken Sie die Taste `F1`. Daraufhin werden Informationen über die Arbeitsweise, die Argumente und den Rückgabebetyp der Funktion angezeigt. Die gleichen Informationen werden angezeigt, wenn Sie den Namen einer Funktion eingeben und dann `F1` drücken.

23.5 Das Ressourcenfenster

Das vom Programmierer angezeigte Ressourcenfenster enthält eine Reihe von Elementen, die in den Code gezogen werden können. Die Kategorien „Data Tags“ (Daten-Tags) und „Programs“ (Programme) bedürfen keiner weiteren Erklärung. Sie bieten einen schnellen Zugriff auf die jeweiligen Elemente in der Datenbank, indem sie es ermöglichen, den Namen eines Elements in den Editor einzufügen. Über die Kategorie „System“ haben Sie Zugriff auf die umfangreiche Crimson-Bibliothek von Systemvariablen und Funktionen:



Wie Sie sehen, sind Variablen und Funktionen in Kategorien unterteilt. Wenn eine Funktion ausgewählt wird, werden der Rückgabebetyp und die Argumenttypen in der Statusleiste angezeigt. Wenn Sie eine Funktion in Ihrem Code ablegen, wird der entsprechende Text eingegeben, und der Cursor wird in die Klammern nach dem Funktionsnamen gesetzt, damit Sie die erforderlichen Argumente eingeben können.

23.6 Programmdatentypen

Das Feld oberhalb des Programmeditors kann zum Bearbeiten der Datentypen des Programms verwendet werden:

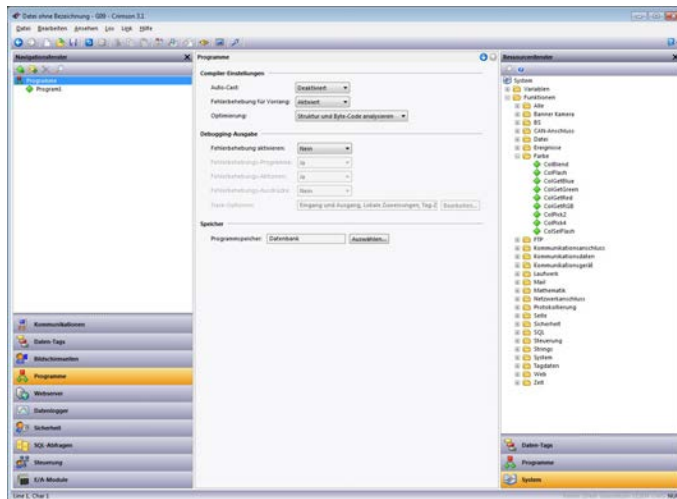
Parameter	
Typ	Name
1: Keine	Param1
2: Keine	Param2
3: Keine	Param3
4: Keine	Param4
5: Keine	Param5
6: Keine	Param6

- Mit der Eigenschaft *Data Type* (Datentyp) wird angegeben, ob dieses Programm einfach eine Reihe von Aktionen ausführen soll oder ob es eine Berechnung durchführen und den Wert dieser Berechnung an den Aufrufer zurückgeben soll. Programme, die Werte zurückgeben, können definitionsgemäß nicht im Hintergrund ausgeführt werden.
- Im Eigenschaftsbereich *Parameters* (Parameter) werden bis zu sechs Parameter definiert, die das Programm akzeptiert. Jeder Parameter hat einen Namen und einen Datentyp. In diesem Beispiel akzeptiert das Programm die beiden Parameter `Value1` und `Value2`. Beide Parameter sind vorzeichenbehaftete 32-Bit-Ganzzahlen.

Die Rückgabe von Werten und die Übergabe von Parametern werden noch ausführlicher erörtert.

23.7 Globale Einstellungen

Indem Sie in der Navigationsliste das Stammelement auswählen, erhalten Sie Zugriff auf die globalen Programmeinstellungen...



- Die Eigenschaft *Compiler Settings* (Compiler-Einstellungen) wird verwendet, um die Kompatibilität mit Code, der aus deutlich früheren Versionen von Crimson importiert wurde, sicherzustellen. Diese Einstellungen sollten Sie nur dann ändern, wenn Sie vom Red Lion Technical Support dazu aufgefordert werden.
- Die Eigenschaft *Enable Debug* (Debug aktivieren) wird verwendet, um die Ausgabe von Ausführungs-Tracedaten an die verschiedenen Debug-Konsolen des Systems global zu aktivieren bzw. zu deaktivieren. Die Trace-Ausgabe gibt eine Meldung an die Konsole aus, wenn eine bestimmte Aktion von einem bestimmten Codefragment ausgeführt wird. Informationen zur Debug-Konsole finden Sie im Kapitel „Erweitertes Debugging“.
- Die Eigenschaft *Debug Programs* (Programme debuggen) wird verwendet, um die Trace-Ausgabe aus der Ausführung von Programmen global zu aktivieren oder zu deaktivieren. Jedes Programm verfügt über eine ähnliche Eigenschaft, mit der diese Einstellung außer Kraft gesetzt werden kann. Dadurch lässt sich genauer steuern, welche Trace-Informationen ausgegeben werden.
- Die Eigenschaft *Debug Actions* (Aktionen debuggen) wird verwendet, um die Trace-Ausgabe aus der Ausführung von Aktionen global zu aktivieren oder zu deaktivieren. Diese Aktionen können durch Benutzeraktivitäten wie das Drücken einer Taste oder durch Systemaktivitäten ausgelöst werden, z. B. durch einen versehentlich gesetzten Haken oder die Aktivierung eines Auslösers, der einem Tag zugeordnet ist.
- Die Eigenschaft *Debug Expressions* (Ausdrücke debuggen) wird verwendet, um die Trace-Ausgabe aus der Evaluierung von Ausdrücken global zu aktivieren oder zu deaktivieren. Da Crimson Ausdrücke für allerlei Vorgänge verwendet, von der Berechnung der anzuzeigenden Werte bis hin zu der Entscheidung, welche Farben für Anzeige-Prims verwendet werden sollen, erzeugt die Aktivierung dieser Option eine Ausgabe mit einem großen Volumen.
- Die Eigenschaft *Trace Options* (Trace-Optionen) wird verwendet, um die Ereignisse global auszuwählen, die in der Debug-Konsole protokolliert werden. Zu den Optionen gehören das Öffnen oder Verlassen von Programmen, Zuweisungen zu lokalen Variablen oder Tags, Aufrufe an Systemfunktionen, die Aktionen ausführen, Aufrufe an Systemfunktionen, die Werte berechnen, oder Aufrufe an Programme.
- Mit der Eigenschaft *Program Storage* (Programmspeicher) wird angegeben, ob Programme wie gewohnt in der Crimson-Datenbankdatei oder in externen Dateien im selben Verzeichnis wie die Datenbank gespeichert werden sollen. Der externe Speicher ist eine erweiterte Option, mit der ein

externes Quellenkontrollsystem Programmänderungen vornehmen und Änderungen von verschiedenen Benutzern zusammenführen kann. Crimson hält interne und externe Bearbeitungen so weit wie möglich synchron; dennoch sollten Sie nicht versuchen, dasselbe Programm intern und extern gleichzeitig zu bearbeiten.

23.8 Programmeigenschaften

Auf der zweiten Registerkarte des Editors wird die Ausführungsumgebung des Programms festgelegt.



- Mit der Eigenschaft *Execution Task* (Ausführungsaufgabe) wird angegeben, wie Crimson das Programm ausführen soll. Wenn die Eigenschaft auf „Same as Caller“ (Wie Aufrufer) eingestellt ist, wird das Programm von der Aufgabe ausgeführt, die es aufgerufen hat, und Crimson wartet, bis die Ausführung des Programms abgeschlossen ist, bevor es mit seiner eigentlichen Aufgabe fortfährt. Wenn Sie z. B. diese Einstellung auswählen und ein Programm als Reaktion auf die Betätigung einer Taste ausführen, werden Anzeigeaktualisierungen unterbrochen, bis das Programm abgeschlossen ist. (Dies fällt möglicherweise kaum auf, da die meisten Programme sehr schnell ausgeführt werden.) Wenn diese Eigenschaft auf eine der Hintergrundeinstellungen gesetzt ist, verwendet Crimson eine von drei Aufgaben, die für die Ausführung im Hintergrund zugewiesen wurden, und die aufrufende Aufgabe kann sofort ihre eigene Arbeit fortführen. Jede Aufgabe kann immer nur jeweils ein Hintergrundprogramm ausführen. Nachfolgende Anforderungen werden daher zur späteren Ausführung in die Warteschlange gestellt. Beachten Sie auch, dass Programme, die Werte zurückgeben, nicht im Hintergrund ausgeführt werden können, da ihr Rückgabewert sonst nicht für den Anrufer verfügbar wäre. Programme mit der Einstellung „Same as Caller“ (Wie Aufrufer), die von im Hintergrund ausgeführten Programmen aufgerufen werden, werden selbst im Hintergrund ausgeführt und unterbrechen das zuvor ausgeführte Programm. Programme, die für die Ausführung im Hintergrund eingestellt sind, die wiederum von Programmen aufgerufen werden, die bereits im Hintergrund ausgeführt werden, werden in die Warteschlange gestellt, wenn sie an die gleiche Aufgabe oder eine bereits tätige Aufgabe geleitet werden. Sie werden sofort ausgeführt, wenn sie an eine ungenutzte Aufgabe geleitet werden.
- Mit den Eigenschaften *External Data* (Externe Daten) und *Timeout* (Timeout) wird gesteuert, wie das Programm in Bezug auf externe Datenelemente, auf die sich das Programm bezieht, mit der Kommunikationsinfrastruktur von Crimson interagiert. Wie bereits erwähnt, liest Crimson Datenelemente nur, wenn sie verwendet werden. Mit dieser Eigenschaft wird die genaue Interpretation dieser Regel in Bezug auf Programme gesteuert:

Modus	Verhalten
Lesen wenn Bezug	Externe Daten, die vom Programm verwendet werden, werden der Kommunikationsprüfung hinzugefügt, wenn auf das Programm verwiesen wird. Wenn eine Anzeigeseite auf das Programm verweist, werden die Daten gelesen, sobald diese Seite angezeigt wird. Wenn eine globale Aktion oder ein Auslöser auf das Programm verweist, werden die Daten immer gelesen. Dies ist der Standardmodus. Er eignet sich für alle Programme mit Ausnahme von denen, die sehr große Mengen externer Daten verwenden.
Immer lesen	Externe Daten, die vom Programm verwendet werden, werden immer gelesen, unabhängig davon, ob auf das Programm verwiesen wird. Dies bedeutet, dass das Programm stets zur Ausführung bereit ist und dass die Meldung „NOT READY“ (Nicht bereit) nicht angezeigt wird, wie es sonst eventuell der Fall ist, sobald erstmalig auf das Programm verwiesen wird. Der Nachteil dieses Modus ist, dass die Kommunikationsleistung beeinträchtigt werden kann, wenn das Programm auf große Datenmengen verweist.
Lesen bei Ausführung	Externe Daten, die innerhalb des Programms verwendet werden, werden nur gelesen, wenn das Programm aufgerufen wird. Das Programm wartet bis zum Ende des in der Eigenschaft „Timeout“ angegebenen Zeitraums darauf, dass diese Daten verfügbar werden. Wenn die Daten nicht gelesen werden können, etwa weil ein Gerät offline ist, wird das Programm nicht ausgeführt. Dieser Modus wird in der Regel bei global referenzierten Programmen verwendet, die große Datenmengen verbrauchen, die andernfalls die Kommunikationsprüfung verlangsamen würden.
Lesen, aber dennoch ausführen	Externe Daten werden behandelt wie unter „Immer lesen“ beschrieben, doch das Programm wird unabhängig davon ausgeführt, ob die Daten erfolgreich gelesen wurden. Der Bediener sieht daher nie die Meldung „NOT READY“ (Nicht bereit). Wenn ein Gerät jedoch offline ist, gibt es keine Garantie dafür, dass die Datenelemente des Programms gültige Daten enthalten.

- Die Eigenschaft *Enable Debug* (Debugging aktivieren) wird verwendet, um die Trace-Ausgabe aus der Ausführung dieses Programms lokal zu aktivieren bzw. zu deaktivieren. Wenn Sie einen anderen Wert als die Standardeinstellung auswählen, wird die globale Eigenschaft *Debug Programs* (Programme debuggen) überschrieben.
- Die Eigenschaft *Trace Options* (Trace-Optionen) wird verwendet, um die Ereignisse auszuwählen, die in der Debug-Konsole für dieses Programm protokolliert werden. Zu den Optionen gehören das Öffnen oder Verlassen von Programmen, Zuweisungen zu lokalen Variablen oder Tags, Aufrufe an Systemfunktionen, die Aktionen ausführen, Aufrufe an Systemfunktionen, die Werte berechnen, oder Aufrufe an Programme.

23.9 Hinzufügen von Kommentaren

Es gibt zwei Möglichkeiten, Ihren Programmen Kommentare hinzuzufügen. Erstens können Sie mit den Zeichen `//` einen Kommentar einleiten, der bis zum Ende der aktuellen Zeile reicht. Zweitens können Sie mit den Zeichen `/*` einen Kommentar einleiten, der über eine oder mehrere Zeilen reicht. Das Ende des Kommentars wird mit den Zeichen `*/` signalisiert. In der folgenden Abbildung sehen Sie Beispiele für beide Kommentarstile:

```
// Dies ist ein einzeiliger Kommentar

/* Dies ist Zeile 1 des Kommentars
   Dies ist Zeile 2 des Kommentars
   Dies ist Zeile 3 des Kommentars */
```

Ein einzeiliger Kommentar kann auch am Ende einer Zeile eingefügt werden, die Code enthält.

23.10 Zurückgeben von Werten

Wie oben erwähnt, können Programme Werte zurückgeben. Derartige Programme können von anderen Programmen oder von Ausdrücken in der Datenbank aufgerufen werden. Wenn Sie beispielsweise eine besonders komplexe Decodierung für eine Reihe von Bedingungen in Bezug auf einen Motor durchführen und einen Wert zurückgeben möchten, der den aktuellen Status angibt, könnten Sie ein Programm erstellen, das wie folgt eine Ganzzahl zurückgibt:

```
if( MotorRunning )
return 1;
else {
if( MotorTooHot )
return 2;
if( MotorTooCold )
return 3;
return 0;
}
```

Sie könnten dann ein Tag konfigurieren, um dieses Programm aufzurufen, und mit einem Mehrfachstatus-Format Namen für die verschiedenen Zustände bereitstellen. Der Aufruf wird ausgeführt, indem Sie die Eigenschaft „Value“ (Wert) des Tags auf `Program()` setzen, wobei `Program` der Name des betreffenden Programms ist. Die Klammern dienen zum Angeben eines Funktionsaufrufs und können nicht weggelassen werden.

23.10.1 Vermeiden von potenziellen Problemen

Beachten Sie, dass Sie bei Programmen, die Werte zurückgeben, ein gewisses Maß an Vorsicht walten lassen müssen. Insbesondere sollten Sie länger andauernde Schleifen oder die Ausführung von Aktionen vermeiden, die unter den Umständen, unter denen die Funktion aufgerufen wird, nicht sinnvoll sind. Angenommen, das obige Codefragment würde die Funktion `GotoPage` aufrufen, um die Seite zu ändern. In diesem Fall würde sich die Anzeige jedes Mal ändern, wenn das Programm aufgerufen wird. Stellen Sie sich nun einmal vor, dies würde passieren, wenn Sie versuchen, Daten aus dem zugehörigen Tag zu protokollieren. Sie werden feststellen, dass dies keine gute Idee ist. Halten Sie daher Programme, die Werte zurückgeben, möglichst einfach, und berücksichtigen Sie stets den Kontext, in dem sie ausgeführt werden. Im Zweifelsfall sollten Sie sich auf einfache Berechnungen und `if`-Anweisungen beschränken und alles andere vermeiden.

23.11 Übergeben von Argumenten

Wie oben erwähnt, können Programme Argumente annehmen. Angenommen, Sie möchten ein Programm namens `FindMean` schreiben, um den Durchschnitt zweier Ganzzahlwerte zu ermitteln. Das Programm würde so konfiguriert, dass es die beiden Ganzzahlargumente `a` und `b` annimmt und eine Ganzzahl zurückgibt. Der Code innerhalb des Programms würde wie folgt definiert:

```
return (a+b)/2;
```

Nachdem dieses Programm erstellt und übersetzt wurde, können Sie einen Ausdruck wie `FindMean(Tag1, Tag2)` eingeben, um es mit den entsprechenden Argumenten aufzurufen. In diesem Fall würde der Ausdruck dem Durchschnitt von `Tag1` und `Tag2` entsprechen.

23.12 Tipps für das Programmieren

In den folgenden Abschnitten finden Sie eine Übersicht über die von Crimson unterstützten Programmierkonstrukte. Die verwendete grundlegende Syntax entspricht der Programmiersprache C. Beachten Sie, dass es hier nicht darum geht, Sie zu einem Programmierer auszubilden oder Ihnen die Feinheiten der Sprache C näher zu bringen. Solche Themen würden den Rahmen dieses Handbuchs sprengen. Vielmehr geht es darum, einen schnellen Überblick über die verfügbaren Merkmale zu geben, sodass interessierte Benutzer nach Wunsch weiter experimentieren können.

23.12.1 Mehrere Aktionen

Die einfachste Art von Programm besteht aus einer Liste von Aktionen, wobei jede Aktion eine einzelne Zeile belegt und mit einem Semikolon beendet wird. Alle im Abschnitt „Schreiben von Aktionen“ erörterten Aktionen können verwendet werden. Einfache Programme wie diese werden in der Regel verwendet, wenn sich die Zusammenfassung der Aktionen in einer einzigen Aktionsdefinition als unleserlich erweisen würde.

Im nachstehenden Beispiel werden zuerst mehrere Variablen festgelegt und dann wird zu einer anderen Anzeigeseite gewechselt:

```
Motor1 = 0;  
Motor2 = 1;  
Motor3 = 0;
```

```
GotoPage(Page1);
```

Die Aktionen werden nacheinander ausgeführt, anschließend kehrt das Programm zum Aufrufer zurück.

23.12.2 If-Anweisungen

Diese Art von Anweisung wird in einem Programm verwendet, um eine Entscheidung zu treffen. Das Konstrukt besteht aus einer `if`-Anweisung mit einer Bedingung in Klammern, gefolgt von einer Aktion (oder Aktionen), die ausgeführt werden soll, wenn die Bedingung „true“ ist. Werden mehrere Aktionen festgelegt, muss jede Aktion in einer separaten Zeile platziert werden, und die Anweisungen müssen mithilfe geschweifeter Klammern gruppiert werden. Mit einer optionalen `else`-Klausel kann Code bereitgestellt werden, der ausgeführt werden soll, wenn die Bedingung „false“ ist.

Im folgenden Beispiel sehen Sie eine `if`-Anweisung mit einer einzigen Aktion:

```
if( TankFull )  
StartPump = 1;
```

Im folgenden Beispiel sehen Sie eine `if`-Anweisung mit zwei Aktionen:

```
if( TankEmpty ) {  
StartPump = 0;
```

```
OpenValue = 1;  
}
```

Im folgenden Beispiel sehen Sie eine `if`-Anweisung mit einer `else`-Klausel:

```
if( MotorHot )  
StartFan = 1;  
else  
StartFan = 0;
```

Beachten Sie, dass es sehr wichtig ist, Gruppen von Aktionen, die im `if`- oder `else`-Teil der Anweisung ausgeführt werden sollen, in geschweifte Klammern zu setzen. Wenn Sie die Klammern weglassen, versteht Crimson wahrscheinlich nicht, welche Aktionen von der `if`-Bedingung abhängig sein sollen. Obwohl Zeilenumbrüche zwischen Aktionen empfohlen werden, werden sie nicht dazu verwendet, herauszufinden, was in der bedingten Anweisung enthalten ist.

23.12.3 Switch-Anweisungen

Bei einer `switch`-Anweisung wird ein Ganzzahlwert mit einer Reihe möglicher Konstanten verglichen und auf Basis des übereinstimmenden Werts eine Aktion durchgeführt. Die genaue Syntax unterstützt mehr Optionen als im unten stehenden Beispiel gezeigt, doch für die große Mehrheit der Anwendungen ist diese einfache Form akzeptabel.

Im unten gezeigten Beispiel wird ein Motor gestartet, der anhand des Werts im Tag `MotorIndex` ausgewählt wird:

```
switch( MotorIndex ) {  
  
case 1:  
MotorA = 1;  
break;  
case 2:  
case 3:  
MotorB = 1;  
break;  
case 4:  
MotorC = 1;  
break;  
default:  
MotorD = 1;  
break;  
}
```

Beim Wert 1 wird Motor A gestartet, beim Wert 2 oder 3 Motor B und beim Wert 4 Motor C. Bei allen nicht ausdrücklich genannten Werten wird Motor D gestartet. Erwähnenswert sind bei dieser Syntax die geschweiften Klammern um die `case`-Anweisungen, die Verwendung von `break` zum Beenden der einzelnen Bedingungsblöcke, die Verwendung zweier aufeinander folgender `case`-Anweisungen für die Übereinstimmung mit mehr als einem Wert und die Verwendung der optionalen `default`-Anweisung, um eine Aktion anzugeben, die ausgeführt werden soll, wenn keiner der angegebenen Werte mit dem Wert im Steuerausdruck übereinstimmt. (Falls diese Syntax

abschreckend wirkt, kann stattdessen auch eine Reihe von `if`-Anweisungen verwendet werden. Damit lässt sich zwar das gleiche Ergebnis erzielen, doch die Leistung und die Lesbarkeit werden geringfügig herabgesetzt.)

23.12.4 Lokale Variablen

Einige Programme verwenden zum Speichern von Zwischenergebnissen oder zum Steuern eines der nachstehend beschriebenen Schleifenkonstrukte Variablen. Anstatt ein Tag zu definieren, das diese Werte aufnimmt, können Sie sogenannte lokale Variablen deklarieren und dabei folgende Syntax verwenden:

```
int    a;// Declare local integer 'a'
float  b;// Declare local real   'b'
cstring c;// Declare local string 'c'
```

Lokale Variablen können optional initialisiert werden, wenn sie deklariert werden, indem Sie `=` und den zuzuweisenden Wert an den Variablennamen anhängen. Variablen, die nicht auf diese Weise initialisiert werden, werden entweder auf Null oder eine leere Zeichenfolge gesetzt.

