

---

# ASCET V5.2

Benutzerhandbuch

## Copyright

---

Die Angaben in diesem Schriftstück dürfen nicht ohne gesonderte Mitteilung der ETAS GmbH geändert werden. Desweiteren geht die ETAS GmbH mit diesem Schriftstück keine weiteren Verpflichtungen ein. Die darin dargestellte Software wird auf Basis eines allgemeinen Lizenzvertrages oder einer Einzellizenz geliefert. Benutzung und Vervielfältigung ist nur in Übereinstimmung mit den vertraglichen Abmachungen gestattet.

Unter keinen Umständen darf ein Teil dieser Veröffentlichung in irgendeiner Form ohne schriftliche Genehmigung der ETAS GmbH kopiert, vervielfältigt, in einem Retrievalsystem gespeichert oder in eine andere Sprache übersetzt werden.

© **Copyright 2007** ETAS GmbH, Stuttgart

Die verwendeten Bezeichnungen und Namen sind Warenzeichen oder Handelsnamen ihrer entsprechenden Eigentümer.

Der Name INTECRIO ist ein eingetragenes Warenzeichen der ETAS GmbH.

Dokument EC010001 R5.2.2 DE

---

# Inhalt

<b>1</b>	Einführung	13
<b>1.1</b>	Typischer Arbeitsablauf	13
<b>1.2</b>	Programmstart	14
<b>2</b>	Der Komponentenmanager	15
<b>2.1</b>	Komponentenmanager – Benutzeroberfläche	15
<b>2.1.1</b>	Allgemeine Beschreibung	15
<b>2.1.2</b>	Benutzeroberfläche Komponentenmanager	15
	Die Fensterelemente	16
	Bedienelemente in den Schaltflächenleisten	17
	Symbolgrafiken für Datenbankobjekte	18
	Die Menüfunktionen	19
	Kontextsensitive Menüfunktionen	27
<b>2.1.3</b>	Ansichtskonzepte im Komponentenmanager	29
<b>2.1.4</b>	ETAS Support-Funktion „Problem Report“	34
<b>2.2</b>	Komponentenmanager – Einrichten von ASCET	37
<b>2.2.1</b>	Grundlegende Optionen	42
<b>2.2.2</b>	Optionen zum Erscheinungsbild	43
	Optionen für Bestätigungsfenster	47
<b>2.2.3</b>	Build-Optionen	50

2.2.4	Standardoptionen. . . . .	52
	Implementierungsoptionen. . . . .	53
2.2.5	Optionen für die automatische Dokumentation . . . . .	55
2.2.6	Optionen für Editoren. . . . .	55
	Optionen für Blockdiagramme . . . . .	56
	Optionen für Texteditoren. . . . .	59
	Optionen für Tabelleneditoren . . . . .	59
	Optionen für Zustandsautomaten. . . . .	59
	Optionen für Verstellfenster . . . . .	60
	Optionen für Messfenster. . . . .	60
2.2.7	Experiment-Optionen . . . . .	60
2.2.8	Optionen für externe Werkzeuge . . . . .	61
	Optionen für den ASCII Editor . . . . .	61
	Compiler-Optionen. . . . .	62
2.2.9	Optionen für die Integration. . . . .	63
	Exportoptionen. . . . .	63
	Importoptionen . . . . .	65
	Lizenzierungsoptionen . . . . .	67
	Optionen für den Datenaustausch . . . . .	68
	Optionen für ausführbare Dateien . . . . .	69
2.2.10	Externe Optionen . . . . .	69
2.2.11	Arbeiten mit Benutzerprofilen. . . . .	73
	Verwenden der Benutzerauswahl . . . . .	73
2.3	Komponentenmanager – Daten verwalten . . . . .	76
	Datenbankeinträge. . . . .	77
2.3.1	Datenbankeinträge verwalten. . . . .	78
2.3.2	Arbeiten in den Ansichtskonzepten . . . . .	86
	Bearbeiten von Komponenten und Projekten im Komponentenmanager. . . . .	86
	Auswahl eines weiteren Daten- oder Implementierungs- satzes. . . . .	93
2.3.3	Exportieren von Ordnern und Datenbankeinträgen. . . . .	94
	Binärer Export. . . . .	94
	AMD-Export. . . . .	94
	Sonstige Exportformate . . . . .	95
	Export durchführen . . . . .	96
2.3.4	Importieren von Ordnern und Datenbankeinträgen. . . . .	97
	Import durchführen . . . . .	99
	Importieren eines Verzeichnisinhalts . . . . .	102
	Besonderheiten beim AMD-Import . . . . .	103