Beachten Sie, dass lokale Variablen sowohl im Geltungsbereich als auch in der Lebensdauer wirklich „lokal“ sind. Dies bedeutet, dass sie nicht außerhalb des Programms referenziert werden können und ihre Werte nicht zwischen Funktionsaufrufen beibehalten. Wenn eine Funktion rekursiv aufgerufen wird, hat jeder Aufruf seine eigenen Variablen.

23.12.5 Schleifenkonstrukte

Die drei verschiedenen Schleifenkonstrukte können dazu verwendet werden, einen bestimmten Teil des Codes auszuführen, während eine bestimmte Bedingung „true“ ist. Die `while`-Schleife prüft ihre Bedingung, bevor der Code ausgeführt wird. Die `do`-Schleife hingegen prüft die Bedingung hinterher. Die `for`-Schleife ist eine schnellere Methode, eine `while`-Schleife zu definieren, da sie es ermöglicht, drei gängige Elemente in einer Anweisung zusammenzufassen.

Beachten Sie, dass bei der Verwendung von Schleifen in Programmen Vorsicht geboten ist. Wenn Sie einen Programmierfehler machen, könnte eine niemals endende Schleife entstehen. Je nach der Situation, in der das Programm aufgerufen wird, könnte dies die Benutzeroberflächenaktivität oder die Kommunikation des Terminals erheblich beeinträchtigen. Auch Schleifen mit zu vielen Wiederholungen können für das Untersystem, das sie aufruft, Leistungsprobleme verursachen.

23.12.5.1 Die While-Schleife

Diese Art von Schleife wiederholt die Aktion, die auf sie folgt, solange die Bedingung in der `while`-Anweisung „true“ lautet. Wenn die Bedingung nie „true“ ist, wird die Aktion nie ausgeführt, und die Schleife führt abgesehen von der Auswertung der steuernden Bedingung keinerlei Vorgänge aus. Wenn Sie mehrere Aktionen in die Schleife einfügen möchten, müssen Sie die entsprechenden Anweisungen wie bei der `if`-Anweisung in geschweifte Klammern setzen. Das folgende Beispiel initialisiert ein Paar lokaler Variablen. Dann wird die erste Variable dazu verwendet, eine Schleife durch die Inhalte eines Arrays auszuführen, die ersten zehn Elemente zusammenzuzählen und den Gesamtwert an den Aufrufer zurückzugeben:

```
int i=0, t=0;

while( i < 10 ) {
    t = t + Data[i];
    i = i + 1;
}
```

```
}  
  
return t;
```

Das folgende Beispiel zeigt das gleiche Programm, das jedoch in einer komprimierten Form neu geschrieben wurde. Da die loop-Anweisung jetzt nur eine einzelne Aktion steuert, wurden die geschweiften Klammern weggelassen:

```
int i=0, t=0;  
  
while( i < 10 )  
t += Data[i++];  
  
return t;
```

23.12.5.2 Die For-Schleife

Wie Sie sehen, umfasst die oben gezeigte `while`-Schleife vier Elemente:

1. Die Initialisierung der Schleifensteuervariablen
2. Die Bewertung einer Prüfung, um festzustellen, ob die Schleife fortgesetzt werden muss
3. Die Ausführung der Aktion, die von der Schleife durchgeführt werden soll
4. Die Durchführung einer Änderung an der Steuervariablen

Mit der `for`-Schleife können die Elemente 1, 2 und 4 in einer einzelnen Anweisung vereint werden, sodass die Aktion, die auf die Anweisung folgt, nur Element 3 implementieren muss. Diese Syntax führt zu einem Konstrukt, das der FOR-NEXT-Schleife in BASIC und anderen derartigen Sprachen ähnelt.

Mit dieser Anweisung kann das oben gezeigte Beispiel wie folgt umgeschrieben werden:

```
int i, t;  
  
for( i=t=0; i<10; i++ )  
t += Data[i];  
  
return t;
```

Wie Sie sehen, enthält die `for`-Anweisung drei separate Elemente, die jeweils durch ein Semikolon voneinander getrennt sind. Das erste Element ist der Initialisierungsschritt, der einmal ausgeführt wird, wenn die Schleife beginnt. Das nächste Element ist die Bedingung, die am Anfang jeder Schleifeniteration geprüft wird, um festzustellen, ob die Schleife fortgeführt werden muss. Das dritte und letzte Element ist der Induktionsschritt, mit dem Änderungen an der Steuervariablen vorgenommen werden und die Schleife zur nächsten Iteration weitergeschoben wird. Auch hier gilt, dass Sie der Schleife mehr als eine Aktion hinzufügen können, diese Aktionen jedoch in geschweifte Klammern setzen müssen.

23.12.5.3 Die Do-Schleife

Dieser Schleifentyp ähnelt der `while`-Schleife mit der Ausnahme, dass die Bedingung am Ende der Schleife geprüft wird. Dies bedeutet, dass die Schleife immer mindestens einmal ausgeführt wird.

Das folgende Beispiel zeigt das Beispiel von oben, diesmal mit einer `do`-Schleife:

```
int i=0, t=0;

do {
  t += Data[i];
} while( ++i < 10 );

return t;
```

23.12.5.4 Schleifensteuerung

In Schleifen können zwei weitere Anweisungen verwendet werden. Mit der `break`-Anweisung kann die Schleife vorzeitig beendet werden. Mit der `continue`-Anweisung kann der Rest der Schleife übersprungen und eine weitere Iteration begonnen werden, ohne weiteren Code auszuführen. Damit dies Sinn ergibt, müssen diese Anweisungen mit `if`-Anweisungen verwendet werden, um eine bedingte Ausführung zu erzielen.

Das folgende Beispiel zeigt eine Schleife, die vorzeitig beendet wird, wenn ein anderes Programm „true“ zurückgibt:

```
for( i=0; i<10; i++ ) {
  if( LoopAbort() )
    break;
  LoopBody();
}
```

Kapitel 24 Schreiben von Ausdrücken

Wie bereits an anderer Stelle in diesem Handbuch erwähnt, sind viele Felder in Crimson® 3.1 als sogenannte Ausdruckseigenschaften konfiguriert. Für die Bearbeitung diese Felder steht, wie Sie vielleicht noch wissen, ein Benutzeroberflächenelement bereit, das ungefähr wie in der folgenden Abbildung aussieht:



In vielen Situationen werden Sie diese Eigenschaften so konfigurieren, dass sie mit dem Wert eines Tags oder dem Inhalt eines Registers in einem externen Kommunikationsgerät übereinstimmen. In diesen Fällen ziehen Sie entweder Elemente aus dem Ressourcenfenster oder Sie klicken im Dropdown-Menü auf die gewünschte Option und wählen im daraufhin angezeigten Dialogfeld das gewünschte Element aus.

Es wird jedoch auch Situationen geben, in denen Sie eine Eigenschaft von einer komplexeren Kombination von Datenelementen abhängig machen und deren Werte vielleicht mithilfe mathematischer Berechnungen verbinden oder vergleichen möchten. Für derartige Zwecke sind die sogenannten Ausdrücke vorgesehen, die in das Bearbeitungsfeld der Eigenschaft eingegeben werden können, wann immer im Dropdown-Menü der Modus „General“ (Allgemein) ausgewählt ist.

24.1 Datenwerte

Alle Ausdrücke enthalten mindestens einen Datenwert. Die einfachsten Ausdrücke sind Verweise auf einzelne Konstanten, einzelne Tags oder einzelne SPS-Register. Wenn Sie eine der letzten beiden Optionen eingeben, vereinfacht Crimson den Bearbeitungsvorgang, indem es den Eigenschaftsmodus automatisch in der erforderlichen Weise ändert. Wenn Sie beispielsweise im Modus „General“ (Allgemein) einen Tag-Namen eingeben, wechselt Crimson in den Tag-Modus und zeigt den Tag-Namen im Auswahlfeld an.

24.1.1 Konstanten

Konstanten stellen konstante Zahlen oder Zeichenfolgen dar.

24.1.1.1 Integerkonstanten

Integerkonstanten stellen eine einzelne vorzeichenbehaftete 32-Bit-Zahl dar. Sie können bei Bedarf im Dezimal-, Binär-, Oktal- oder Hexadezimalformat eingegeben werden. Die folgenden Beispiele zeigen, wie dieselbe Zahl in den vier verschiedenen Zahlenbasen aussieht:

Basis	Beispiel
Dezimal	123
Binär	0b1111011
Oktal	0173
Hexadezimal	0x7B

Die von früheren Versionen der Software unterstützten Suffixe „U“ und „L“ werden nicht verwendet.

24.1.1.2 Zeichenkonstanten

Zeichenkonstanten stellen ein einzelnes Unicode-Zeichen dar, das in den unteren 16 Bits einer vorzeichenbehafteten 32-Bit-Zahl codiert ist. Eine Zeichenkonstante besteht aus einem einzelnen Zeichen in einfachen Anführungszeichen, sodass 'A' zum Darstellen des Werts von 65 verwendet werden kann. Bestimmte ansonsten nicht druckbare oder nicht darstellbare Zeichen können mit sogenannten Escapesequenzen codiert werden, die jeweils mit einem einzelnen umgekehrten Schrägstrich eingeleitet werden:

Sequenz	Wert	ASCII
\a	Hex 0x07, Dezimal 7	BEL
\t	Hex 0x09, Dezimal 9	TAB
\n	Hex 0x0A, Dezimal 10	LF
\f	Hex 0x0C, Dezimal 12	FF
\r	Hex 0x0D, Dezimal 13	CR
\e	Hex 0x1B, Dezimal 27	ESC
\xnn	Der durch nn dargestellte Hexadezimalwert	-
\unnnn	Der durch nnnn dargestellte Hexadezimalwert	-
\nnn	Der durch nnn dargestellte Hexadezimalwert	-
\\	Ein einzelner umgekehrter Schrägstrich	-
\'	Ein einzelnes Anführungszeichen	-
\"	Ein doppeltes Anführungszeichen	-

24.1.1.3 Logische Konstanten

Logische Konstanten stellen den Wert 1 oder 0 dar, der die Wahrheit oder etwas Anderes eines Ja-oder-Nein-Ausdrucks angibt. Ein Beispiel für etwas, das einer logischen Konstante gleichgesetzt werden kann, ist ein Tag, das einen digitalen Ausgang in einer SPS darstellt. Logische Konstanten können entweder einfach als 1 oder 0 oder mit den Schlüsselwörtern `true` oder `false` eingegeben werden.

24.1.1.4 Gleitkommakonstanten

Gleitkommakonstanten stellen einen 32-Bit-Gleitkommawert mit einfacher Genauigkeit von dar. Sie bestehen aus einem Ganzzahlanteil, gefolgt von einem einzelnen Dezimalpunkt und dem Bruchanteil. Auch die wissenschaftliche Schreibweise wird unterstützt. Dazu wird ein Wert für die Mantisse angegeben, gefolgt von einem „E“ und einem Exponenten.

24.1.1.5 String-Konstanten

String-Konstanten stellen Buchstabenfolgen dar. Sie bestehen aus den darzustellenden Buchstaben, die zwischen doppelten Anführungszeichen stehen. Der String "ABCD" z. B. stellt eine vierstellige Zeichenfolge dar, die die Werte 65, 66, 67 und 68 umfasst. (Eigentlich werden fünf 16-Bit-Wörter verwendet, um die Zeichenfolge zu speichern, d. h. als Endezeichen wird ein Nullwert angehängt.) Die verschiedenen oben erwähnten Escapesequenzen können ebenfalls innerhalb von Zeichenfolgen verwendet werden.

24.1.2 Tag-Werte

Der Wert eines Tags wird in einem Ausdruck durch den Tag-Namen dargestellt. Tags, die in Ordnern organisiert sind, werden durch den Pfadnamen des Tags dargestellt, wobei jedes Elementpaar durch einen Punkt getrennt wird. Der Verweis auf ein Tag namens „PV“ in einem Ordner namens „Loop“ würde daher `Loop.PV` lauten.

Beachten Sie, dass Groß- und Kleinbuchstaben bei der Suche nach einem Tag als gleichwertig betrachtet werden. Sobald ein Ausdruck eingegeben wurde, werden bei einer Änderung des Tag-Namens auch alle Ausdrücke geändert, die darauf verweisen.

24.1.3 Tag-Eigenschaften

Daten-Tags haben bestimmte Eigenschaften. Für den Zugriff auf eine Eigenschaft geben Sie den Tag-Namen, einen Punkt und den Namen der gewünschten Eigenschaft ein. Folgende Eigenschaften sind definiert:

Eigenschaft	Beschreibung	Datentyp
Name	Name des Tags	String
AsText	Der als Text formatierte Wert des Tags	String
Bezeichnung	Die Eigenschaft „Label“ (Bezeichnung) des Tags	String
Desc	Die Eigenschaft „Description“ (Beschreibung) des Tags	String
Vorzeichen	Das vom Tag-Format definierte Vorzeichen	String
Einheiten	Die vom Tag-Format definierten Einheiten	String
SP	Die Eigenschaft „Setpoint“ (Sollwert) des Tags	Wie Tag
Min.	Der untere Dateneingabegrenzwert des Tags	Wie Tag
Max.	Der obere Dateneingabegrenzwert des Tags	Wie Tag
Fore	Die aktuelle Vordergrundfarbe des Tags	Integer
Zurück	Die aktuelle Hintergrundfarbe des Tags	Integer

24.1.4 Seiteneigenschaften

Auch Anzeigeseiten haben bestimmte Eigenschaften, auf die auf die gleiche Weise zugegriffen werden kann:

Eigenschaft	Beschreibung	Datentyp
Name	Name der Seite	String
Bezeichnung	Die Eigenschaft „Label“ (Bezeichnung) der Seite	String
Desc	Die Eigenschaft „Description“ (Beschreibung) der Seite	String

24.1.5 Kommunikationsverweise

Verweise auf Register in Master-Kommunikationsgeräten können in einen Ausdruck mithilfe einer Syntax eingegeben werden, die aus einer öffnenden eckigen Klammer, dem Registernamen und einer schließenden eckigen Klammer besteht. Dem Registernamen kann ein optionaler Gerätenamen vorangestellt und durch einen Punkt getrennt werden. Der Gerätenamen ist nicht erforderlich, wenn auf das einzige Gerät in einer Datenbank verwiesen wird.

Beispiele für diese Syntax sind im Folgenden zu sehen:

Beispiel	Bedeutung
[D100]	Register D100 im ersten Gerät
[AB.N7:0]	Register N7 : 0 im Gerät AB
[FX.D100]	Register D100 im Gerät FX.

24.2 Einfache Berechnungen

Wie bereits erwähnt enthalten Ausdrücke oft mehr als einen Datenwert, wobei deren Werte mathematisch verbunden werden. Die einfachsten dieser Ausdrücke können zwei Werte addieren und komplexere Ausdrücke könnten den Durchschnitt von drei Werten ermitteln. Diese Vorgänge werden mit der vertrauten Syntax durchgeführt, die Sie aus Anwendungen wie Excel kennen. Die unten gezeigten Beispiele zeigen die Grundrechenarten, die durchgeführt werden können:

Bediener	Priorität	Beispiel
Addition	Gruppe 4	Tag1 + Tag2
Subtraktion	Gruppe 4	Tag1 - Tag2
Multiplikation	Gruppe 3	Tag1 * Tag2
Division	Gruppe 3	Tag1 / Tag2
Rest	Gruppe 3	Tag1 % Tag2

In den Beispielen stehen zwar Leerzeichen vor und nach den Operatoren, diese sind jedoch nicht erforderlich.

24.3 Operator-Priorität

In der oben gezeigten Tabelle ist Ihnen wahrscheinlich die Spalte „Priorität“ aufgefallen. Wie Sie sicher noch aus dem Mathematikunterricht wissen, werden die Operatoren bei mathematischen Ausdrücken mit gemischten Operatoren in einer bestimmten Reihenfolge ausgewertet. Beispielsweise hat die Multiplikation immer Vorrang vor der Addition („Punkt vor Strich“). Crimson implementiert diese Reihenfolge mithilfe der Operator-Priorität, wobei die einzelnen Operatoren in einer Gruppe platziert werden und Operatoren von der Gruppe mit der niedrigsten Nummer bis hin zur höchsten angewendet werden. Sofern nicht anders angegeben, werden die Operatoren innerhalb einer Gruppe von links nach rechts bewertet. Die Standardreihenfolge der Bewertung kann mit Klammern außer Kraft gesetzt werden.

24.4 Typenkonvertierung

In der Regel unterscheidet Crimson automatisch, wann es bei der Bewertung eines Ausdrucks notwendig ist, von Integermathematik zur Berechnung durch die Verwendung eines Gleitkommawerts zu wechseln. Wenn Sie z. B. einen Integerwert durch einen Gleitkommawert dividieren, wird der Integer in eine Gleitkommazahl umgewandelt, bevor die Division ausgeführt wird. Es kann Situationen geben, in denen Sie eine solche Umwandlung erzwingen möchten.

Angenommen, Sie möchten drei Integer addieren, die die Füllstände dreier Behälter darstellen, und dann die Summe durch die Anzahl der Behälter teilen, um den Durchschnittswert zu erhalten. Wenn Sie einen Ausdruck wie $(\text{Tank1}+\text{Tank2}+\text{Tank3})/3$ verwenden, dann ist das Ergebnis möglicherweise nicht so präzise wie erforderlich, da die Division mit Integer erfolgt und der berechnete Durchschnittswert keine Dezimalstellen enthält. Um Crimson zu zwingen, das Ergebnis mit Gleitkommamathematik auszuwerten, können Sie einfach die 3 in $3,0$ ändern. Dadurch zwingen Sie Crimson, die Summe vor der Division in einen Gleitkommawert umzuwandeln. Eine etwas komplexere Lösung ist die Verwendung einer Syntax wie `float(Tank1+Tank2+Tank3)/3`. Dadurch wird für den Term in Klammern eine sogenannte Typenkonvertierung aufgerufen, d. h., er wird manuell in einen Gleitkommawert umgewandelt.

Mit der Typenkonvertierung können Sie auch einen Gleitkommawert in einen Integerwert umwandeln, vielleicht um bewusst einen Teil der Genauigkeit eines Zwischenwerts einzubüßen, bevor er in einem SPS-Register gespeichert wird. Zum Beispiel berechnet der Ausdruck `int(cos(Theta)*100)` den Cosinus eines Winkels, multipliziert diesen Wert mithilfe von Gleitkommamathematik mit 100 und wandelt ihn in eine Ganzzahl um, wobei alle Ziffern nach dem Dezimalkomma wegfallen.

24.5 Vergleichen von Werten

Es dürfte häufig vorkommen, dass Sie den Wert eines Datenelements mit einem anderen vergleichen und auf der Grundlage des Ergebnisses eine Entscheidung treffen müssen. Beispielsweise könnten Sie eine Markierungsformel definieren, die angibt, wann ein Behälter einen bestimmten Wert überschreitet, oder vielleicht Sie möchten eine `if`-Anweisung in einem Programm verwenden, um ein wenig Code auszuführen, wenn ein Motor die gewünschte Geschwindigkeit erreicht. Die folgenden Vergleichsoperatoren werden bereitgestellt:

Bediener	Priorität	Beispiel
Gleich	Gruppe 7	Data == 100
Ungleich	Gruppe 7	Data != 100
Größer als	Gruppe 6	Data > 100
Größer oder gleich	Gruppe 6	Data >= 100
Kleiner als	Gruppe 6	Data < 100
Kleiner oder gleich	Gruppe 6	Data <= 100

Jeder Operator erzeugt je nach der geprüften Bedingung einen Wert von 0 oder 1. Die Operatoren können für Integer, Gleitkommawerte oder Zeichenfolgen verwendet werden. Wenn Zeichenfolgen verglichen werden, wird die Groß- und Kleinschreibung nicht beachtet, d. h. „abc“ und „ABC“ werden als gleich betrachtet.

24.6 Testen von Bits

Crimson ermöglicht Ihnen, den Wert eines Bits in einem Datenwert mithilfe des Bitauswahloperators zu testen, der durch einen einzelnen Punkt dargestellt wird. Auf der linken Seite des Operators muss der Wert stehen, in dem das Bit getestet werden soll, und auf der rechten Seite ein Ausdruck, der die zu testende Bitnummer angibt. Dieser rechte Wert muss zwischen 0 und 31 liegen. Das Ergebnis des Operators ist gleich 0 oder 1, je nach dem Wert des fraglichen Bits.

Bediener	Priorität	Beispiel
Bitauswahl	Gruppe 1	Input.2

Das gezeigte Beispiel testet Bit 2 (d. h. das Bit mit dem Wert 4) in dem angegebenen Tag.

Wenn Sie testen möchten, ob ein Bit gleich Null ist, können Sie den logischen Operator NICHT verwenden:

Bediener	Priorität	Beispiel
Logisch NOT	Gruppe 2	!Input.2

Dieses Beispiel ist gleich 1, wenn Bit 2 des angegebenen Tags gleich 0 ist und umgekehrt.

24.7 Mehrere Bedingungen

Wenn Sie einen Ausdruck definieren möchten, der „true“ ist, wenn von mehreren Bedingungen *alle* „true“ sind, können Sie den logischen Operator AND verwenden. Wenn Sie einen Ausdruck definieren möchten, der „true“ ist, wenn von mehreren Bedingungen *beliebig viele* „true“ sind, können Sie den logischen Operator OR verwenden. Die folgenden Beispiele zeigen, wie die Operatoren eingesetzt werden:

Bediener	Priorität	Beispiel
Logisches NOT	Gruppe 11	A>10 && B>10
Logisches OR	Gruppe 12	A>10 B>10

Der logische AND-Operator erzeugt den Wert 1 nur dann, wenn die Ausdrücke auf der linken und rechten Seite „true“ sind. Der logische OR-Operator hingegen erzeugt den Wert 1, wenn einer der beiden Ausdrücke „true“ ist. Beachten Sie, dass die logischen Operatoren anders als die bitweisen Operatoren, die an anderer Stelle in diesem Abschnitt erwähnt werden, die Auswertung beenden, sobald sie die Antwort kennen. Dies bedeutet, dass im obigen Beispiel für das logische AND die rechte Seite des Operators nur dann ausgewertet wird, wenn A größer ist als 10. Wenn dies nicht zutrifft, muss das Ergebnis des AND-Operators bereits Null sein. In den oben genannten Beispielen macht diese Eigenschaft keinen großen Unterschied. Wenn jedoch die Ausdrücke auf der linken oder rechten Seite ein Programm aufrufen oder einen Datenwert ändern, muss dieses Verhalten berücksichtigt werden.

24.8 Auswählen von Werten

Es können Situationen entstehen, in denen Sie je nach dem Wert einer Bedingung zwischen zwei Werten wählen möchten, seien dies Integer, Gleitkommawerte oder Zeichenfolgen. Angenommen, Sie möchten die Drehzahl eines Motors ausgehend von einem Markierungs-Tag gleich 500 U/min bzw. 2000 U/min setzen. Dieser Vorgang kann mit dem Operator `?` durchgeführt werden. Dieser Operator ist einzigartig, da er drei Argumente benötigt, wie im folgenden Beispiel gezeigt:

Bediener	Priorität	Beispiel
Auswahl	Gruppe 13	Fast ? 2000 : 500

Dieses Beispiel wird als 2000 ausgewertet, wenn `Fast` „true“ ist, ansonsten als 500. Der Operator kann als gleichwertig zur Funktion `IF` betrachtet werden, die Sie in Anwendungen wie Microsoft Excel finden.

24.9 Manipulation von Bits

Crimson stellt eine Reihe von Operatoren für die Durchführung von Vorgängen bereit, die Ganzzahlen nicht als numerische Werte behandeln, sondern als Bitfolgen. Diese Operatoren werden als bitweise Operatoren bezeichnet.

24.9.1 AND, OR und XOR

Diese drei bitweisen Operatoren erzeugen jeweils ein Ergebnis, in dem jedes Bit als gleich den entsprechenden Bits in den Werten auf der linken und rechten Seite des Operators definiert ist, wobei eine bestimmte Wahrheitstabelle verwendet wird:

Bediener	Priorität	Beispiel
Bitweises AND	Gruppe 8	Data & Mask
Bitweises OR	Gruppe 9	Data Mask
Bitweises XOR	Gruppe 10	Data ^ Mask

Die folgende Tabelle zeigt die zugehörigen Wahrheitstabellen:

A	B	A & B	A B	A ^ B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

24.9.2 Shift-Operatoren

Crimson stellt auch Operatoren bereit, die einen Integer um eine Anzahl von Bits nach links oder rechts verschieben:

Bediener	Priorität	Beispiel
Nach links verschieben	Gruppe 5	Data << 2
Nach rechts verschieben	Gruppe 5	Data >> 2

Jedes Beispiel verschiebt `Data` um zwei Bits in die angegebene Richtung.

24.9.3 Bitweises NOT

Crimson stellt einen bitweisen NOT-Operator bereit, mit dem die Richtung der Bits in einem Wert umgekehrt wird:

Bediener	Priorität	Beispiel
Bitweises NOT	Gruppe 2	~Mask

Dieses Beispiel erzeugt einen Wert, bei dem jedes Bit gleich dem Gegenteil seines Werts in `Mask` ist.

24.10 Indizierung von Arrays

Elemente innerhalb eines Array-Tags können ausgewählt werden, indem nach dem Array-Namen eckige Klammern eingefügt werden, die einen Indizierungsausdruck enthalten. Dieser Ausdruck muss zwischen 0 und der Anzahl der Elemente in dem Array minus 1 liegen. Wenn Sie beispielsweise ein Array aus 10 Elementen erstellen, ist das erste Element `Name[0]` und das letzte `Name[9]`.

24.11 Indizierung von Strings

Eckige Klammern können auch verwendet werden, um Zeichen innerhalb eines Strings auszuwählen. Bei einem Tag namens „Text“ mit der Zeichenfolge „ABCD“ beispielsweise gibt der Ausdruck `Text[0]` den Wert 65 zurück, weil dies dem Unicode-Wert des ersten Zeichens entspricht. Indexwerte jenseits Stringendes geben immer Null zurück.

24.12 Hinzufügen von Strings

Der Additionsoperator kann nicht nur Zahlen addieren, sondern auch Strings verketteten. Der Ausdruck `"AB" + "CD"` ergibt somit „ABCD“. Mit dem Additionsoperator können Sie einer Zeichenfolge auch einen Integer hinzufügen. In diesem Fall wird ein einzelnes Zeichen, das der Unicode-Darstellung des Integers entspricht, an die Daten im String angehängt.

24.13 Aufrufen von Programmen

Programme, die Werte zurückgeben, können innerhalb von Ausdrücken aufgerufen werden, indem nach dem Programmnamen ein Paar Klammern eingegeben wird. Zum Beispiel ruft `Program1()*10` das zugehörige Programm auf und multipliziert den Rückgabewert mit 10. Der Rückgabebetyp für `Program1` muss auf Integer oder Gleitkomma gesetzt werden, damit dies Sinn ergibt.

24.14 Verwenden von Funktionen

Crimson bietet eine Reihe vordefinierter Funktionen, die für den Zugriff auf Systeminformationen oder für die Durchführung gängiger Rechenoperationen verwendet werden können. Diese Funktionen werden in der Funktionsreferenz ausführlich beschrieben. Sie werden mit einer Syntax aufgerufen, die der Syntax für Programme ähnelt, wobei Argumente für die Funktion in Klammern eingeschlossen werden. Zum Beispiel ruft `cos(0)` die Cosinus-Funktion mit dem Argument 0 auf und gibt den Wert +1,0 zurück.

24.15 Priorität Zusammenfassung

Die folgende Tabelle zeigt die Priorität aller in diesem Abschnitt definierten Operatoren:

Gruppe	Operatoren
Gruppe 1	.
Gruppe 2	! ~
Gruppe 3	* / %
Gruppe 4	+ -
Gruppe 5	<< >>
Gruppe 6	< > <= >=
Gruppe 7	== !=
Gruppe 8	&
Gruppe 9	
Gruppe 10	^
Gruppe 11	&&
Gruppe 12	
Gruppe 13	?:

Die Operatoren in den Gruppen mit niedrigen Nummern werden zuerst angewendet.

Kapitel 25 Schreiben von Aktionen

Während Ausdrücke Werte definieren, definieren Aktionen, was geschehen soll, wenn ein Ereignis eintritt. Die meisten Aktionen in einer Datenbank beziehen sich auf Interaktionen mit Primitiven oder mit der Tastatur. Da Crimson® 3.1 eine einfache Methode zum Definieren häufig verwendeter Aktionen für diese Elemente bietet, können Sie oft vermeiden, Aktionen von Hand zu schreiben. Aktionen sind jedoch erforderlich, wenn Sie Auslöser verwenden, Programme schreiben oder Primitive im Modus „User-Defined“ (Benutzerdefiniert) verwenden möchten.

25.1 Ändern einer Seite

Um eine Aktion zu erstellen, welche die auf dem Bedienfeld-Display angezeigte Seite ändert, verwenden Sie die Syntax `GotoPage (Name)`. Dabei ist `Name` der Name der fraglichen Anzeigeseite. Die aktuelle Seite wird entfernt und stattdessen wird die neue Seite angezeigt.

25.2 Ändern numerischer Werte

Crimson bietet verschiedene Möglichkeiten zum Ändern von Datenwerten.

25.2.1 Einfache Zuweisung

Um eine Aktion zu erstellen, die einem Tag oder einem Register in einem Kommunikationsgerät einen neuen Wert zuweist, verwenden Sie die Syntax `Data=Value`, wobei `Data` das zu ändernde Datenelement ist und `Value` der Wert, der zugewiesen werden soll. Beachten Sie, dass `Value` nicht nur ein konstanter Wert sein muss, sondern ein beliebiger gültiger Ausdruck des korrekten Typs sein kann. Weitere Informationen zum Schreiben von Ausdrücken finden Sie im vorherigen Abschnitt. Zum Beispiel kann Code wie `[N7:0]=Tank1+Tank2` dazu verwendet werden, zwei Behälterfüllstände hinzuzufügen und die Gesamtmenge direkt in einem SPS-Register zu speichern. Beachten Sie, dass alle Zuordnungsoperatoren in Gruppe 14 fallen. Mit anderen Worten, sie werden nach allen anderen Operatoren in einer Aktion ausgewertet. Sie sind außerdem einzigartig, weil sie von rechts nach links gruppiert werden. Dies bedeutet, dass ein Codefragment wie `Tag1=Tag2=Tag3=0` dazu verwendet werden kann, alle drei Tags gleichzeitig zu löschen.

25.2.2 Zusammengesetzte Zuweisung

Zum Erstellen einer Aktion, die einen Datenwert mithilfe eines der im vorherigen Abschnitt beschriebenen Operatoren gleich seinem aktuellen Wert in Kombination mit einem anderen Wert setzt, verwenden Sie die Syntax `Dataop=Value`. Dabei ist `Data` das zu ändernde Tag; `Value` der Wert, der vom Operator verwendet werden soll und `op` ein beliebiger der verfügbaren Operatoren. Dazu ein Beispiel: Der Code `Tag+=10` erhöht `Tag` um den Wert 10, während `Tag*=10` den aktuellen Wert mit 10 multipliziert.

25.2.3 Schrittweise erhöhen und verringern

Um eine Aktion zu erstellen, die einen Datenwert um 1 erhöht, verwenden Sie die Syntax `Data++`. Um eine Aktion zu erstellen, die einen Datenwert um 1 verringert, verwenden Sie die Syntax `Data--`. Beachten Sie, dass die Operatoren `++` oder `--` vor oder hinter dem betreffenden Datenwert platziert werden können. Bei der Platzierung vor dem Datenwert ist der Wert des Ausdrucks, der durch `++Data` dargestellt wird, gleich dem Wert von `Data`, nachdem er erhöht wurde. Bei der Platzierung nach dem Datenwert ist der Ausdruck gleich dem Wert, bevor er geändert wurde.

25.3 Ändern von Bitwerten

Um ein Bit innerhalb eines Tags zu ändern, verwenden Sie die Syntax `Data.Bit=1` oder `Data.Bit=0`, um das Bit nach Bedarf zu setzen oder zu löschen. Dabei ist `Data` das besagte Tag und `Bit` die auf Null basierende Bitnummer. Beachten Sie auch hier, dass der Wert auf der rechten Seite des Operators = auf Wunsch ein Ausdruck sein kann. Das heißt, dass ein Beispiel wie `Data.1=(Level>10)` verwendet werden kann, um ein Bit zu setzen oder zu löschen, je nachdem, ob eine Behälterfüllung einen voreingestellten Wert überschreitet.

25.4 Ausführen von Programmen

Programme können innerhalb von Aktionen aufgerufen werden, indem nach dem Programmnamen ein Paar Klammern eingegeben wird. Mit `Program1()` z. B. wird das zugehörige Programm aufgerufen. Das Programm wird je nach den definierten Programmeigenschaften im Vorder- oder Hintergrund ausgeführt.

25.5 Verwenden von Funktionen

Crimson bietet eine Reihe vordefinierter Funktionen, die für die Durchführung verschiedener Vorgänge verwendet werden können. Diese Funktionen werden in der Funktionsreferenz ausführlich beschrieben. Sie werden mit einer Syntax aufgerufen, die der Syntax für Programme ähnelt, wobei Argumente für die Funktion in Klammern eingeschlossen werden. Zum Beispiel setzt `SetLanguage(1)` die Terminalsprache auf 1.

Kapitel 26 Erweitertes Debugging

Crimson 3.1 verfügt über verschiedene Funktionen, mit denen Sie komplexe Programme einfacher erstellen und debuggen können. Auf diese Funktionen wird über Debug-Konsolen zugegriffen, die jeweils einen Mechanismus zur Anzeige von Diagnosenachrichten und optional zur Eingabe von Systembefehlen bieten.

26.1 Auf eine Konsole zugreifen

Debug-Konsolen können auf drei Arten erstellt werden:

- Der Webserver kann so konfiguriert werden, dass er eine Konsole anzeigt, auf die über das Internet zugegriffen werden kann.
- Eine Serienschchnittstelle kann als serielle Debug-Konsole konfiguriert werden.
- Ein Netzwerkprotokoll kann so konfiguriert werden, dass es eine TCP/IP-Debug-Konsole bereitstellt.

Da Debug-Konsolen einen Systemzugriff auf niedriger Ebene bieten, sollten sie in Produktionseinheiten nicht aktiviert bleiben, es sei denn, es wurde eine entsprechende Sicherheit eingerichtet. Für den Webserver bedeutet dies, dass die Sicherheitseinstellungen verwendet werden, um den Zugriff auf eine Untergruppe authentifizierter Benutzer zu beschränken. Bei einer seriellen Konsole bedeutet dies eine Einschränkung des physischen Zugriffs auf das Gerät. Bei einer TCP/IP-Konsole bedeutet dies, dass die Funktion vollständig deaktiviert wird, da sie nicht angemessen gesichert werden kann.

26.2 Anzeigen der Ausgabe

Verschiedene Elemente innerhalb von Crimson geben Informationen an alle Debug-Konsolen aus, die zu diesem Zeitpunkt aktiv sind. Die MQTT-Konnektoren und der OPC UA-Server können so konfiguriert werden, dass Informationen mittels der Eigenschaft *Debug Output* (Debug-Ausgabe) ausgegeben werden, ebenso Dienste wie der FTP-Server, der FTP-Client des Synchronisationsmanagers und der SMTP-Client des Mail-Managers. **Da sich die Debug-Ausgabe auf die Leistung auswirkt, sollte sie in Produktionseinheiten nicht aktiviert bleiben.**

Die von Crimson erzeugte Debug-Ausgabe ist für erfahrenere Benutzer vorgesehen, die bereits mit dem Protokoll vertraut sind, das sie überwachen. Daher geben wir in diesem Handbuch keine Informationen zum Inhalt dieser Ausgabe oder zu seiner Interpretation. Wenn Sie Hilfe bei der Diagnose von Verbindungsproblemen benötigen, wenden Sie sich bitte an den technischen Support von Red Lion, der Sie möglicherweise zur Weiterleitung der entsprechenden Debugging-Informationen auffordert.

26.3 Debugging-Programme

Bei der Erstellung komplexer Programmgruppen ist es manchmal schwierig, zu wissen, was in einem Crimson-Gerät vor sich geht. Wenn Sie im vorherigen Abschnitt *Programme erstellen* nachlesen, werden Sie feststellen, dass mehrere Eigenschaften hinzugefügt wurden, die es einfacher machen, der Software unter die Haube zu sehen.

Wenn Sie die Debug-Ausgabe aktivieren, wird bei bestimmten Aktionen eine Nachricht an alle aktiven Konsolen ausgegeben. Das folgende Beispiel zeigt die Ausgabe, die generiert werden könnte, wenn eine Schaltfläche auf einer Seite ein Programm aufruft, das dann bestimmte lokale Variablen festlegt (*i*, *j* und *k*) und anschließend ein weiteres Programm aufruft, welches eine andere lokale Variable festlegt, wonach einige Tags angepasst werden und dann ein Wert zurückgegeben wird, der von der ursprünglichen Aktion `Tag1` zugewiesen wird...

```
957.570 : USER      : Pages.Page1.ImageButton1.Action.Press(px = 69, py = 32)
957.570 : USER      :   Program1(a = 1234, b = 1.234, c = "1234")
957.570 : USER      :     i := 1
957.570 : USER      :     j := 3.141
957.570 : USER      :     k := "FRED"
957.570 : USER      :   Program2(void)
957.570 : USER      :     b := 2
957.570 : USER      :     return
957.570 : USER      :   Tag2 := 1
957.570 : USER      :   Tag3 := 3.141
957.570 : USER      :   Tag4 := "FRED"
957.570 : USER      :   return 5678
957.570 : USER      : Tag1 := 5678
957.570 : USER      : return
```

Jede Zeile beginnt mit einem Zeitstempel in Sekunden und Millisekunden, gefolgt vom Namen der Task, die den Code ausführt. (Normalerweise ist dies *USER*, es kann aber auch *SCANNER* für Auslöser oder *DISPATCH* für Programme sein, die im Hintergrund ausgeführt werden.) Der Rest der Zeile zeigt die ausgeführte Aktion an, wobei verschachtelte Programmaufrufe durch Einrückungen dargestellt werden. Die erste Zeile zeigt die Aktion, die die gesamte Kette eingeleitet hat. Dabei handelt es sich um eine Aktion *beim Drücken auf eine Bildschaltfläche*, die sich auf *Seite 1* befindet. Sie werden feststellen, dass die Aktion zwei Parameter hat, d. h. die X- und Y-Koordinate des Berührungseignisses, das die Schaltfläche ausgelöst hat. (Das wussten Sie wahrscheinlich nicht, da diese nicht dokumentiert wurden.) Nachfolgende Funktionsaufrufe enthalten auch Parameter und ggf. den Rückgabewert.

26.3.1 Ausführen von Befehlen

Wenn Sie die Webkonsole im Befehlsmodus oder eine andere Konsole verwenden, können Sie auch Befehle eingeben. Diese Befehle haben in der Regel die Form eines Objektnamens, gefolgt von einem Punkt und der auszuführenden Aktion. Eine Liste der verfügbaren Befehle erhalten Sie, indem Sie `diag.help` oder nur `help` eingeben. Beachten Sie jedoch, dass viele dieser Befehle den internen Betrieb von Crimson betreffen und hier nicht dokumentiert sind. **Diese Systembefehle stehen nur für den Fall zur Verfügung, dass Sie vom technischen Support von Red Lion dazu aufgefordert werden, diese zu verwenden. Wenn Sie sie verwenden, ohne ihre Funktionen zu verstehen, kann dies den Betrieb des Geräts unterbrechen oder sogar zu Datenverlust führen.** Einige der gängigeren Befehle werden dennoch im Folgenden beschrieben.

26.3.1.1 Tag-Befehle

- Mit `tags.list` werden alle Tags im System aufgelistet.
- Mit `tags.list name` werden alle Tags aufgelistet, die mit *name* beginnen.
- Mit `tags.get tag` wird der aktuelle Wert von *Tag* angezeigt.
- Mit `tags.set tag value` wird *tag* auf *value* gemäß dem Formatobjekt gesetzt.

23.3.1.2 Programmbefehle

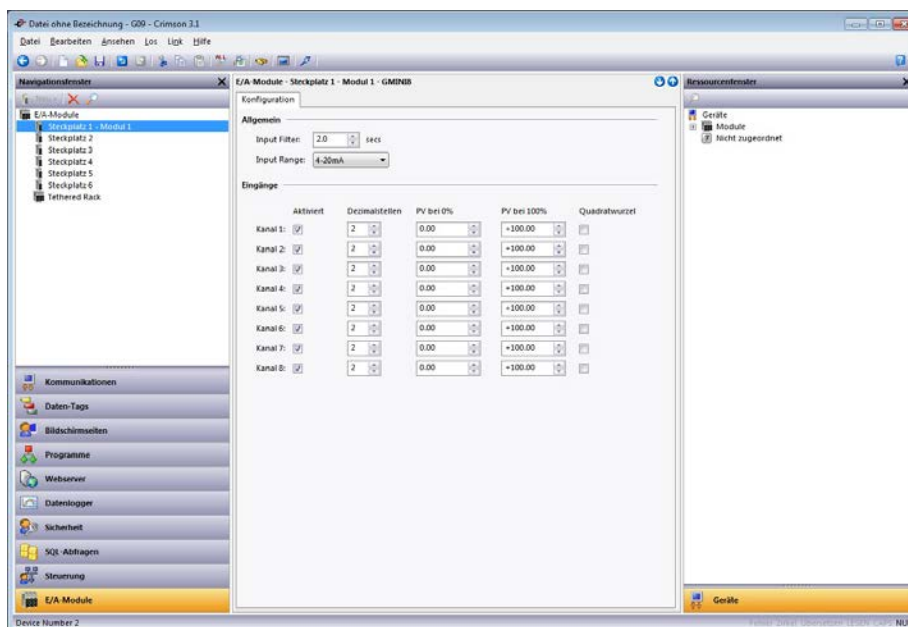
- Mit `progs.list` werden alle Programme im System aufgelistet.
- Mit `progs.run prog` wird *prog* im Hintergrund ausgeführt.

Kapitel 27 Graphite E/A-Module

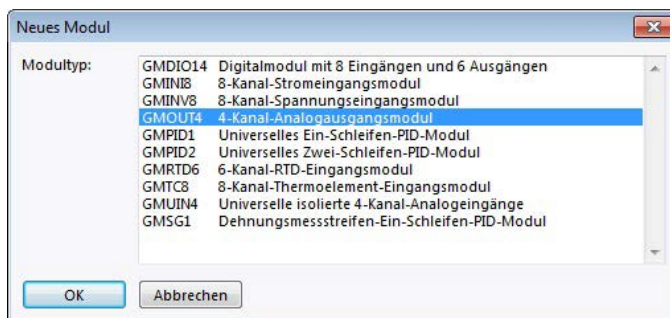
Dieses Kapitel enthält Informationen zum Konfigurieren der E/A-Module, die von den HMIs der Graphite-Serie und den zugehörigen Reglern unterstützt werden. Während Kommunikationsmodule über die Registerkarte „Communications“ (Kommunikationen) konfiguriert werden, verfügen die E/A- und Steuermodule über eine eigene Kategorie namens „I/O Modules“ (E/A-Module), die am unteren Ende der Kategorielliste angezeigt wird.

27.1 Auswählen von Modulen

Wie oben erwähnt, werden Module über die Kategorie „I/O Modules“ (E/A-Module) ausgewählt:



Um ein Modul hinzuzufügen, wählen Sie den gewünschten Steckplatz aus und klicken auf die Schaltfläche „New“ (Neu). Daraufhin wird folgendes Dialogfeld angezeigt:

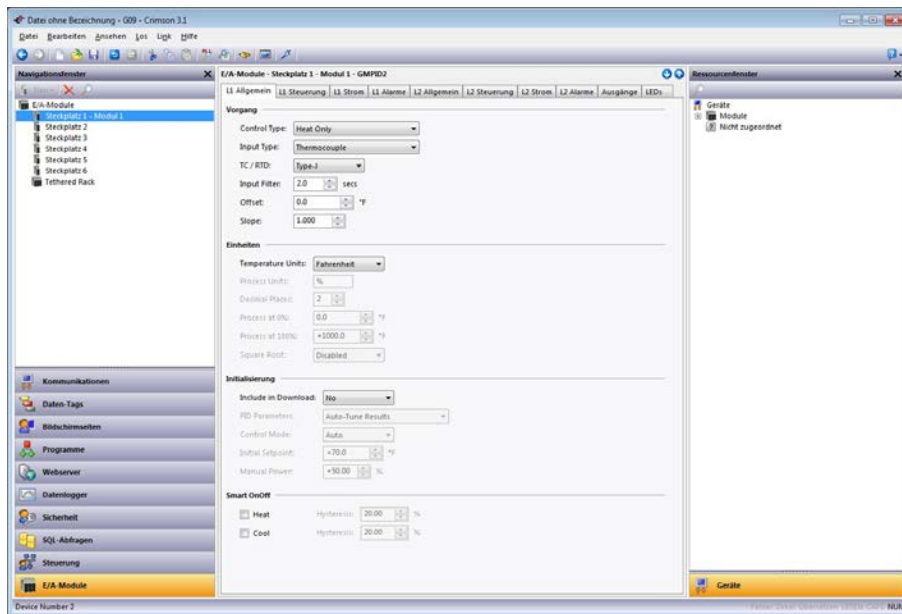


Wählen Sie das gewünschte Modul aus und klicken Sie auf „OK“, um das Dialogfeld zu schließen. Nachdem das Modul erstellt wurde, können Sie einen beschreibenden Namen eingeben. Klicken Sie dazu mit der rechten Maustaste auf das hinzugefügte Modul und wählen Sie dann „Rename“ (Umbenennen) aus oder drücken Sie einfach die Taste F2. Wenn Sie weitere Module hinzufügen möchten, wiederholen Sie den Vorgang. Die Module können per Drag-and-Drop zwischen den Steckplätzen verschoben werden.