	Importieren von Projekten . . . . .	108
	Importieren von Objekten aus anderen ASCET-Versionen . . . . .	108
<b>2.3.5</b>	Arbeiten mit Datenbankeinträgen. . . . .	109
	Referenzen auf Einträge . . . . .	109
	Bearbeiten . . . . .	116
	Suchen und Ersetzen in C-Code- und ESDL-Komponenten . . . . .	118
<b>2.3.6</b>	Datenbanken verwalten . . . . .	125
	Basisaufgaben . . . . .	125
	Datenbanken aus früheren ASCET-Versionen verwenden . . . . .	133
	ANSI C-Umwandlung . . . . .	137
	Auflösen von Namenskonflikten . . . . .	138
	Datenbank-Service-routinen. . . . .	138
	Generierter Code . . . . .	139
	Durchsuchen einer Datenbank . . . . .	139
<b>2.3.7</b>	Datenbankzugriff . . . . .	146
<b>2.4</b>	Das ASCET-Monitorfenster . . . . .	149
	Benutzeroberfläche . . . . .	150
<b>2.4.1</b>	Register „Monitor“ . . . . .	153
<b>2.4.2</b>	Register „Build“ . . . . .	156
	Meldungen im Register „Build“ konfigurieren . . . . .	159
	Meldungen im Fenster „CodeGen Message Configuration“ konfigurieren . . . . .	162
<b>3</b>	Einfügen benutzerdefinierter Funktionen . . . . .	169
<b>3.1</b>	Menüfunktionen definieren . . . . .	170
<b>3.2</b>	Autostartaktion definieren . . . . .	174
<b>3.3</b>	Abschlussaktion definieren . . . . .	175
<b>3.4</b>	Struktur der Skriptdateien . . . . .	175
<b>3.4.1</b>	EXECUTE . . . . .	175
<b>3.4.2</b>	NOWAIT . . . . .	176
<b>3.4.3</b>	OBJECT . . . . .	177
<b>3.4.4</b>	SELECTOR . . . . .	178
<b>3.4.5</b>	MENUITEM . . . . .	182
<b>3.4.6</b>	FILE . . . . .	183
<b>3.4.7</b>	FORK . . . . .	183
<b>3.4.8</b>	SEND . . . . .	183
<b>3.4.9</b>	POST . . . . .	184
<b>4</b>	Spezifikation von Komponenten und Projekten. . . . .	185
<b>4.1</b>	Der Blockdiagrammeditor . . . . .	186
<b>4.1.1</b>	Die Benutzeroberfläche des Blockdiagrammeditors. . . . .	188