27.2 Die GMPID PID-Module

Die Parameter der GMPID-Module sind in Gruppen unterteilt, für die es jeweils eine eigene Seite gibt. Das GMPID2-Modul verfügt über mehrere zusätzliche Registerkarten für die Konfiguration der zweiten Schleife. Verwenden Sie die Registerkarten oben im Fenster, um die verschiedenen Seiten anzuzeigen.

27.2.1 Allgemeine Eigenschaften



27.2.1.1 Vorgang

- Mit der Eigenschaft *Control Type* (Steuertyp) können Sie je nach Art des zu steuernden Prozesses „Heat Only“ (Nur heizen), „Cool Only“ (Nur kühlen) oder „Heat and Cool“ (Heizen und kühlen) auswählen. Wählen Sie für andere als thermische Anwendungen die Option „Heat“ (Heizen) für revers wirkende Anwendungen und „Cool“ (Kühlen) für direkt wirkende.
- Unter der Eigenschaft *Input Type* (Eingangstyp) stehen „RTD“, „Thermocouple“ (Thermoelement) oder ein Prozesseingangstyp zur Auswahl. Bei Auswahl des Modus „RTD“ oder „Thermocouple“ (Thermoelement) wird mit der Eigenschaft *TC/RTD* der verwendete Sensorstandard ausgewählt.
- Der *Input Filter* (Eingangsfiler) ist eine Zeitkonstante zur Stabilisierung schwankender Eingangssignale.
- Mit den Eigenschaften *Slope* (Steigung) und *Offset* (Offset) kann der PV-Wert angepasst oder neu skaliert werden, um einen Fehler im angeschlossenen Sensor auszugleichen. Sie können auch die Korrektur des PV-Werts in Anwendungen ermöglichen, in denen der Sensor den Prozess nicht direkt misst, wodurch ein fester oder variabler Offset entsteht. Der folgende Abschnitt enthält ein Berechnungsbeispiel für die Konfiguration dieser Eigenschaften.

27.2.1.2 Einheiten

- *Temperature Units* (Temperatureinheiten) dient zur Auswahl zwischen Kelvin, Fahrenheit und Celsius.
- Mit der Eigenschaft *Process Units* (Prozesseinheiten) können Sie die Engineering Units (Bedieneinheiten) für den Prozess eingeben. Die Eigenschaft *Decimal Places* (Dezimalstellen) wird verwendet, um Crimson® 3.1 die Anzeige der Engineering Units (Bedieneinheiten) in der richtigen

Auflösung zu ermöglichen. Diese werden nur zum Identifizieren der entsprechenden Felder in der Software verwendet. Die Parameter werden als Teil der Crimson-Datei gespeichert, aber nicht innerhalb des Moduls verwendet.

- Die Eigenschaften *Process at 0%* (Prozess bei 0 %) und *Process at 100%* (Prozess bei 100 %) werden zur Skalierung von DC-Eingangssignalen verwendet. Geben Sie die gewünschte PV-Messung für die minimalen und maximalen Eingangssignalpegel ein. Wenn die Anwendung z. B. einen Eingang von einem Durchflusssensor mit einem Leistungsausgang von 4 bis 20 mA akzeptiert, der 5 bis 105 Gallonen pro Minute darstellt, wählen Sie unter *Input Type* (Eingangstyp) die Option „Process 420mA“, geben Sie unter *Process at 0%* (Prozess bei 0 %) den Wert „5“ ein und unter *Process at 100%* (Prozess bei 100 %) den Wert „105“.
- Mit der Eigenschaft *Square Root* (Quadratwurzel) kann das Gerät in Anwendungen eingesetzt werden, in denen das gemessene Signal das Quadrat des PV ist. Dies ist nützlich bei Anwendungen wie der Messung des Durchflusses mit einem Differenzdruckaufnehmer.

27.2.1.3 Initialisierung

Die Initialisierungsparameter liefern Anfangswerte für Einstellungen, die normalerweise von einem PC oder einer SPS gesteuert werden. In typischen Anwendungen werden diese Einstellungen nur verwendet, bis die Kommunikation zum ersten Mal hergestellt wurde. Ab diesem Punkt übernimmt das externe Gerät die Kontrolle.

- Mit der Eigenschaft *Include in Download* (In Download einbeziehen) wird bestimmt, ob die Initialisierungswerte auf das Modul heruntergeladen werden. Bei Auswahl von „No“ (Nein) können Datenbanken nach Wunsch geändert und heruntergeladen werden, ohne dabei die festgelegten Prozessparameter wie Sollwert und PID-Werte versehentlich zu überschreiben.
- Die Eigenschaft *PID Parameters* (PID-Parameter) legt fest, welche PID-Parameter vom Modul geladen und anschließend zum Steuern des Prozesses verwendet werden. Das Modul steuert den Prozess mithilfe der „Active PID“-Werte und dem „Active Power Filter“ (Wirkleistungsfilter). (Siehe die Variablen *ActConstP*, *ActConstI*, *ActConstD* und *ActFilter* im Diagramm „Verfügbare Daten“ am Ende dieses Abschnitts.) Der aktive Satz wird entweder mit den „User PID Settings“ (Benutzer-PID-Einstellungen) oder mit den Werten für „Auto-Tune Results“ (Auto-Abstimmungsergebnisse) geladen, je nach dem Status des Bits *ReqUserPID*. Wenn das Bit „true“ ist, wird der aktive Satz mit den Variablen des Benutzers geladen. Ist das Bit „false“, werden die Werte geladen, die durch die Auto-Abstimmung ermittelt wurden. Durch die Anpassung der Eigenschaft *PID-Parameters* (PID-Parameter) wird das Bit *ReqUserPID* bei der Initialisierung entsprechend geschrieben.
- Die Eigenschaft *Control Mode* (Steuermodus) bestimmt, ob das Modul bei der Initialisierung im automatischen oder manuellen Modus ist. Im automatischen Modus berechnet der Regler den Ausgang, der erforderlich ist, um den Sollwert zu erreichen und aufrecht zu erhalten, und handelt entsprechend. Im manuellen Modus kann der Ausgang direkt durch Schreiben in die Variable *power* gesteuert werden.
- Die Eigenschaft *Initial Setpoint* (Anfänglicher Sollwert) wird bei der Initialisierung als Sollwert verwendet.
- *Manual Power* (Manuelle Leistung) ist das Niveau, das der PID-Regler im manuellen Modus als gegeben annimmt. Werte über +100 % und unter -100 % können eingegeben werden, um sicherzustellen, dass die im Abschnitt „Power Transfer“ (Leistungsübertragung) definierten Verstärkungen und Offsets die Ausgänge nicht einschränken.

27.2.1.4 Smart OnOff

Smart OnOff (Intelligentes Ein-/Ausschalten) ist für Situationen gedacht, in denen normalerweise eine Ein/Aus-Steuerung verwendet wird, aber auch die Vorteile von PID gewünscht werden. Wenn „Heat“ (Heizen) oder „Cool“ (Kühlen) in diesen Modus versetzt wird, wird der Steuerausgang entweder auf „Ein“ oder „Aus“ gesetzt, ohne Zwischenwerte oder Zeitaufteilung (Time Proportioning). Anstatt den Prozesswert zu verwenden, um zu entscheiden, wann der Ausgang eingeschaltet wird, prüft SmartOnOff den Ausgang der PID-Berechnung und

aktiviert den Ausgang, wenn er die Hälfte der definierten Verstärkung für diesen Kanal überschreitet. Bei den Standardeinstellungen würde SmartOnOff für Heizung beispielsweise die Heizleistung einschalten, wenn der PID-Algorithmus 50 % oder mehr Leistung verlangt, wobei der Hysteresewert verwendet wird, um sicherzustellen, dass kleine Änderungen in der PID-Berechnung kein „Kontaktprellen“ bei Relais erzeugen. Mit der Eigenschaft *Hysteresis* (Hysterese) kann Ausgangs-„Kontaktprellen“ eliminiert werden, indem die Ein- und Aus-Punkte bei der SmartOnOff-Steuerung getrennt werden. Der Wert *Hysteresis* (Hysterese) wird um den Sollwert herum zentriert, sodass die Übergangspunkte des Ausgangs oberhalb und unterhalb des Sollwerts um die Hälfte dieses Werts versetzt werden.

27.2.2 Beispiel für Steigung und „Offset“

Angenommen, der Messwert eines Thermoelements liegt um 3 °F niedriger als die tatsächliche Temperatur, wenn der Prozess bei 200 °F ist, aber nur um 1 °F niedriger als die tatsächliche Temperatur, wenn der Prozess bei 300 °F ist. Um diese Fehler zu korrigieren, müssen Sie einen benutzerdefinierten Wert für „Slope“ (Steigung) und „Offset“ festlegen, damit das Modul den richtigen Wert empfängt und verarbeitet.

Die Steigung kann anhand der folgenden Werte berechnet werden:

$$\text{Slope} = \frac{\text{ActualHigh} - \text{ActualLow}}{\text{ReadingHigh} - \text{ReadingLow}}$$

In unserem Beispiel entspricht dies:

$$\text{Slope} = \frac{300 - 200}{299 - 197} = 0.980$$

Sobald die Steigung bekannt ist, können wir den Offset wie folgt berechnen:

$$\text{Offset} = \text{ActualLow} - (\text{Slope} \times \text{ReadingLow})$$

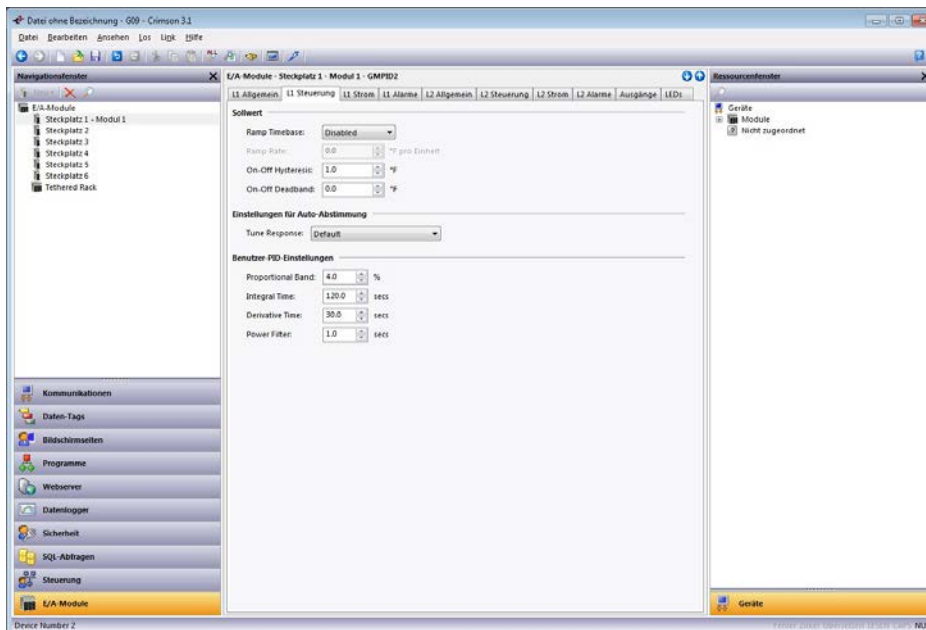
In unserem Beispiel entspricht dies:

$$\text{Offset} = 200 - (0.980 \times 197) = 6.940$$

Mit dem *Slope*-Wert von 0,980 und dem *Offset*-Wert von 6,940 wird der Sensorfehler korrigiert.

Die oben gezeigten Berechnungen sind unabhängig von der Temperaturskala und funktionieren mit °C oder °K genauso.

27.2.3 Steuerungseigenschaften



27.2.3.1 Sollwert

- Die Eigenschaft *Ramp Timebase* (Anstieg Zeitbasis) wählt Sekunden, Minuten oder Stunden als Zeiteinheit für den Anstieg des Prozesses aus.
- Die Eigenschaft *Ramp Rate* (Anstiegsrate) wird verwendet, um eine Schockreaktion eines Prozesses während Sollwertänderungen und Systemstarts zu reduzieren. Eine Sollwert-Anstiegsrate kann verwendet werden, um den tatsächlichen Sollwert kontrolliert zu erhöhen bzw. zu verringern. Der Wert wird in Einheiten/Zeit eingegeben. Ein Wert von „0“ deaktiviert die Sollwert-Änderung. Wenn die „Ramp Rate“ (Anstiegsrate) für den Sollwert ein Wert von nicht Null ist und der angeforderte Sollwert geändert bzw. das Modul eingeschaltet wird, setzt der Regler den tatsächlichen Sollwert auf den aktuellen Prozessmesswert und verwendet diesen Wert als Sollwert. Er passt dann den tatsächlichen Sollwert anhand der „Ramp Rate“ (Anstiegsrate) für den Sollwert an. Wenn der tatsächliche Sollwert den angeforderten Sollwert erreicht, verwendet der Regler wieder den angeforderten Sollwert. (In einem ordnungsgemäß konstruierten und funktionierenden System folgt der Prozess dem tatsächlichen Sollwert zum gewünschten Sollwert.)
- Mit der Eigenschaft *On/Off Hysteresis* (Ein/Aus-Hysterese) wird Ausgangs-„Kontaktprellen“ eliminiert, indem die Ein- und Aus-Punkte der Ausgänge beim Durchführen der Ein/Aus-Steuerung getrennt werden. Der Hysteresewert wird um den Sollwert herum zentriert, sodass die Übergangspunkte des Ausgangs oberhalb und unterhalb des Sollwerts um die Hälfte dieses Werts versetzt werden. Dieser Wert wirkt sich auf die für Heizung oder Kühlung programmierten Ausgänge aus. Während der Auto-Abstimmung lässt der Regler den Prozess 4 Ein- und Ausschaltzyklen durchlaufen. Daher ist es wichtig, die Hysterese auf einen geeigneten Wert einzustellen, bevor der Abstimmungsprozess gestartet wird.
- Die Eigenschaft *On-Off Deadband* (Ein-Aus-Totzone) bietet eine Möglichkeit, die Ein-Punkte von Heizungs- und Kühlausgängen zu versetzen, die für Ein/Aus-Betrieb programmiert sind. Dies führt zu einer Totzone, wenn der Wert positiv ist, und zu einer Überschneidung, wenn der Wert negativ ist. Beim Ermitteln der tatsächlichen Übergangspunkte der Ausgänge muss auch der Wert *On / Off Hysteresis* (Ein/Aus-Hysterese) berücksichtigt werden.

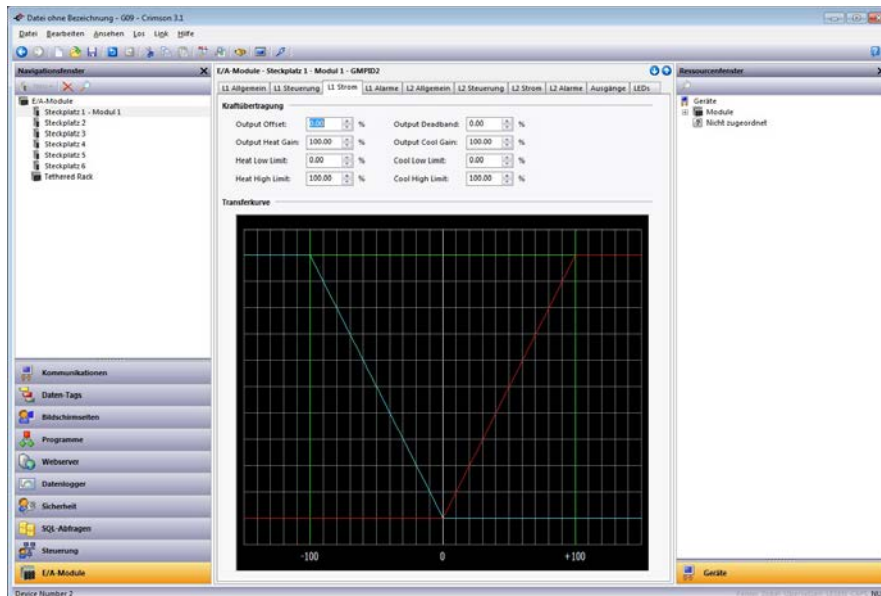
27.2.3.2 Einstellungen für Auto-Abstimmung

- Mit der Eigenschaft *Tune Response* (Abstimmungsreaktion) wird gewährleistet, dass die Auto-Abstimmung die optimalen P-, I- und D-Werte für Ihre spezifische Anwendung liefert. Die Einstellung „Very Aggressive“ (Sehr aggressiv) führt zu einem PID-Parametersatz, der so schnell wie möglich den Sollwert erreicht und dabei ein Überschwingen in Kauf nimmt. Die Einstellung „Very Conservative“ (Sehr vorsichtig) hingegen erreicht den Sollwert langsamer, um ein Überschwingen zu vermeiden. Wenn die Eigenschaft *Tune Response* (Abstimmungsreaktion) geändert wird, muss die Auto-Abstimmung neu initiiert werden, damit die PID-Einstellungen entsprechend angepasst werden. Weitere Informationen finden Sie im Abschnitt „Auto-Abstimmung“.

27.2.3.3 Benutzer-PID-Einstellungen

- Die Eigenschaft *Proportional Band* (Proportionalbereich), die als Prozentsatz des vollen Eingangsbereichs eingegeben wird, ist der Umfang der Eingangsänderung, die erforderlich ist, um den Ausgang um die Vollaussteuerung zu variieren. Bei Temperatureingängen wird der Eingangsbereich anhand des eingegebenen Thermoelements oder RTD-Typs festgelegt. Bei Prozesseingängen ist der Eingangsbereich die Differenz zwischen den Werten für „Process at 0%“ (Prozess bei 0 %) und „Process at 100%“ (Prozess bei 100 %). Die Eigenschaft *Proportional Band* (Proportionalbereich) kann auf einen Prozentsatz von 0 % bis 1000 % eingestellt werden. Es sollte ein Wert gewählt werden, der die beste Reaktion auf eine Prozessstörung bei minimalem Überschwingen bietet. Wenn *Proportional Band* (Proportionalbereich) auf 0 % gesetzt ist, wird der Regler zur Ein-Aus-Steuerung gezwungen. Der optimale Wert für diesen Parameter kann durch Aufrufen der Auto-Abstimmung ermittelt werden.
- Die *Integral Time* (Integralzeit) ist die Zeit in Sekunden, bis das I-Verhalten dem P-Verhalten für einen konstanten Prozessfehler entspricht. Wenn der Fehler weiterhin auftritt, wird das I-Verhalten jede *Integral Time* (Integralzeit) wiederholt. Je höher der Wert, desto langsamer die Reaktion. Der optimale Wert kann durch Aufrufen der Auto-Abstimmung ermittelt werden. Die *Integral Time* (Integralzeit) ist auf einen Wert von 0 bis 6000 Sekunden einstellbar.
- Die *Derivative Time* (Vorhaltezeit) sind die Sekunden pro Wiederholung, die der Regler auf den Anstiegsfehler vorausblickt, um zu sehen, wie hoch der P-Anteil ist, und gleicht diesen Wert bei jeder Vorhaltezeit an. Wenn der Anstiegsfehler weiterhin auftritt, wird der D-Anteil bei jeder Vorhaltezeit wiederholt. Durch eine Erhöhung des Werts wird die Reaktion stabilisiert, doch ein zu hoher Wert in Verbindung mit verrauschten Signalprozessen kann zu zu starken Schwankungen des Ausgangs führen. Bei der Einstellung der Zeit auf Null wird das D-Verhalten deaktiviert. Der optimale Wert für *Derivative Time* (Vorhaltezeit) kann durch Aufrufen der Auto-Abstimmung ermittelt werden. Der Wert lässt sich auf 0 bis 600 Sekunden einstellen.
- Der *Power Filter* (Leistungsfilter) ist eine Zeitkonstante in Sekunden, die die berechnete Ausgangsleistung dämpft. Durch Erhöhung des Wertes wird der Dämpfungseffekt verstärkt. Im Allgemeinen ist ein Leistungsfilter im Bereich von einem Zwanzigstel bis zu einem Fünftel der Integralzeit des Reglers oder der Prozesszeit-Konstanten wirksam. Höhere Werte können aufgrund des zusätzlichen Verzögerungseffekts zu einer Instabilität des Reglers führen.

27.2.4 Leistungseigenschaften



27.2.4.1 Kraftübertragung

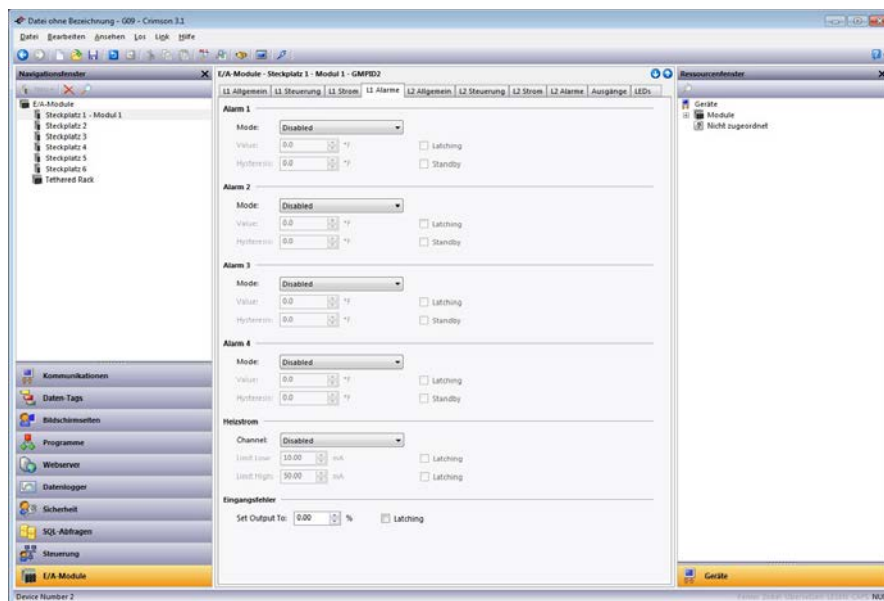
- Der Wert *Output Offset* (Ausgangs-Offset) verschiebt den Nullpunkt der Ausgangsleistungsberechnung des Moduls wirksam. Diese Funktion wird häufig verwendet, um die bleibende Regelabweichung in rein proportionalen Anwendungen zu entfernen.
- Die Eigenschaft *Output Deadband* (Ausgangstotzone) definiert den Bereich, in dem Heiz- und Kühlausgänge beide inaktiv („Totzone“) bzw. aktiv („Überschneidung“) sind. Ein positiver Wert führt zu einer Totzone, ein negativer Wert zu einer Überschneidung.
- *Output Heat Gain* (Ausgangs-Wärmeverstärkung) definiert die Verstärkung des Heizungsausgangs im Verhältnis zur Verstärkung durch den *Proportional Band* (Proportionalbereich). Ein Wert von 100 % führt dazu, dass die Wärmeverstärkung die durch den Proportionalbereich bestimmte Verstärkung imitiert. Ein Wert von weniger als 100 % kann in Anwendungen verwendet werden, in denen die Heizung überdimensioniert ist, und ein Wert von mehr als 100 % kann verwendet werden, wenn die Heizung zu klein ist. Bei den meisten Anwendungen ist der Standardwert 100 % angemessen und Anpassungen sollten nur vorgenommen werden, wenn der Prozess dies erfordert.
- *Output Cool Gain* (Ausgangs-Kühlungsverstärkung) definiert die Verstärkung des Kühlausgangs im Verhältnis zu der durch den *Proportional Band* (Proportionalbereich) etablierten Verstärkung. Ein Wert von 100 % führt dazu, dass die Kühlungsverstärkung die durch den Proportionalbereich bestimmte Verstärkung imitiert. Ein Wert von weniger als 100 % kann in Anwendungen verwendet werden, in denen das Kühlgerät überdimensioniert ist, und ein Wert von mehr als 100 % kann verwendet werden, wenn das Kühlgerät zu klein ist. Bei den meisten Anwendungen ist der Standardwert 100 % angemessen und Anpassungen sollten nur vorgenommen werden, wenn der Prozess dies erfordert.
- Die Eigenschaften *Heat Low Limit* (Heizung Untergrenze) und *Heat High Limit* (Heizung Obergrenze) können verwendet werden, um die Reglerleistung bei Prozessstörungen oder Sollwertänderungen zu begrenzen. Geben Sie die Grenzwerte ein, die beim Prozess für eine sichere Ausgangsleistung sorgen. Sie können Werte über +100 % und unter -100 % eingeben, um den Effekt zu überwinden, bei dem die verschiedenen Verstärkungen und Offsets andernfalls die Ausgänge auf weniger als ihre Maximalwerte begrenzen würden.
- Die Eigenschaften *Cool Low Limit* (Kühlung Untergrenze) und *Cool High Limit* (Kühlung Obergrenze) können verwendet werden, um die Reglerleistung bei Prozessstörungen oder Sollwertänderungen zu

begrenzen. Geben Sie die Grenzwerte ein, die beim Prozess für eine sichere Ausgangsleistung sorgen. Sie können Werte über +100 % und unter -100 % eingeben, um den Effekt zu überwinden, bei dem die verschiedenen Verstärkungen und Offsets andernfalls die Ausgänge auf weniger als ihre Maximalwerte begrenzen würden.

27.2.4.2 Transferkurve

- Das vorherige Leistungsübertragungsdiagramm veranschaulicht die Beziehung zwischen der Sollleistung und den Werten, die an die Heizungs- und Kühlungsausgänge gesendet werden. Die blaue Linie repräsentiert die Kühlung und die rote Linie die Heizung. Das Diagramm wird in Echtzeit aktualisiert, sobald Anpassungen an den anderen Eigenschaften vorgenommen werden.

27.2.5 Alarmeigenschaften



Mit den vier Prozessalarmen kann der Prozessstatus überwacht werden. Sie können optional zur Ansteuerung der physischen Ausgänge des Moduls verwendet werden. Der Status des Alarmbits kann auch über das System selbst oder über externe Geräte überwacht werden.

- Die Eigenschaft *Mode* (Modus) bestimmt, welches Verhalten der Alarm als gegeben annimmt. Die Tabelle unten beschreibt die verschiedenen Auswahlmöglichkeiten.

Modus	Beschreibung
Absoluter Tiefstwert	Der Alarm wird aktiviert, wenn der Prozesswert den Alarmwert unterschreitet. Der Alarm wird deaktiviert, wenn der Prozesswert den Alarmwert + Hysterese überschreitet.
Absoluter Höchstwert	Der Alarm wird aktiviert, wenn der Prozesswert den Alarmwert überschreitet. Der Alarm wird deaktiviert, wenn der Prozesswert den Alarmwert – Hysterese unterschreitet.
Abweichung Tiefstwert	Wenn der Prozesswert den Sollwert um den Alarmwert unterschreitet, wird der Alarm aktiviert. In diesem Modus verfolgt der Alarmpunkt den Sollwert.
Abweichung Höchstwert	Wenn der Prozesswert den Sollwert um den Alarmwert überschreitet, wird der Alarm aktiviert. In diesem Modus verfolgt der Alarmpunkt den Sollwert.
Bandintern	Wenn die Differenz zwischen Sollwert und Prozesswert nicht größer ist als der Alarmwert, wird der Alarm aktiviert.
Bandextern	Wenn der Prozesswert den Sollwert um einen Betrag gleich dem Alarmwert über- oder unterschreitet, wird der Alarm aktiviert. In diesem Modus verfolgt der Alarmpunkt den Sollwert.

- Die Eigenschaft „Value“ (Wert) ist der Punkt, an dem der Alarm eingeschaltet wird. Die Alarmwerte werden als Prozesseinheiten oder Grade eingegeben.
- Der Wert *Hysteresis* (Hysterese) trennt die Ein- und Ausschaltpunkte des Alarms. Zum Beispiel wird ein hoch aktiver Alarm, der laut Programmierung beim Wert 500 und der *Hysterese* 10 eingeschaltet wird, bei 500 eingeschaltet, aber wieder ausgeschaltet, sobald der PV unter 490 fällt. Dies können Sie sich leichter merken, wenn Sie beachten, dass die Hysterese immer darauf hinwirkt, den Alarm aktiv zu halten.
- Die Eigenschaft *Latching* (Verriegelung) bestimmt, wie sich der Alarm nach der Aktivierung verhält. Weitere Informationen finden Sie im nachstehenden Diagramm zum Alarmverhalten. Die Eigenschaft *Standby* (Bereitschaft) bietet eine Möglichkeit, Fehlalarme während der Hochfahrens zu vermeiden. Weitere Informationen finden Sie im nachstehenden Diagramm zum Alarmverhalten.

27.2.5.1 Diagramm zum Alarmverhalten

Zwischen speichern	Standby	Alarmverhalten
Aus	Aus	Der Alarm schaltet sich automatisch ein und aus, wenn der Prozesswert in den Alarmstatus eintritt und diesen verlässt. Das Bit <code>AlarmAccept</code> deaktiviert den Alarm unabhängig vom Status des Prozesses. Wenn die Alarmbedingung vorhanden ist und das Bit <code>AlarmAccept</code> auf 0 gesetzt ist, wird der Alarm aktiviert.
Ein	Aus	Sobald er aktiviert ist, bleibt der Alarm aktiv, bis er angenommen wird. Wenn die Alarmbedingung nicht mehr vorhanden ist, wird sie zurückgesetzt, indem das Bit <code>AlarmAccept</code> auf 1 gesetzt wird. Solange das Bit <code>AlarmAccept</code> 1 lautet, schaltet der Alarm sich automatisch ein und aus, wenn der Prozesswert in den Alarmstatus eintritt und diesen verlässt.
Aus	Ein	Der Alarm schaltet sich automatisch ein und aus, wenn der Prozesswert in den Alarmstatus eintritt und diesen verlässt. Der Alarm wird automatisch deaktiviert, wenn eine Sollwertänderung auftritt oder wenn das Modul zum ersten Mal eingeschaltet wird. Dadurch werden Fehlalarme verhindert. Der Alarm bleibt so lange deaktiviert, bis der Prozess in einen Nicht-Alarmstatus wechselt. Wenn der Prozesswert das nächste Mal in einen Alarmstatus wechselt, wird der Alarm entsprechend aktiviert. Das Bit <code>AlarmAccept</code> deaktiviert den Alarm unabhängig vom Status des Prozesses. Wenn die Alarmbedingung vorhanden ist und das Bit <code>AlarmAccept</code> auf 0 gesetzt ist, wird der Alarm aktiviert.
Ein	Ein	Sobald er aktiviert ist, bleibt der Alarm aktiv, bis er angenommen wird. Der Alarm wird automatisch deaktiviert, wenn eine Sollwertänderung auftritt oder wenn das Modul zum ersten Mal eingeschaltet wird. Dadurch werden Fehlalarme verhindert. Der Alarm bleibt so lange deaktiviert, bis der Prozess in einen Nicht-Alarmstatus wechselt. Wenn der Prozesswert das nächste Mal in einen Alarmstatus wechselt, wird der Alarm entsprechend aktiviert. Ein aktiver Alarm wird ausgeschaltet, wenn das Bit <code>Alarm-Accepts</code> auf 1 gesetzt wird. Falls die Alarmbedingung noch besteht, wenn das Bit auf 0 gesetzt ist, bleibt der Alarm ausgeschaltet und wird in den Standby-Modus versetzt. Dies bedeutet, dass der Alarm ausgeschaltet bleibt, bis die Alarmbedingung entfernt wurde und dann wieder zutrifft. Wenn das Bit <code>Alarm-Accept</code> bei 1 bleibt, wird der Alarm deaktiviert und funktioniert nicht.

27.2.5.2 Heizstrom

Der Heizungsstromalarm ist nützlich für die Überwachung des Zustands von externen Steuerschaltkreisen über den Heizungsstrom-Überwachungseingang. In Abhängigkeit von den Ausgangsbedingungen kann er Relais erkennen, die im offenen oder geschlossenen Zustand ausgefallen sind, sowie defekte Heizelemente.

- Mit der Eigenschaft *Channel* (Kanal) wird ausgewählt, welcher der vier separaten Ausgänge zur Steuerung des Überwachungsprozesses verwendet wird. Dieser Ausgang sollte jener Ausgang sein, der zum Steuern der Heizung verwendet wird, deren Strom gemessen wird.
- Die Eigenschaft *Limit Low* (Grenze niedrig) steht für den maximal zulässigen Strom, der gemessen werden darf, während die Heizung ausgeschaltet ist. Hier kann ein Wert von 0 bis 100 mA eingegeben werden. Wenn der Heizungsstrom-Überwachungseingang einen Stromwert misst, der den Wert *Limit Low* (Grenze niedrig) überschreitet, während der Ausgang ausgeschaltet ist, wird der Alarm aktiv.
- Die Eigenschaft *Limit High* (Grenze hoch) steht für den mindestens akzeptablen Strom, der gemessen werden darf, während der Heizausgang eingeschaltet ist. Hier kann ein Wert von 0 bis 100 mA

eingegeben werden. Wenn der Heizstrom-Überwachungseingang einen Stromwert misst, der den Wert *Limit High* (Grenze hoch) unterschreitet, während der Ausgang eingeschaltet ist, wird der Alarm aktiv.

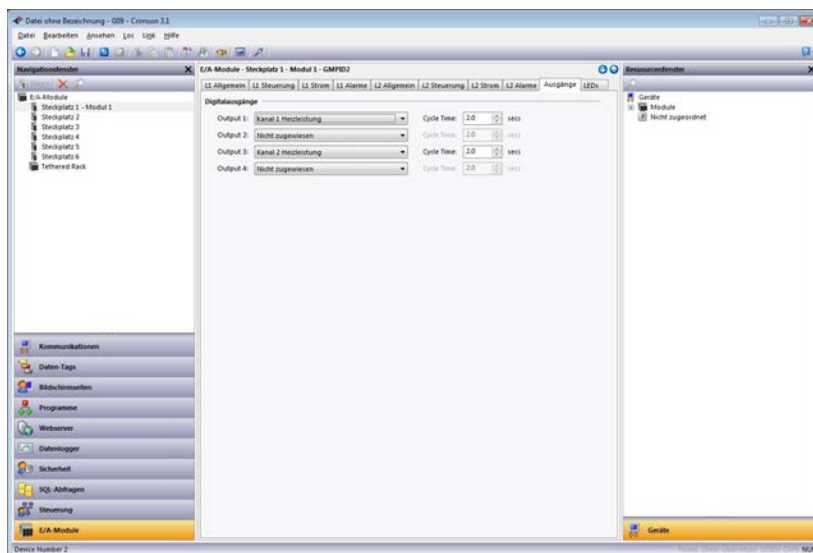
- Die Eigenschaft *Latching* (Verriegelung) bestimmt, ob ein aktivierter Alarm aktiv bleibt, bis er angenommen wird. Um einen gehaltenen Alarm anzunehmen, muss dessen `AlarmAccept`-Bit auf 1 gesetzt werden. Wenn die Verriegelung nicht ausgewählt ist, wird der Alarm automatisch deaktiviert, sobald die Alarmbedingung nicht mehr besteht, und das Bit `AlarmAccept` kann zum Deaktivieren des Alarms verwendet werden.

27.2.5.3 Eingangsfehler

Mit dem Abschnitt „Input Fault“ (Eingangsfehler) wird die Reaktion der Steuerausgänge des GMPID-Moduls im Falle eines Eingangsfehlers definiert. Der Alarm „Input Fault“ (Eingangsfehler) wird als Prozessalarm für Elemente betrachtet, die „Any Process Alarm“ (Beliebiger Prozessalarm) zugeordnet werden können.

- Die Einstellung *Set Output To* (Ausgang setzen auf) ist der Ausgangswert, den der Regler bei einem Eingangssensorfehler als gegeben annimmt. Werte über +100 % und unter -100 % können eingegeben werden, um Einschränkungen zu überwinden, die durch die Leistungsübertragungswerte verursacht werden.
- Wenn die Eigenschaft *Latching* (Verriegelung) aktiviert ist, bleibt das Eingangsfehlerbit so lange aktiv, bis es angenommen wird, unabhängig vom Status des Eingangs. Um den Fehler zurückzusetzen, muss das Bit `InputAccept` auf 1 gesetzt werden. Wenn keine Verriegelung ausgewählt ist, wird der Fehler automatisch deaktiviert, sobald der Eingangsfehler behoben ist.

27.2.6 Ausgangseigenschaften



27.2.6.1 Digitalausgänge

- Die Eigenschaften *Output* (Ausgang) werden verwendet, um den physischen Ausgängen des Moduls verschiedene interne Eigenschaften oder Werte zuzuweisen. Die Liste von GMPID2 wird ausgedehnt, so dass sie die Kanalnummern für jede mögliche Auswahl enthält.
- Mit dem Wert *Cycle Time* (Zykluszeit) wird die Zykluszeit eines Ausgangs bei Verwendung im zeitproportionalen Modus definiert. Dieser Modus ist aktiviert, wenn ein analoger Wert einem Digitalausgang zugewiesen wird, und konvertiert einen Prozentwert in Ein-und-Aus-Zeiten für den Ausgang. Wenn der Algorithmus des Reglers beispielsweise 65 % Strom anfordert und die Zykluszeit

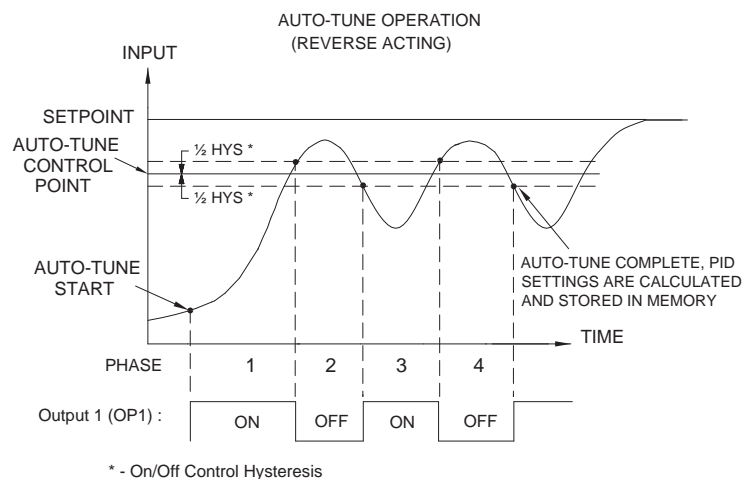
auf 10 Sekunden eingestellt ist, wird der Ausgang 6,5 Sekunden eingeschaltet und 3,5 Sekunden ausgeschaltet. Es wird eine *Cycle Time* (Zykluszeit) kleiner oder gleich einem Zehntel der Prozesszeitkonstante empfohlen.

27.2.6.2 Linearer Ausgang (nur GMPID1)

- Mit der Eigenschaft *Output Type* (Ausgangstyp) wird zwischen Ausgängen mit 0-10 V, 0-20 mA oder 4-20 mA gewählt.
- Mit der Eigenschaft *Mapping* (Zuordnung) wird der Analogausgang des Moduls einer der verschiedenen internen Eigenschaften zugewiesen. Die häufigste Konfiguration ist, die Eigenschaft auf „Heat Power“ (Heizleistung) zu setzen, um einen reinen Heizvorgang über den Analogausgang zu steuern.
- Mit den Werten *Drive Min At* (Antrieb mindestens bei) und *Drive Max At* (Antrieb höchstens bei) kann der Analogausgang mit denselben Einheiten skaliert werden wie der zugeordnete Wert.
- Der *Output Filter* (Ausgangsfilter) ist eine Zeitkonstante, die in Sekunden eingegeben wird und die Reaktion des Analogausgangs dämpft. Durch Erhöhung des Wertes wird der Dämpfungseffekt verstärkt.
- Mit dem Wert *Output Deadband* (Ausgangs-Totzone) kann eine Veränderung des analogen Ausgangs verhindert werden, wenn nur kleine Anpassungen erforderlich sind. Beim Betrieb eines Proportionalventils kann dies einen mechanischen Verschleiß verhindern. Der Analogausgang wird nur dann angepasst, wenn die geforderte Änderung die Hälfte des Totzonenwerts überschreitet. Bei einer Totzone von 10 % und einem Ausgangswert von 50 % ändert sich der Ausgang beispielsweise erst, wenn ein Wert von 45 % oder 55 % angefordert wird.
- Mit der Zeitangabe unter *Output Update* (Ausgangs-Aktualisierung) kann die Aktualisierungshäufigkeit des Analogausgangs verringert werden. Wenn der Timer für *Output Update* (Ausgangs-Aktualisierung) abläuft, prüft der Analogausgang, ob die erforderliche Änderung größer ist als der Wert unter *Output Deadband* (Ausgangs-Totzone). Wenn die erforderliche Änderung größer ist, spiegelt der Ausgang den neuen Wert wider. Andernfalls ändert sich der Ausgang nicht und der Timer beginnt von vorn. Diese Eigenschaft kann mit der Totzone kombiniert werden, um ein Hin- und Herwechseln zwischen Ein und Aus (sogenanntes „Hunting“) bei mechanischen Stellgliedern zu verhindern.

27.2.7 Auto-Abstimmung

Mithilfe der Auto-Abstimmung können die optimalen P-, I-, D- und Stromfilterwerte eingerichtet werden. Das Modul lässt den Prozess vier Ein- und Ausschaltzyklen durchlaufen. Anhand der hieraus erlangten Informationen über den Prozess kann es die am besten geeigneten Steuerschleifenparameter bestimmen.

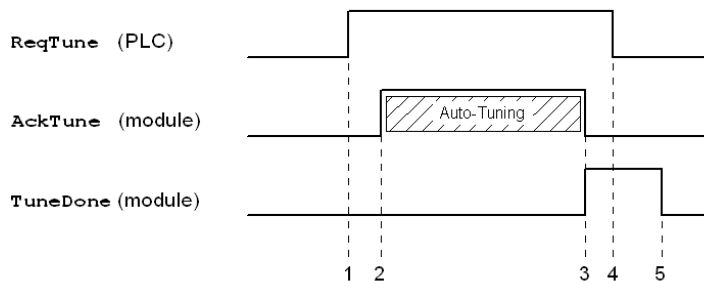


Der bei der Auto-Abstimmung verwendete Sollwert liegt 75 % über der Differenz zwischen aktuellem PV und Sollwert. Die Oszillationen treten damit in der Nähe des Sollwerts auf und gleichzeitig wird übermäßiges Überspringen vermieden. Da das Modul während der Auto-Abstimmung eine Ein/Aus-Steuerung durchführt, ist es wichtig, „On / Off Hysterisis“ (Ein/Aus Hysterese) auf einen geeigneten Wert einzustellen, bevor die Auto-Abstimmung aufgerufen wird. Der PID-Satz, den die Auto-Abstimmung ergibt, kann individuell eingestellt werden, indem der Wert für *Tune Response* (Abstimmungsreaktion) auf „Very Aggressive“ (Sehr aggressiv), „Aggressive“ (Aggressiv), „Default“ (Standard), „Conservative“ (Vorsichtig) oder „Very Conservative“ (Sehr vorsichtig) gesetzt wird. Diese Einstellungen entsprechen den Werten 0 bis 4 im Register „Tune Response“ (Abstimmungsreaktion).

27.2.7.1 Aufrufen der Auto-Abstimmung

Die Auto-Abstimmungssequenz verwendet Request-Acknowledgement-Handshaking (Anforderungsquittierungsbetrieb). Um die Auto-Abstimmung aufzurufen, setzen Sie das Bit `ReqTune` auf 1. Das Modul signalisiert, dass eine Auto-Abstimmung ausgeführt wird, indem es das Bit `AckTune` auf einen hohen Wert setzt. Wenn die Auto-Abstimmung abgeschlossen ist, steigt der Wert des Bits `TuneDone`. An diesem Punkt sollte die externe Logik das Bit zur Anforderung der Auto-Abstimmung ausschalten. Wenn das Modul diese Änderung beobachtet, setzt es das Bit `AckTune` auf 0 zurück. Wenn die Auto-Abstimmung aus irgendeinem Grund fehlschlägt, steigen die Werte für die Bits `TuneDone` und `TuneFail`. Dies kann vorkommen, wenn z. B. ein Eingangsfehler auftritt. In diesem Fall muss die Auto-Abstimmung erneut initialisiert werden.

Ein Anforderungszyklus für die Auto-Abstimmung sieht daher wie folgt aus:



1. SPS gibt `ReqTune` einen hohen Wert.
2. Das Modul startet die Auto-Abstimmung, gibt `AckTune` einen hohen Wert.
3. Die Auto-Abstimmung ist abgeschlossen. `AckTune` nimmt ab, `TuneDone` nimmt zu.
4. SPS sieht den hohen Wert von `TuneDone`, setzt `ReqTune` auf einen niedrigen Wert.
5. Das Modul sieht, dass der Wert von `ReqTune` abnimmt und setzt `TuneDone` auf einen niedrigen Wert.

27.2.8 Verfügbare Daten

Die nachfolgenden Tabellen enthalten die GMPID1-Datenwerte, die dem Graphite-Host zur Verfügung stehen. Beim GMPID2-Modul werden in der Regel dieselben Daten für Schleife 1 und Schleife 2 repliziert. Dezimalstellen dienen nur zur Kennzeichnung der Auflösung und werden nicht gelesen oder geschrieben. Zum Beispiel wird unter `Power` (Leistung) zwar ein Wertebereich von 0 % bis 100,00 % angezeigt, geschrieben werden sollte jedoch 0 bis 10000.

27.2.8.1 Schleifenstatus

Daten	Beschreibung	Bereich	Zugriff
PV	Prozesswert oder Eingabewert des Moduls	Pro Sensor	R
Output	Berechnete Ausgangsleistung der PID-Schleife	+/- 200,00 %	R
HeatPower	Ausgang für Kanäle, die der Heizung zugewiesen sind	0 – 100,00 %	R
CoolPower	Ausgang für Kanäle, die der Kühlung zugewiesen sind	0 – 100,00 %	R
ActSP	Tatsächlicher Sollwert, der derzeit von der PID-Schleife verwendet wird. Dieser darf nicht mit dem angeforderten Sollwert während der Auto-Abstimmung oder in Anwendungen identisch sein, die Sollwert-Änderungen verwenden.	Pro Sensor	R
Fehler	Differenz zwischen Prozesswert und eigentlichem Sollwert	Pro Sensor	R
ColdJunc	Kalibrierwert der Vergleichsstelle	n. z.	R
HCMValue	Heizstrom Eingangswert mA	0 – 100,00 mA	R
AckManual	Bestätigen des manuellen Modus	0 oder 1 (Bit)	R
AckTune	Bestätigen der Auto-Abstimmungsanforderung	0 oder 1 (Bit)	R
TuneDone	Auto-Abstimmung abgeschlossen	0 oder 1 (Bit)	R
TuneFail	Auto-Abstimmung nicht erfolgreich abgeschlossen	0 oder 1 (Bit)	R
Alarm1	Status Alarm 1	0 oder 1 (Bit)	R
Alarm2	Status Alarm 2	0 oder 1 (Bit)	R
Alarm3	Status Alarm 3	0 oder 1 (Bit)	R
Alarm4	Status Alarm 4	0 oder 1 (Bit)	R
HCMAlarmLo	Heizstromüberwachung unterer Alarmgrenzwert	0 oder 1 (Bit)	R
HCMAlarmHi	Heizstromüberwachung oberer Alarmgrenzwert	0 oder 1 (Bit)	R
InputAlarm	Eingang außerhalb des zulässigen Bereichs	0 oder 1 (Bit)	R
AckProfile	Festlegen, ob das Sollwertprofil aktiviert ist	0 oder 1 (Bit)	R
AckHold	Festlegen, ob manuelles Anhalten des Sollwertprofils aktiv ist	0 oder 1 (Bit)	R
AutoHold	Festlegen, ob automatisches Anhalten des Sollwertprofils aktiv ist	0 oder 1 (Bit)	R
ProfDone	Festlegen, ob das Sollwertprofil abgeschlossen ist	0 oder 1 (Bit)	R
ActSegment	Derzeit aktives Segment des Sollwertprofils	0 – 29	R
SegRemain	Verbleibende Zeit für das aktuelle Sollwert-Profilsegment	0 – 32767	R

27.2.8.2 Schleife/Steuerung

Daten	Beschreibung	Bereich	Zugriff
ReqSP	Angeforderter Sollwert, der in den Regler geschrieben wurde. Der tatsächliche Wert kann bei Anwendungen, die Sollwertanstiege („Ramping“) verwenden, sowie im Auto-Abstimmungsmodus anders sein.	Pro Sensor	R/W
Strom	Manuelle Einstellung der Ausgangsleistung	+/- 200,00 %	R/W
AltSP	Alternativer Sollwert, der vom Regler anstelle des Wertes ReqSP verwendet werden soll, wenn das Bit ReqAltSP auf 1 gesetzt ist.	0 oder 1 (Bit)	R/W
AltPV	Alternativer Prozesswert, der vom Regler anstelle des Hardware-PV verwendet werden soll, wenn das Bit ReqAltPV auf 1 gesetzt ist.	0 oder 1 (Bit)	R/W
SetHyst	Sollwert-Hysterese für Ein/Aus-Steuerung	Pro Sensor	R/W
SetDead	Sollwert-Totzone für Ein/Aus-Steuerung	Pro Sensor	R/W
SetRamp	Sollwert-Anstiegsrate	Pro Sensor	R/W
InputFilter	Eingangsfiler	0 – 60,0 Sek	R/W
InputOffset	Eingangs-Offset	Pro Sensor	R/W
InputSlope	Eingangssteigung	0,000 – 10,000	R/W
ReqManual	Manuellen Modus anfordern. Schreiben Sie 1, um den manuellen Modus aufzurufen. Im manuellen Modus wird die Ausgangsleistung durch Schreiben in das Leistungsregister gesteuert.	0 oder 1 (Bit)	R/W
ReqTune	Auto-Abstimmung anfordern Schreiben Sie 1, um die Auto-Abstimmung aufzurufen.	0 oder 1 (Bit)	R/W
ReqUserPID	Benutzer-PID anfordern Schreiben Sie 1, um Benutzerwerte in den aktiven PID-Parametersatz zu laden, oder 0, um Auto-Abstimmungsergebnisse in den aktiven PID-Parametersatz zu laden.	0 oder 1 (Bit)	R/W
ReqProfile	Profil anfordern Schreiben Sie 1, um die Sollwert-Profilerstellung zu starten, oder 0, um den Profilmodus abzubrechen.	0 oder 1 (Bit)	R/W
ReqHold	Anhalten anfordern Schreiben Sie 1, um die Sollwert-Profilerstellung manuell anzuhalten, oder 0, um fortzufahren.	0 oder 1 (Bit)	R/W
ReqAltSP	Alternativen Sollwert anfordern Schreiben Sie 1, um den alternativen Sollwert auszuwählen, oder 0, um den regulären Sollwert zu verwenden.	0 oder 1 (Bit)	R/W
ReqAltPV	Alternativen Prozesswert anfordern Schreiben Sie 1, um den alternativen PV auszuwählen, oder 0, um den Hardware-Eingang zu verwenden.	0 oder 1 (Bit)	R/W
ReqSegment	Segment anfordern Wert des ersten Segments für die Profilerstellung.	0 – 29	R/W
EndSegment	Endsegment Letztes aktives Segment plus 1 für Profilerstellung	1 – 30	R/W
ProfError	Profilfehler Legt die Größe des Fehlerbereichs für das Sollwertprofil fest.	Pro Sensor	R/W
ProfHyst	Profilhysterese – Legt die Größe der Hysterese auf dem Fehlerband fest.	Pro Sensor	R/W

27.2.8.3 Schleife/Leistung

Daten	Beschreibung	Bereich	Zugriff
PowerFault	Leistungsausgangswert für Eingangsfehler	+/- 200,00 %	R/W
PowerOffset	Offset-Wert des Leistungsausgangs	+/- 100,00 %	R/W
PowerDead	Totzonenwert des Leistungsausgangs	+/- 100,00 %	R/W
PowerHeatGain	Wärmeverstärkungswert des Leistungsausgangs	0 – 500,00 %	R/W
PowerCoolGain	Kühlungsverstärkungswert des Leistungsausgangs	0 – 500,00 %	R/W
PowerHeatHyst	SmartOnOff Wärmehysterese des Leistungsausgangs	0 – 50,00 %	R/W
PowerCoolHyst	SmartOnOff Kühlungshysterese des Leistungsausgangs	0 – 50,00 %	R/W
HeatLimitLo	Heizung Untergrenze	0 – 200,00 %	R/W
HeatLimitHi	Heizung Obergrenze	0 – 200,00 %	R/W
CoolLimitLo	Kühlung Untergrenze	0 – 200,00 %	R/W
CoolLimitHi	Kühlung Obergrenze	0 – 200,00 %	R/W

27.2.8.4 Schleife/Alarmer

Daten	Beschreibung	Bereich	Zugriff
AlarmData1	Wert Alarm 1	Pro Sensor	R/W
AlarmData2	Wert Alarm 2	Pro Sensor	R/W
AlarmData3	Wert Alarm 3	Pro Sensor	R/W
AlarmData4	Wert Alarm 4	Pro Sensor	R/W
AlarmHyst1	Hysteresewert Alarm 1	Pro Sensor	R/W
AlarmHyst2	Hysteresewert Alarm 2	Pro Sensor	R/W
AlarmHyst3	Hysteresewert Alarm 3	Pro Sensor	R/W
AlarmHyst4	Hysteresewert Alarm 4	Pro Sensor	R/W
AlarmAccept1	Annahme-Bit Alarm 1	0 oder 1 (Bit)	R/W
AlarmAccept2	Annahme-Bit Alarm 2	0 oder 1 (Bit)	R/W
AlarmAccept3	Annahme-Bit Alarm 3	0 oder 1 (Bit)	R/W
AlarmAccept4	Annahme-Bit Alarm 4	0 oder 1 (Bit)	R/W
HCMLimitLo	Alarmwert Untergrenze Heizstrom	0 – 100,00 mA	R/W
HCMLimitHi	Alarmwert Obergrenze Heizstrom	0 – 100,00 mA	R/W
HCMAcceptLo	Alarm-Annahme Untergrenze Heizstrom	0 oder 1 (Bit)	R/W
HCMAcceptHi	Alarm-Annahme Obergrenze Heizstrom	0 oder 1 (Bit)	R/W
InputAccept	Eingang außerhalb des zulässigen Bereichs – Alarm-Annahme	0 oder 1 (Bit)	R/W

27.2.8.5 Schleife/PID

Daten	Beschreibung	Bereich	Zugriff
TuneCode	Abstimmungsreaktions-Code	0 – 4	R/W
UserConstP	Proportionalwert Benutzer	0 – 1000,0 %	R/W
UserConstI	Integralwert Benutzer	0 – 6000,0 Sek	R/W
UserConstD	Vorhaltewert Benutzer	0 – 600,0 Sek	R/W
UserFilter	Leistungsfilterwert Benutzer	0 – 60,0 Sek	R/W
AutoConstP	Automatisch abgestimmter Proportionalwert	0 – 1000,0 %	R
AutoConstI	Automatisch abgestimmter Integralwert	0 – 6000,0 Sek	R
AutoConstD	Automatisch abgestimmter Vorhaltewert	0 – 600,0 Sek	R
AutoFilter	Automatisch abgestimmter Leistungsfilterwert	0 – 60,0 Sek	R
ActConstP	Aktiver Proportionalwert	0 – 1000,0 %	R
ActConstI	Aktiver Integralwert	0 – 6000,0 Sek	R
ActConstD	Aktiver Vorhaltewert	0 – 600,0 Sek	R
ActFilter	Aktiver Leistungsfilterwert	0 – 60,0 Sek	R

27.2.8.6 Schleife/Profil

Daten	Beschreibung	Bereich	Zugriff
Seg00Time bis Seg29Time	Segmentzeit Anstiegs-/Reduzierungsdauer für bestimmtes Segment.	Siehe unten	R/W
Seg00SP bis Seg29SP	Segmentsollwert Anstiegs-/Reduzierungssollwert für bestimmtes Segment.	Siehe unten	R/W
Seg00Mode bis Seg29Mode	Segmentmodus Anstiegs-/Reduzierungsmodus für bestimmtes Segment.	Siehe unten	R/W

Im technischen Hinweis TNPC18 finden Sie Einzelheiten zu Anstiegs-/Reduzierungsvorgängen. Er ist hier erhältlich:

http://files.redlion.net/filedepot_download/1300/4443

27.2.8.7 Ausgänge/Zykluszeiten

Daten	Beschreibung	Bereich	Zugriff
CycleTime1	Zykluszeit für Ausgang 1	0,1 – 60,0 Sek	R/W
CycleTime2	Zykluszeit für Ausgang 2	0,1 – 60,0 Sek	R/W
CycleTime3	Zykluszeit für Ausgang 3	0,1 – 60,0 Sek	R/W
CycleTime4	Zykluszeit für Ausgang 4 (nur GMPID2)	0,1 – 60,0 Sek	R/W

27.2.8.8 Ausgänge/Externe Daten

Daten	Beschreibung	Bereich	Zugriff
DigRemote1-4	Digital Remote (Digital extern) Ausgänge, die Digital Remote (Digital extern) zugeordnet sind, können gesteuert werden, indem das entsprechende Bit als 1 oder 0 geschrieben wird.	0 oder 1 (Bit)	R/W
AnlRemote1-4	Analog Remote Value (Wert für analog extern) Ausgänge, die Analog Remote (Analog extern) zugeordnet sind, können gesteuert werden, indem eine Zahl an dieses Wort geschrieben wird.	Pro Anwendung	R/W

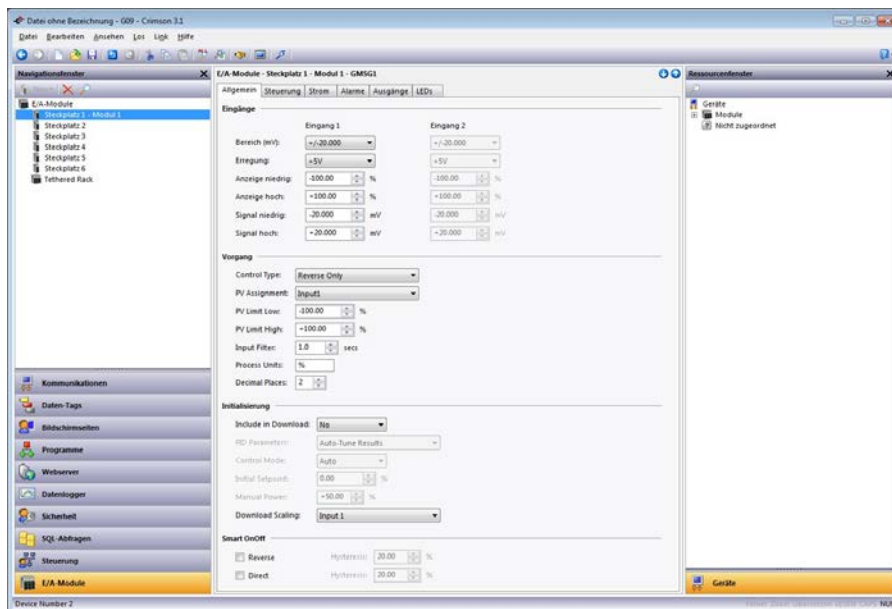
27.2.8.9 Ausgänge/Informationen

Daten	Beschreibung	Bereich	Zugriff
OP1State	Status von Ausgang 1	0 oder 1 (Bit)	R
OP2State	Status von Ausgang 2	0 oder 1 (Bit)	R
OP3State	Status von Ausgang 3	0 oder 1 (Bit)	R
OP4State	Status von Ausgang 4	0 oder 1 (Bit)	R

27.3 Das MSG-Eingangsmodul für Dehnungsmessstreifen

Die Parameter des MSG-Moduls sind in Gruppen unterteilt, für die es jeweils eine eigene Seite gibt.

27.3.1 Allgemeine Eigenschaften



27.3.1.1 Eingänge

Der Abschnitt für Eingangsparameter enthält Einstellungen für beide Eingänge. Wenn das Modul nicht mit dem optionalen sekundären Eingang ausgestattet ist, können die sekundären Eingangsparameter dennoch konfiguriert werden, werden aber ignoriert.

- Mit der Eigenschaft *Range* (Bereich) wird der Eingang für verschiedene Signalpegel konfiguriert.
- Die Eigenschaft *Excitation* (Erreger) stellt die Erregerspannung auf 5 V oder 10 V ein.
- Die Eigenschaften *Display Low* (Anzeige niedrig) und *Display High* (Anzeige hoch) werden zusammen mit den Eigenschaften *Signal Low* (Signal niedrig) und *Signal High* (Signal hoch) verwendet, um die Eingangsspannung auf geeignete Anzeigeeinheiten zu skalieren. Angenommen, eine Anwendung verwendet einen Dehnungsmessstreifen, der für ein Gewicht von 0 bis 1000 lbs eine Spannung von 0 bis 21,00 mV erzeugt. Geben Sie für *Display Low* (Anzeige niedrig) 0 und für *Display High* (Anzeige hoch) 1000 ein, um den Anzeigebereich einzustellen. Geben Sie dann für *Signal Low* (Signal niedrig) 0 und für *Signal High* (Signal hoch) 21,00 ein, um den entsprechenden Eingangsbereich einzustellen.

27.3.1.2 Vorgang

- Mit der Eigenschaft *Control Type* (Steuertyp) können Sie je nach Art des zu steuernden Prozesses „Reverse Only“ (Nur rückwärts), „Direct Only“ (Nur direkt) oder „Direct“ (Direkt) auswählen.
- Die Eigenschaft *PV Assignment* (PV-Zuweisung) gibt an, wie das Modul den gemessenen Prozesswert bestimmt, den der PID-Algorithmus des Moduls zu steuern versuchen wird. Dies kann einfach der Wert „Input1“ oder eines von mehreren mathematischen Ergebnissen von Eingang 1 und Eingang 2 sein.
- Mit den Eigenschaften *PV Limit Low* (PV-Untergrenze) und *PV Limit High* (PV-Obergrenze) wird der Messbereich des PV-Werts bestimmt und danach der Bereich, den das Modul steuern kann. Der gemeldete PV-Wert bleibt an der jeweiligen Grenze fixiert, wenn der Prozess diese Grenzen überschreitet. Bei Überschreitung eines Grenzwerts um mehr als 5 % des Gesamtbereichs geht das Modul davon aus, dass ein Prozessfehler vorliegt. Daraufhin wird der gemeldete PV-Wert gleich dem Wert „PV Limit High“ (PV-Obergrenze).
- Der *Input Filter* (Eingangsfiler) ist eine Zeitkonstante zur Stabilisierung schwankender Eingangssignale.
- Mit der Eigenschaft *Process Units* (Prozesseinheiten) können Sie die Engineering Units (Bedienereinheiten) für den Prozess eingeben. Die Eigenschaft *Decimal Places* (Dezimalstellen) wird verwendet, um Crimson die Anzeige der Engineering Units (Bedienereinheiten) in der richtigen Auflösung zu ermöglichen. Diese werden nur zum Identifizieren der entsprechenden Felder in der Software verwendet. Die Parameter werden als Teil der Crimson-Datei gespeichert, aber nicht innerhalb des Moduls verwendet.

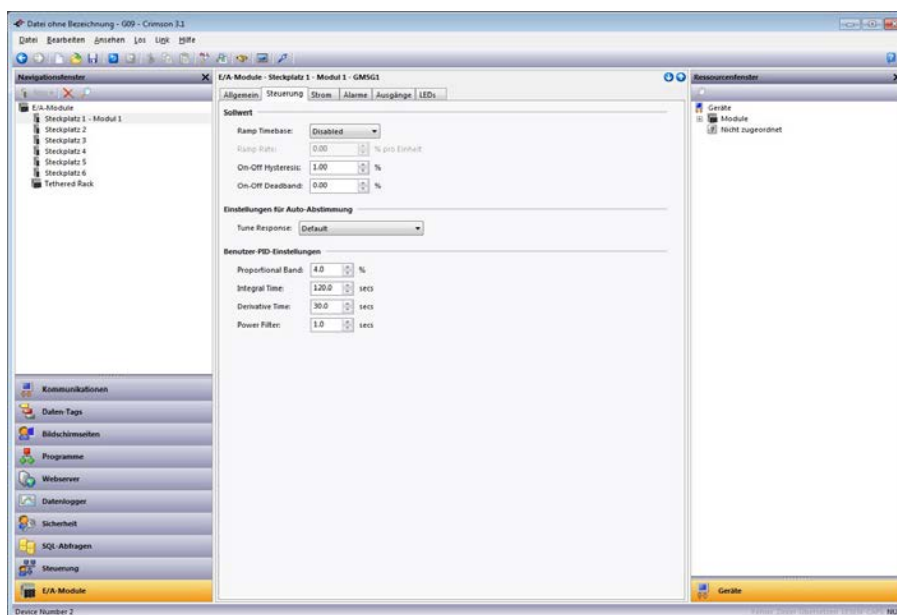
27.3.1.3 Initialisierung

Weitere Informationen zu diesen Eigenschaften finden Sie im Abschnitt „GMPID“.

27.3.1.4 Smart OnOff

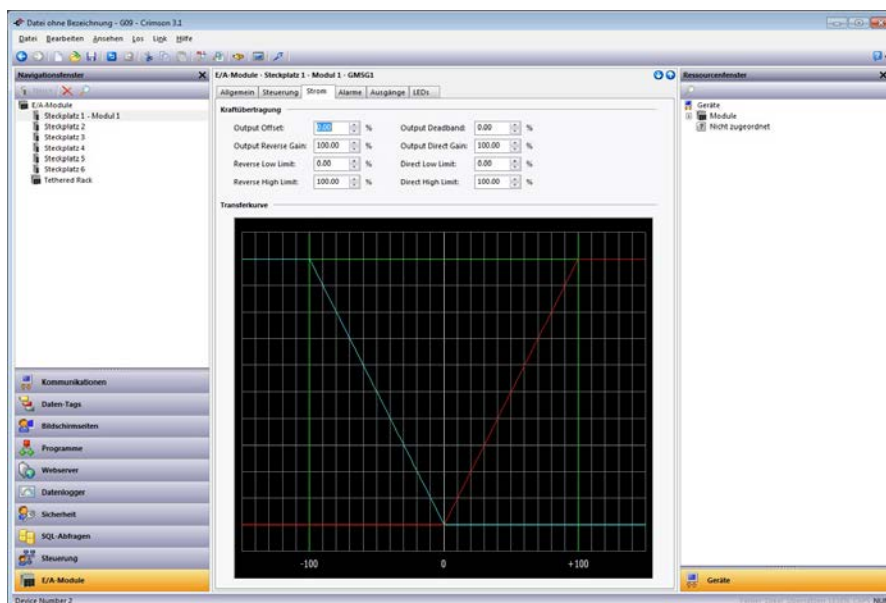
Weitere Informationen zu diesen Eigenschaften finden Sie im Abschnitt „GMPID“.

27.3.2 Steuerungseigenschaften



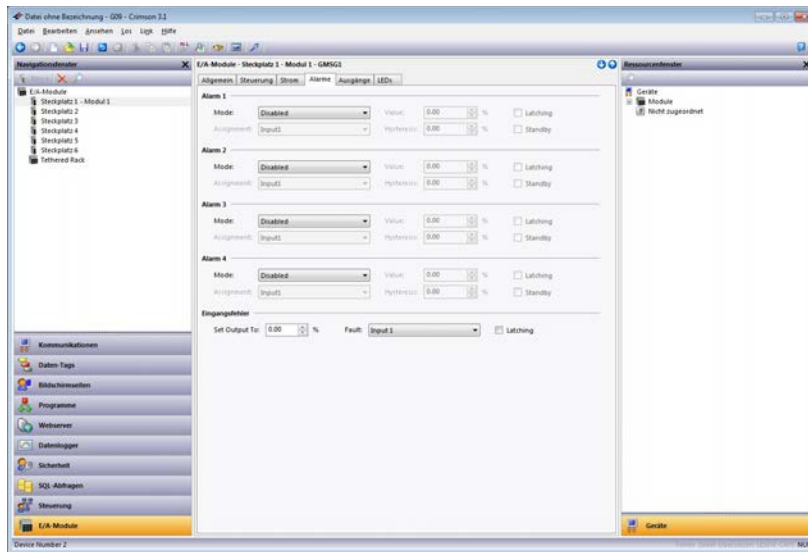
Weitere Informationen zu diesen Eigenschaften finden Sie im Abschnitt „GMPID“.

27.3.3 Leistungseigenschaften



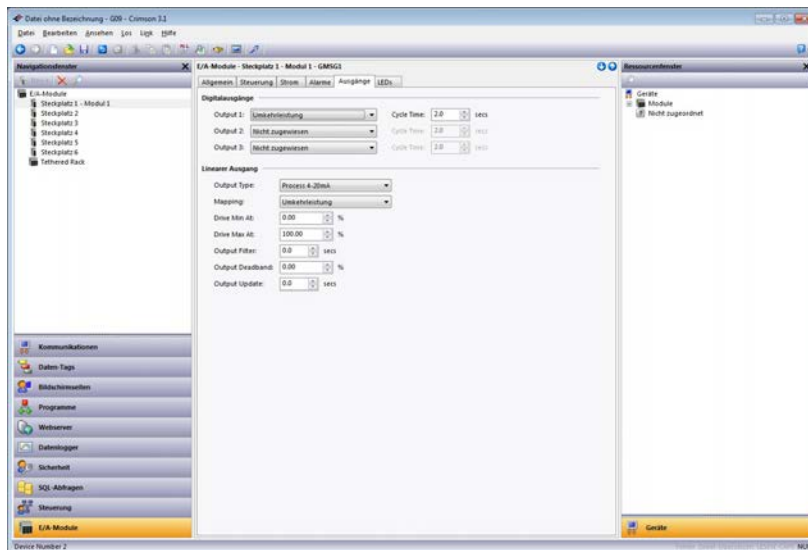
Weitere Informationen zu diesen Eigenschaften finden Sie im Abschnitt „GMPID“.

27.3.4 Alarmeigenschaften



Weitere Informationen zu diesen Eigenschaften finden Sie im Abschnitt „GMPID“.

27.3.5 Ausgangseigenschaften



Weitere Informationen zu diesen Eigenschaften finden Sie im Abschnitt „GMPID“.

27.3.6 Auto-Abstimmung

Weitere Informationen zu diesem Prozess finden Sie im Abschnitt „GMPID“.

27.3.7 Verfügbare Daten

Die nachfolgenden Tabellen enthalten die GMSG-Datenwerte, die dem Graphite-Host zur Verfügung stehen. Dezimalstellen dienen nur zur Kennzeichnung der Auflösung und werden nicht gelesen oder geschrieben. Zum Beispiel wird unter `Power` (Leistung) zwar ein Wertebereich von 0 % bis 100,00 % angezeigt, geschrieben werden sollte jedoch 0 bis 10000. In vielen Fällen sind diese Werte den Werten der GMPID-Module sehr ähnlich mit der Ausnahme, dass anstelle von „Heat“ (Heizen) und „Cool“ (Kühlen) die Begriffe „Reverse“ (Rückwärts) und „Direct“ (Direkt) verwendet werden. Weitere Informationen finden Sie daher in jenem Abschnitt dieses Handbuchs.

27.3.7.1 Schleifen/Status

Weitere Informationen zu diesen Eigenschaften finden Sie im Abschnitt „GMPID“. Zu beachten ist jedoch, dass GMSG zwei Eingänge unterstützt und somit jeweils zwei Werte namens `Input` und `InputAlarm` hat. Beachten Sie auch die oben stehenden Anmerkungen zu Heat/Cool (Heizen/Kühlen) und Reverse/Direct (Rückwärts/Direkt). Wo der Begriff `Heat` bzw. `Cool` in einem Eigenschaftsnamen vorkommt, muss er durch `Rev` bzw. `Dir` ersetzt werden.

27.3.7.2 Schleife/Steuerung

Weitere Informationen zu diesen Eigenschaften finden Sie im Abschnitt „GMPID“.

27.3.7.3 Schleife/Alarmer

Weitere Informationen zu diesen Eigenschaften finden Sie im Abschnitt „GMPID“.

27.3.7.4 Schleife/Leistung

Weitere Informationen zu diesen Eigenschaften finden Sie im Abschnitt „GMPID“. Beachten Sie auch die obigen Anmerkungen zu Heat/Cool (Heizen/Kühlen) und Reverse/Direct (Rückwärts/Direkt). Wo der Begriff `Heat` bzw. `Cool` in einem Eigenschaftsnamen vorkommt, muss er durch `Rev` bzw. `Dir` ersetzt werden.

27.3.7.5 Schleife/ScaleInput1

Daten	Beschreibung	Bereich	Zugriff
DispLo1	Eingang 1 – Wert für „Display Low“ (Anzeige niedrig)	+/-30.000	R/W
DispHi1	Eingang 1 – Wert für „Display High“ (Anzeige hoch)	+/-30.000	R/W
SigLoKey1	Eingang 1 – konfigurierter Wert für „Signal Low“ (Signal niedrig)	+/-30.000	R/W
SigHiKey1	Eingang 1 – konfigurierter Wert für „Signal High“ (Signal hoch)	+/-30.000	R/W
SigLoApp1	Eingang 1 – angewendeter Wert für „Signal Low“ (Signal niedrig)	+/-30.000	R
SigHiApp1	Eingang 1 – angewendeter Wert für „Signal High“ (Signal hoch)	+/-30.000	R
StoreSigLo1	„Signal Low“ (Signal niedrig) für Eingang 1 speichern Auf der positiven Flanke wird das Millivolt-Signal, das auf Eingang 1 angewendet wird, als SigLoApp1 gespeichert.	0 oder 1 (Bit)	R/W
StoreSigHi1	„Signal Low“ (Signal niedrig) für Eingang 1 speichern Auf der positiven Flanke wird das Millivolt-Signal, das auf Eingang 1 angewendet wird, als SigHiApp1 gespeichert.	0 oder 1 (Bit)	R/W
SelectScaling1	Angewendete Signale für Eingang 1 auswählen Bei der Einstellung 1 sind die angewendeten Signalwerte aktiv. Bei der Einstellung 0 sind die konfigurieren Signalwerte aktiv.	0 oder 1 (Bit)	R/W

27.3.7.6 Schleife/ScaleInput2

Die für ScaleInput2 verfügbaren Daten sind mit den für ScaleInput1 verfügbaren Daten identisch.

27.3.7.7 Schleife/PeakValley

Daten	Beschreibung	Bereich	Zugriff
PVPeak	Der höchste PV-Wert seit dem letzten Zurücksetzen des Spitzenwerts (Peak).	+/-30.000	R
PVvall	Der kleinste PV-Wert seit dem letzten Zurücksetzen des Minimalwerts (Valley).	+/-30.000	R
ResetPkVall	Spitzenwert und Minimalwert zurücksetzen Bei der Einstellung 1 werden die Spitzenwert- und Minimalwert-Register auf den vorhandenen PV-Wert gesetzt. Bei der Einstellung 0 erfassen die Spitzenwert- und Minimalwert-Register PV-Extremwerte.	0 oder 1 (Bit)	R/W
ResetPeak	Spitzenwert zurücksetzen Bei der Einstellung 1 werden die Spitzenwert-Register auf den vorhandenen PV-Wert gesetzt. Bei der Einstellung 0 erfassen die Spitzenwert-Register PV-Extremwerte.	0 oder 1 (Bit)	R/W
ResetVall	Minimalwert zurücksetzen. Bei der Einstellung 1 werden die Minimalwert-Register auf den vorhandenen PV-Wert gesetzt. Bei der Einstellung 0 erfassen die Minimalwert-Register PV-Extremwerte.	0 oder 1 (Bit)	R/W

27.3.7.8 Schleife/Tara

Daten	Beschreibung	Bereich	Zugriff
PVGross	PV-Bruttowert aus PV-Zuweisungsberechnung vor Tara	+/-30.000	R
Inp1Gross	Bruttowert von Eingang 1 aus Eingangsskalierung vor Tara	+/-30.000	R
Inp2Gross	Bruttowert von Eingang 2 aus Eingangsskalierung vor Tara	+/-30.000	R
PVTareTot	PV-Tara-Gesamtwert Die Summe der tarierten PV-Werte seit der letzten Zurücksetzung des PV-Tara-Gesamtwerts.	+/-30.000	R
Inp1TareTot	Tara-Gesamtwert von Eingang 1 Die Summe der tarierten Werte von Eingang 1 seit der letzten Zurücksetzung des Tara-Gesamtwerts von Eingang 1.	+/-30.000	R
Inp2TareTot	Tara-Gesamtwert von Eingang 2 Die Summe der tarierten Werte von Eingang 2 seit der letzten Zurücksetzung des Tara-Gesamtwerts von Eingang 2.	+/-30.000	R
TarePV	Tara von Prozesswert Beim Übergang von 0 auf 1 wird der PV-Wert auf 0 tariert. Der tarierte Wert wird zum PV-Tara-Gesamtwert addiert.	0 oder 1 (Bit)	R/W
TareInp1	Tara Eingang 1 Beim Übergang von 0 auf 1 wird der Wert von Eingang 1 auf 0 tariert. Der tarierte Wert wird zum Tara-Gesamtwert von Eingang 1 addiert.	0 oder 1 (Bit)	R/W
TareInp2	Tara Eingang 2 Beim Übergang von 0 auf 1 wird der Wert von Eingang 2 auf 0 tariert. Der tarierte Wert wird zum Tara-Gesamtwert von Eingang 2 addiert.	0 oder 1 (Bit)	R/W

RstPVTareTot	PV-Tara-Gesamtwert zurücksetzen Beim Übergang von 0 auf 1 wird der PV-Tara-Gesamtwert auf 0 zurückgesetzt.	0 oder 1 (Bit)	R/W
RstIn1TareTot	Tara-Gesamtwert von Eingang 1 zurücksetzen Beim Übergang von 0 auf 1 wird der Tara-Gesamtwert von Eingang 1 auf 0 zurückgesetzt.	0 oder 1 (Bit)	R/W
RstIn2TareTot	Tara-Gesamtwert von Eingang 2 zurücksetzen Beim Übergang von 0 auf 1 wird der Tara-Gesamtwert von Eingang 2 auf 0 zurückgesetzt.	0 oder 1 (Bit)	R/W

27.3.7.9 Ausgänge/Zykluszeiten

Weitere Informationen zu diesen Eigenschaften finden Sie im Abschnitt „GMPID“.

27.3.7.10 Ausgänge/Externe Daten

Weitere Informationen zu diesen Eigenschaften finden Sie im Abschnitt „GMPID“.

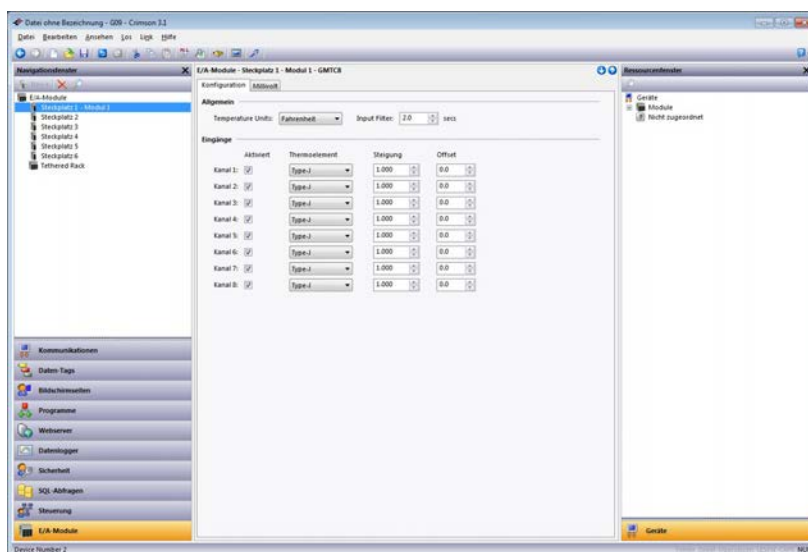
27.3.7.11 Ausgänge/Informationen

Weitere Informationen zu diesen Eigenschaften finden Sie im Abschnitt „GMPID“.

27.4 Die Temperaturmodule GMTC und GMRTD

Die Parameter der Module GMTC und GMRTD sind in Gruppen unterteilt, für die es jeweils eine eigene Seite gibt.

27.4.1 Konfigurationseigenschaften



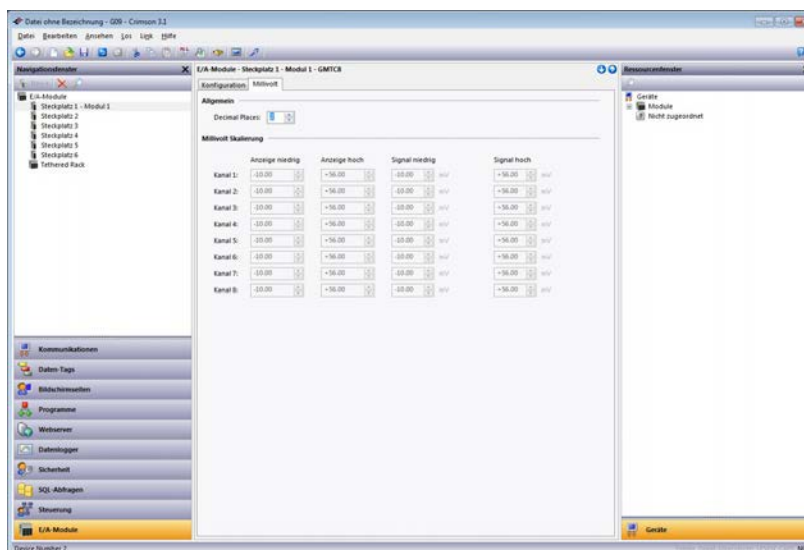
27.4.1.1 Allgemein

- *Temperature Units* (Temperatureinheiten) dient zur Auswahl zwischen den Temperaturskalen Kelvin, Fahrenheit und Celsius.
- Der *Input Filter* (EingangsfILTER) ist eine Zeitkonstante zur Stabilisierung schwankender Eingangssignale.

27.4.1.2 Eingänge

- Mit der Eigenschaft *Enabled* (Aktiviert) können nicht verwendete Eingänge deaktiviert werden, damit Lesevorgänge an den übrigen Eingängen schneller stattfinden. Weitere Informationen zu den daraus resultierenden Aktualisierungshäufigkeiten finden Sie im Bulletin zur Modul-Hardware.
- Die Eigenschaft *Thermocouple* (Thermoelement) oder *RTD* dient zur Auswahl des verwendeten Sensorstandards. Welche Optionen verfügbar sind, richtet sich nach dem Modultyp. Die Option „Millivolt“ ist bei GMTC-Modulen verfügbar. Sie gibt an, dass der Eingang als linear skaliertes Spannungseingang verwendet werden soll anstatt als Thermoelement-Eingang.
- Mit den Eigenschaften *Slope* (Steigung) und *Offset* (Offset) kann der PV-Wert angepasst oder neu skaliert werden, um einen Fehler im angeschlossenen Sensor auszugleichen. Sie können auch die Korrektur des PV-Werts in Anwendungen ermöglichen, in denen der Sensor den Prozess nicht direkt misst, wodurch ein fester oder variabler Offset entsteht. Der oben stehende Abschnitt „GMPID“ enthält ein Berechnungsbeispiel für das Konfigurieren dieser Eigenschaften.

27.4.2 Millivolt – Eigenschaften (nur bei GMTC)



27.4.2.1 Allgemein

- Die Eigenschaft *Decimal Places* (Dezimalstellen) wird verwendet, um Crimson die Anzeige der Engineering Units (Bedieneinheiten) in der richtigen Auflösung zu ermöglichen. Diese Eigenschaft dient lediglich dazu, in der gesamten Software die richtige Auflösung anzuzeigen, und wird nicht im Modul verwendet.

27.4.2.2 Millivolt Scaling (Millivolt-Skalierung)

- Bei Eingängen, die auf der Registerkarte „Configuration“ (Konfiguration) auf den Modus „Millivolt“ gesetzt wurden, werden die Eigenschaften *Display Low* (Anzeige niedrig) und *Display High* (Anzeige hoch) zusammen mit den Eigenschaften *Signal Low* (Signal niedrig) und *Signal High* (Signal hoch) verwendet, um die Eingangsspannung auf geeignete Anzeigeeinheiten zu skalieren. Angenommen, eine Anwendung verwendet einen Sensor, der für einen Druck von 0 bis 1000 psi eine Spannung von 0 bis 42,5 mV erzeugt. Geben Sie für *Display Low* (Anzeige niedrig) 0 und für *Display High* (Anzeige hoch) 1000 ein, um den Anzeigebereich einzustellen. Geben Sie dann für *Signal Low* (Signal niedrig) 0 und für *Signal High* (Signal hoch) 42,5 ein, um den entsprechenden Eingangsbereich einzustellen.

27.4.3 Verfügbare Daten

Die nachfolgenden Tabellen enthalten die Moduldatenwerte, die dem Graphite-Host zur Verfügung stehen.

27.4.3.1 Eingang/Status

Daten	Beschreibung	Bereich	Zugriff
PV1-8	Prozesswert nach Berechnung von Steigung und Offset	Pro Sensor	R
ColdJunc	Kalibrierwert der Vergleichsstelle	n. z.	R
InputAlarm1-8	Indikator für Eingang außerhalb des zulässigen Bereichs	0 oder 1 (Bit)	R

27.4.3.2 Eingang/Steuerung

Daten	Beschreibung	Bereich	Zugriff
InputFilter	EingangsfILTER	0 – 60,0 Sek	R/W
InputOffset1-8	Zum PV addierter Offset-Wert	Pro Sensor	R/W
InputSlope1-8	Auf PV angewendeter Steigungswert	0,001 – 10,000	R/W

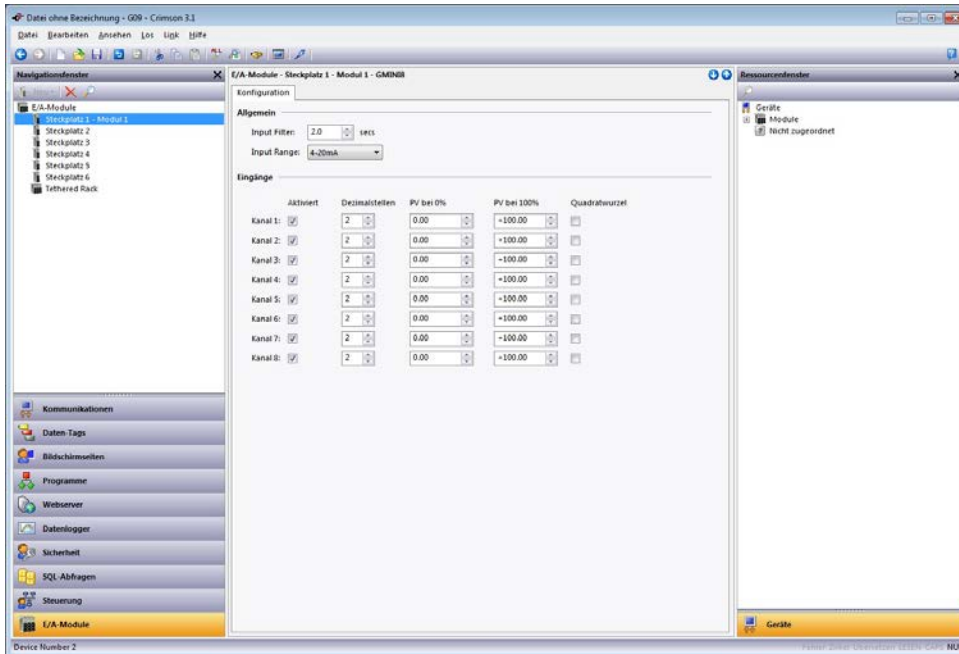
27.4.3.3 Eingang/ScaleMvInputs

Daten	Beschreibung	Bereich	Zugriff
DispLo1-8	„Display Low“ (Anzeige niedrig) 1-8	+/-30.000	R/W
DispHi1-8	„Display High“ (Anzeige hoch) 1-8	+/-30.000	R/W
SigLo1-8	„Signal Low“ (Signal niedrig) 1-8	-10,00 – +56,00	R/W
SigHi1-8	„Signal High“ (Signal hoch) 1-8	-10,00 – +56,00	R/W

27.5 Die Analogeingangsmodule GMINI und GMINV

Die Parameter für GMINI und GMINV werden auf einer einzelnen Seite konfiguriert.

27.5.1 Konfigurationseigenschaften



27.5.1.1 Allgemein

- Der *Input Filter* (Eingangsfilter) ist eine Zeitkonstante zur Stabilisierung schwankender Eingangssignale.
- Unter *Input Range* (Eingangsbereich) stehen verschiedene Eingangsbereiche zur Auswahl.

27.5.1.2 Eingänge

Mit diesen Einstellungen können die einzelnen Eingänge unabhängig voneinander konfiguriert werden.

- Mit der Eigenschaft *Enabled* (Aktiviert) können nicht verwendete Eingänge deaktiviert werden, damit Lesevorgänge an den übrigen Eingängen schneller stattfinden. Weitere Informationen zu den Lesegeschwindigkeiten finden Sie im Hardware-Bulletin.
- Die Eigenschaft *Decimals* (Dezimalstellen) wird verwendet, um Crimson die Anzeige der Engineering Units (Bedieneinheiten) in der richtigen Auflösung zu ermöglichen. Diese Eigenschaft dient lediglich dazu, in der gesamten Software die richtige Auflösung anzuzeigen, und wird nicht im Modul verwendet.
- Die Eigenschaften *PV at 0%* (PV bei 0 %) und *PV at 100%* (PV bei 100 %) werden zur Skalierung von DC-Eingangssignalen verwendet. Geben Sie die gewünschte PV-Messung für die minimalen und maximalen Eingangssignalpegel ein. Wenn die Anwendung z. B. einen Eingang von einem Durchflusssensor mit einem Ausgang von 4 bis 20 mA akzeptiert, der 5 bis 105 Gallonen pro Minute darstellt, wählen Sie unter *Input Type* (Eingangstyp) die Option „Process 420mA“ aus, geben bei *PV at 0%* (PV bei 0 %) den Wert „5“ ein und bei *PV at 100%* (PV bei 100 %) den Wert „105“.

- Mit der Eigenschaft *Square Root* (Quadratwurzel) kann das Gerät in Anwendungen eingesetzt werden, in denen das gemessene Signal das Quadrat des PV ist. Dies ist nützlich bei Anwendungen wie der Messung des Durchflusses mit einem Differenzdruckaufnehmer.

27.5.2 Verfügbare Daten

Die nachfolgenden Tabellen enthalten die Moduldatenwerte, die dem Graphite-Host zur Verfügung stehen.

27.5.2.1 Eingang/Status

Daten	Beschreibung	Bereich	Zugriff
PV1-8	Prozesswert – skaliert nach Kanaleinstellungen	Pro Sensor	R
InputAlarm1-8	Indikator für Eingang außerhalb des zulässigen Bereichs	0 oder 1 (Bit)	R

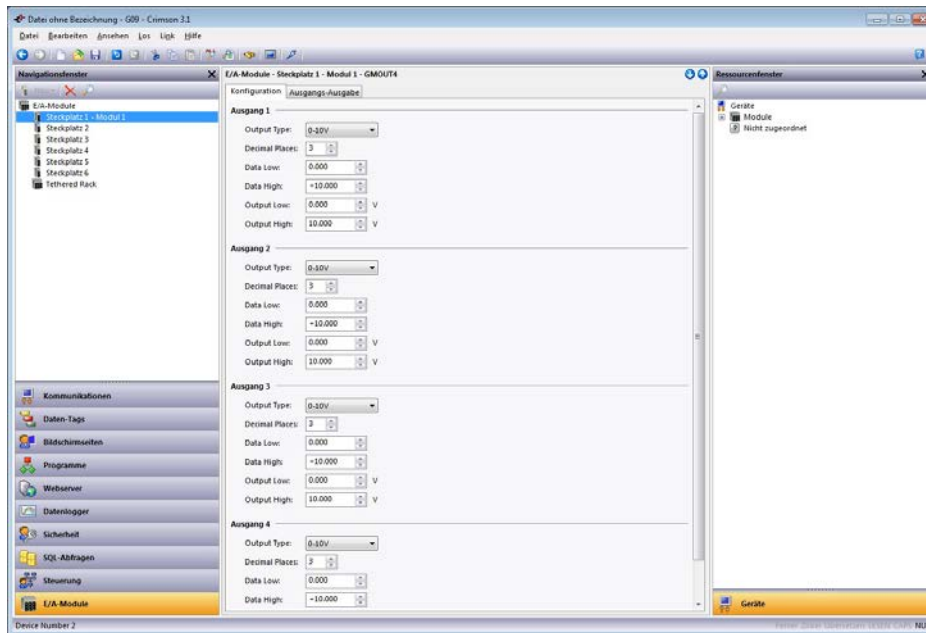
27.5.2.2 Eingang/Steuerung

Daten	Beschreibung	Bereich	Zugriff
InputFilter	Eingangsfiler	0 – 60,0 Sek	R/W
ProcessMin1-8	Gewünschter PV bei minimalem Eingangssignalpegel	+/- 30,000	R/W
ProcessMax1-8	Gewünschter PV bei maximalem Eingangssignalpegel	+/- 30,000	R/W

27.6 Das Analogausgangsmodul GMOUT

Die Parameter des GMOUT-Moduls sind in zwei Gruppen unterteilt, für die es jeweils eine eigene Seite gibt.

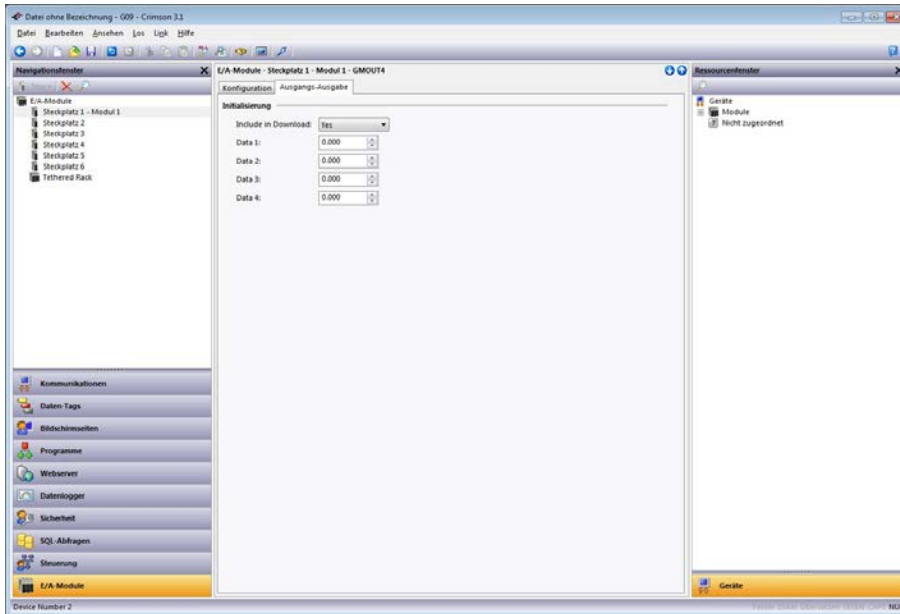
27.6.1 Konfigurationseigenschaften



27.6.1.1 Ausgang 1 bis Ausgang 4

- Mit der Eigenschaft *Output Type* (Ausgangstyp) wird zwischen Ausgängen mit 0-5 V, 0-10 V, +/-10 V, 0-20 mA oder 4-20 mA gewählt.
- Die Eigenschaft *Decimal Places* (Dezimalstellen) wird verwendet, um Crimson die Anzeige der Datenwerte in der richtigen Auflösung zu ermöglichen. Diese Eigenschaft dient lediglich dazu, in Crimson die richtige Auflösung anzuzeigen, und wird nicht im Modul verwendet.
- Die Eigenschaften *Data Low* (Daten niedrig) und *Data High* (Daten hoch) werden in Verbindung mit den Eigenschaften *Output Low* (Ausgang niedrig) und *Output High* (Ausgang hoch) verwendet, um die Leistung anhand der Maßeinheit des RAW-Datenwerts zu skalieren. Wenn die Anwendung beispielsweise einen Datenwert von 0,0 bis 500,0 umfasst, der einen Ausgang von 1,000 bis 5,000 V bereitstellt, geben Sie für die Eigenschaft *Data Low* (Daten niedrig) 0 und für die Eigenschaft *Data High* (Daten hoch) 500 ein, um den Datenbereich festzulegen, und geben dann für „Output Low“ (Ausgang niedrig) 1 V und für „Output High“ (Ausgang hoch) 5 V ein, um den Ausgangsbereich festzulegen.

27.6.2 Anfänglicher Ausgang – Eigenschaften

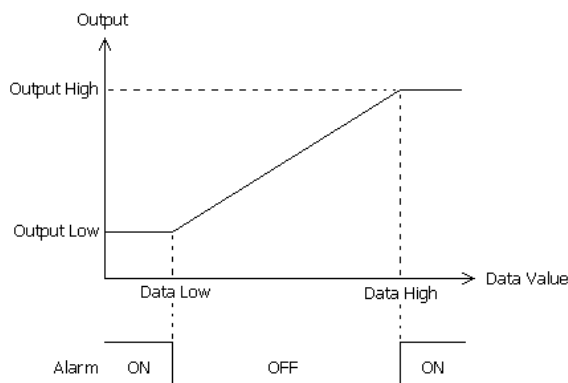


27.6.2.1 Initialisierung

- Mit der Eigenschaft *Include in Download* (In Download einbeziehen) wird bestimmt, ob die Initialisierungswerte auf das Modul heruntergeladen werden. Bei Auswahl von „No“ (Nein) können Datenbanken nach Wunsch geändert und heruntergeladen werden, ohne die Werte versehentlich zu überschreiben.
- Die Eigenschaften *Data 1* (Daten 1) bis *Data 4* (Daten 4) sind die anfänglichen Ausgangswerte, die jeweils für die Ausgänge 1 bis 4 verwendet werden. Diese Werte werden anhand der auf der vorherigen Seite beschriebenen Eigenschaften „Data Low“ (Daten niedrig), „Data High“ (Daten hoch), „Output Low“ (Ausgabe niedrig) und „Output High“ (Ausgabe hoch) skaliert.

27.6.2.2 Ausgangsaktion

Der Ausgang wird so gesteuert, dass die Werte an *Output Low* (Ausgang niedrig) und *Output High* (Ausgang hoch) „festgeklammert“ werden. Wenn also ein Datenwert in das Modul geschrieben wird, der den Wert *Data Low* (Daten niedrig) unterschreitet, bleibt der Ausgang bei *Output Low* (Ausgang niedrig) und der Alarm wird eingeschaltet. Wenn ein Datenwert in verwendet wird, der den Wert *Data High* (Daten hoch) überschreitet, geht der Ausgang zu *Output Low* und wieder wird der Alarm eingeschaltet.



27.6.3 Verfügbare Daten

Die nachfolgenden Tabellen enthalten die Moduldatenwerte, die dem Graphite-Host zur Verfügung stehen.

27.6.3.1 Daten

Daten	Beschreibung	Bereich	Zugriff
Data1	Skalierter Wert für Ausgang 1	DataLo1 bis DataHi1	R/W
Data2	Skalierter Wert für Ausgang 2	DataLo2 bis DataHi2	R/W
Data3	Skalierter Wert für Ausgang 3	DataLo3 bis DataHi3	R/W
Data4	Skalierter Wert für Ausgang 4	DataLo4 bis DataHi4	R/W

27.6.3.2 Skalierung

Daten	Beschreibung	Bereich	Zugriff
DataLo1	Ausgang 1 – Wert für „Data Low“ (Daten niedrig)	+/-30.000	R/W
DataLo2	Ausgang 2 – Wert für „Data Low“ (Daten niedrig)	+/-30.000	R/W
DataLo3	Ausgang 3 – Wert für „Data Low“ (Daten niedrig)	+/-30.000	R/W
DataLo4	Ausgang 4 – Wert für „Data Low“ (Daten niedrig)	+/-30.000	R/W
DataHi1	Ausgang 1 – Wert für „Data High“ (Daten hoch)	+/-30.000	R/W
DataHi2	Ausgang 2 – Wert für „Data High“ (Daten hoch)	+/-30.000	R/W
DataHi3	Ausgang 3 – Wert für „Data High“ (Daten hoch)	+/-30.000	R/W
DataHi4	Ausgang 4 – Wert für „Data High“ (Daten hoch)	+/-30.000	R/W
OutputLo1	Ausgang 1 – Wert für „Output Low“ (Ausgang niedrig)	Pro Modus	R/W
OutputLo2	Ausgang 2 – Wert für „Output Low“ (Ausgang niedrig)	Pro Modus	R/W
OutputLo3	Ausgang 3 – Wert für „Output Low“ (Ausgang niedrig)	Pro Modus	R/W
OutputLo4	Ausgang 4 – Wert für „Output Low“ (Ausgang niedrig)	Pro Modus	R/W
OutputHi1	Ausgang 1 – Wert für „Output High“ (Ausgang hoch)	Pro Modus	R/W
OutputHi2	Ausgang 2 – Wert für „Output High“ (Ausgang hoch)	Pro Modus	R/W
OutputHi3	Ausgang 3 – Wert für „Output High“ (Ausgang hoch)	Pro Modus	R/W
OutputHi4	Ausgang 4 – Wert für „Output High“ (Ausgang hoch)	Pro Modus	R/W

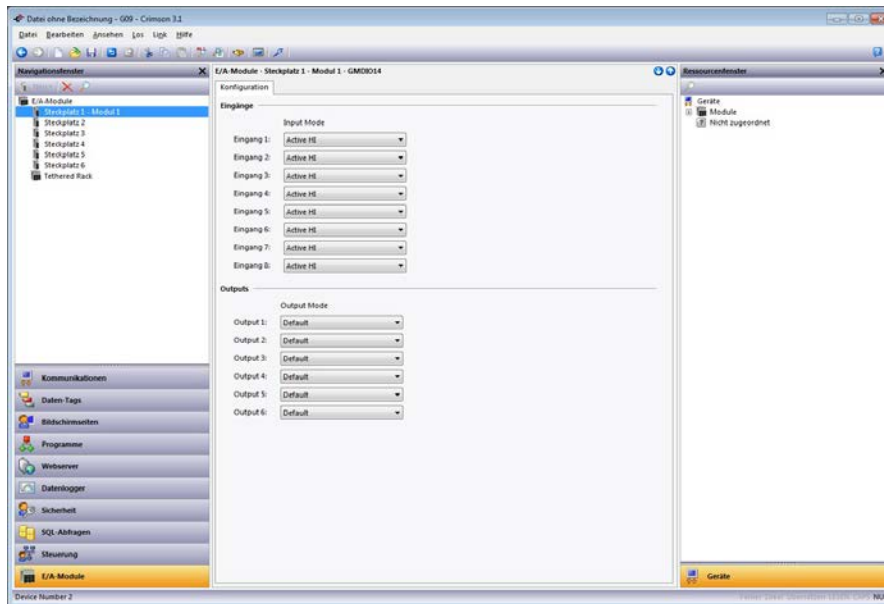
27.6.3.3 Alarme

Daten	Beschreibung	Bereich	Zugriff
Alarm1	Ausgang 1 Alarmzustand	0 oder 1 (Bit)	R
Alarm2	Ausgang 2 Alarmzustand	0 oder 1 (Bit)	R
Alarm3	Ausgang 3 Alarmzustand	0 oder 1 (Bit)	R
Alarm4	Ausgang 4 Alarmzustand	0 oder 1 (Bit)	R

27.7 Das digitale E/A-Modul GMDIO

Die Parameter für GMDIO befinden sich auf einer einzelnen Seite.

27.7.1 Konfigurationseigenschaften



- Die Eingangsmodus Eigenschaft (Modus) legt fest, ob für die einzelnen Eingangskanäle „Hoch“ oder „Niedrig“ als aktiv angesehen werden soll. Beachten Sie, dass der stromliefernde („Sourcing“) bzw. stromsenkende („Sinking“) Modus eines Eingangs über Hardwareverknüpfungen konfiguriert wird. Weitere Informationen finden Sie im Produkt-Bulletin.
- Die Eigenschaft *Output Mode* (Ausgabemodus) bestimmt den Ausgabezustand für jeden Kanal beim Systemstart. Verwenden Sie „Default“ (Standard), um die Ausgabe in den zuletzt bekannten Zustand zu versetzen. Verwenden Sie „Active LO“ oder „Active Hi“, um für die Ausgabe den angegebenen Zustand zu erzwingen.

27.7.2 Verfügbare Daten

Die nachfolgenden Tabellen enthalten die Moduldatenwerte, die dem Graphite-Host zur Verfügung stehen.

27.7.2.1 Variablen/Eingänge

Daten	Beschreibung	Bereich	Zugriff
Input1-8	Eingangsstatus	0 oder 1 (Bit)	R

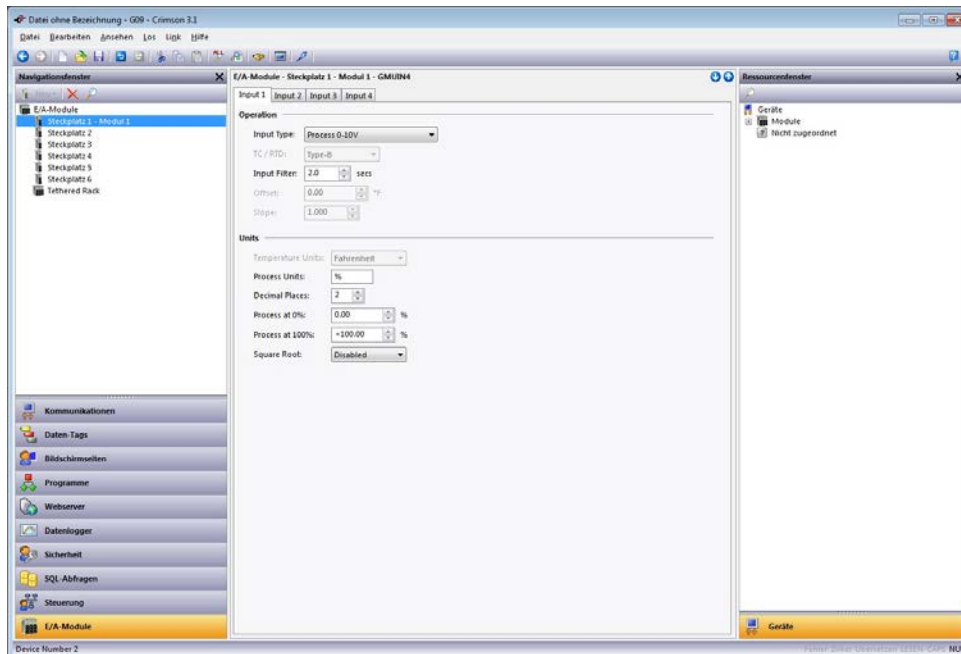
27.7.2.2 Variablen/Ausgänge

Daten	Beschreibung	Bereich	Zugriff
Output1-6	Ausgangsstatus	0 oder 1 (Bit)	R/W

27.8 Das universelle Eingangsmodul GMUIN

Die Parameter des GMUIN-Moduls stehen für jeden der vier Kanäle auf einer eigenen Seite.

27.8.1 Eingangseigenschaften



27.8.1.1 Vorgang

- Die Eigenschaft *Input Type* (Eingangstyp) dient zur Auswahl der folgenden Eingänge: 0-10 V, 0-50 mV, 0-20 mA, 4-20 mA, RTD oder Thermocouple (Thermoelement).
- Mit der Eigenschaft *TC/RTD* (Thermoelement/RTD) wird der verwendete Sensorstandard ausgewählt.
- Der *Input Filter* (Eingangsfiler) ist eine Zeitkonstante zur Stabilisierung schwankender Eingangssignale.
- Mit den Eigenschaften *Slope* (Steigung) und *Offset* (Offset) kann der PV-Wert angepasst oder neu skaliert werden, um einen Fehler im angeschlossenen Sensor auszugleichen. Sie können auch die Korrektur des PV-Werts in Anwendungen ermöglichen, in denen der Sensor den Prozess nicht direkt misst, wodurch ein fester oder variabler Offset entsteht. Der oben stehende Abschnitt „GMPID“ enthält ein Berechnungsbeispiel für das Konfigurieren dieser Eigenschaften.

27.8.1.2 Einheiten

- *Temperature Units* (Temperatureinheiten) dient zur Auswahl zwischen Kelvin, Fahrenheit und Celsius.
- Mit der Eigenschaft *Process Units* (Prozesseinheiten) können Sie die Engineering Units (Bedienereinheiten) für den Prozess eingeben. Die Eigenschaft *Decimal Places* (Dezimalstellen) wird verwendet, um Crimson die Anzeige der Engineering Units (Bedienereinheiten) in der richtigen Auflösung zu ermöglichen. Diese werden nur zum Identifizieren der entsprechenden Felder in der Software verwendet. Die Parameter werden als Teil der Crimson-Datei gespeichert, aber nicht innerhalb des Moduls verwendet.
- Mit den Eigenschaften *Process at 0%* (Prozess bei 0 %) und *Process at 100%* (Prozess bei 100 %) werden DC-Eingangssignale skaliert. Geben Sie die gewünschte PV-Messung für die minimalen und

maximalen Eingangssignalpegel ein. Wenn die Anwendung z. B. einen Eingang von einem Durchflusssensor mit einem Leistungsausgang von 4 bis 20 mA akzeptiert, der 5 bis 105 Gallonen pro Minute darstellt, wählen Sie unter *Input Type* (Eingangstyp) die Option „Process 420mA“, geben Sie unter Process at 0% (Prozess bei 0 %) den Wert „5“ ein und unter Process at 100% (Prozess bei 100 %) den Wert „105“.

- Mit der Eigenschaft *Square Root* (Quadratwurzel) kann das Gerät in Anwendungen eingesetzt werden, in denen das gemessene Signal das Quadrat des PV ist. Dies ist nützlich bei Anwendungen wie der Messung des Durchflusses mit einem Differenzdruckaufnehmer.

27.8.2 Verfügbare Daten

Die nachfolgenden Tabellen enthalten die Moduldatenwerte, die dem Graphite-Host zur Verfügung stehen.

27.8.2.1 Eingang/Status

Daten	Beschreibung	Bereich	Zugriff
PV1-4	Prozesswert, nach Berechnung von Steigung und Offset	Pro Sensor	R
ColdJunc1-4	Kalibrierwert der Vergleichsstelle	n. z.	R
InputAlarm1-4	Indikator für Eingang außerhalb des zulässigen Bereichs	0 oder 1 (Bit)	R

27.8.2.2 Eingang/Steuerung

Daten	Beschreibung	Bereich	Zugriff
InputFilter1-4	Eingangsfilter	0 – 60,0 Sek	R/W
InputOffset1-4	Zum PV addierter Offset-Wert	Pro Sensor	R/W
InputSlope1-4	Auf PV angewendeter Steigungswert	0,001 – 10,000	R/W