	Beschreibung der Menüfunktionen. . . . .	193
<b>4.1.2</b>	Definieren der Schnittstelle einer Komponente . . . . .	201
	Klassen. . . . .	201
	Module . . . . .	208
<b>4.1.3</b>	Komplexe Typen als Schnittstellenelemente. . . . .	208
	Arrays und Matrizen. . . . .	209
	Kennlinien und Kennfelder. . . . .	214
<b>4.1.4</b>	Erzeugen von Blockdiagrammen. . . . .	217
	Elemente, Operatoren und Verbindungen. . . . .	217
	Arrays, Matrizen, Kennlinien und -felder. . . . .	222
	Kontrollflussanweisungen. . . . .	225
	Komponenten als komplexe Elemente . . . . .	229
	Kommentare und Notizen . . . . .	231
<b>4.1.5</b>	Bearbeiten von Blockdiagrammen. . . . .	232
	Elemente betrachten . . . . .	233
	Elemente bearbeiten. . . . .	235
	Aussehen von Diagrammelementen . . . . .	237
	Layout von aufgenommenen Komponenten. . . . .	241
<b>4.1.6</b>	Sequenzaufrufe . . . . .	247
	Einzelne Sequenzaufrufe bearbeiten. . . . .	248
	Mehrere Sequenzaufrufe bearbeiten. . . . .	255
	Verbinder . . . . .	259
<b>4.1.7</b>	Implementation-Casts in Blockdiagrammen . . . . .	260
<b>4.1.8</b>	Grafische Hierarchien . . . . .	265
<b>4.1.9</b>	Navigieren zwischen Diagrammen . . . . .	270
	Komponenten mit mehreren Diagrammen . . . . .	270
	Navigieren zwischen Komponenten . . . . .	273
<b>4.1.10</b>	Analysieren von Komponenten. . . . .	274
<b>4.1.11</b>	Datenaustausch . . . . .	278
<b>4.1.12</b>	Verwendung des Blockdiagrammeditors. . . . .	280
	Speichern von Diagrammen . . . . .	280
	Betrachten und Drucken von Diagrammen . . . . .	280
	Verwendung von referenzierten Komponenten. . . . .	284
<b>4.2</b>	Der Zustandsautomaten-Editor . . . . .	285
	Spezielle Menüfunktionen . . . . .	286
<b>4.2.1</b>	Zeichnen des Zustandsdiagramms. . . . .	287
<b>4.2.2</b>	Hierarchiezustände. . . . .	296
	Geschlossener Hierarchiezustand . . . . .	299
	Offener Hierarchiezustand . . . . .	302
<b>4.2.3</b>	Spezifizieren von Bedingungen und Aktionen. . . . .	303

	Bedingungen und Aktionen in separaten Diagrammen . . . . .	305
	Bedingungen und Aktionen verwenden . . . . .	310
	Bedingungen und Aktionen im Zustandsdiagramm. . . . .	315
	Kommunikation mit anderen Komponenten . . . . .	321
<b>4.2.4</b>	Öffentliche Methoden . . . . .	323
<b>4.2.5</b>	Experimentieren mit Zustandsautomaten . . . . .	326
<b>4.3</b>	Der C-Code-Editor . . . . .	329
	Beschreibung der Menüfunktionen. . . . .	331
<b>4.3.1</b>	Spezifizieren von Komponenten in C-Code. . . . .	338
<b>4.3.2</b>	Externer Editor . . . . .	345
<b>4.3.3</b>	Verwendung des Editors für externe Quellen . . . . .	348
<b>4.4</b>	Der ESDL-Editor . . . . .	351
<b>4.4.1</b>	Spezifizieren von Klassen in ESDL . . . . .	352
<b>4.4.2</b>	Spezifizieren von Modulen in ESDL . . . . .	355
<b>4.4.3</b>	Analysieren von ESDL-Komponenten . . . . .	355
<b>4.5</b>	Spezifizieren zeitkontinuierlicher Blöcke . . . . .	356
<b>4.5.1</b>	Zeitkontinuierliche Blöcke als Blockdiagramme . . . . .	357
<b>4.5.2</b>	Zeitkontinuierliche Blöcke als C-Code . . . . .	359
<b>4.5.3</b>	Zeitkontinuierliche Blöcke in ESDL-Code . . . . .	360
<b>4.5.4</b>	Experimentieren mit zeitkontinuierlichen Blöcken . . . . .	361
	Die Zykluszeit überwachen . . . . .	365
<b>4.6</b>	Der Boolesche Tabelleneditor . . . . .	367
	Beschreibung der Menüfunktionen. . . . .	368
	Boolesche Tabellen spezifizieren . . . . .	371
<b>4.7</b>	Der Editor für Bedingungstabellen . . . . .	377
	Beschreibung der Menüfunktionen. . . . .	379
	Beschreibung der Schaltflächen . . . . .	384
<b>4.7.1</b>	Bedingungstabelle einrichten . . . . .	385
<b>4.7.2</b>	Bedingungstabelle spezifizieren . . . . .	393
<b>4.7.3</b>	Experimentieren mit Bedingungstabellen . . . . .	397
<b>4.8</b>	Der Projekteditor . . . . .	398
<b>4.8.1</b>	Standardprojekt für eine Komponente . . . . .	401
<b>4.8.2</b>	Beschreibung der Menüfunktionen. . . . .	402
	Register „Graphics“ . . . . .	408
	Register „OS“ . . . . .	409
	Register „Formulas“ . . . . .	411
	Register „Impl. Type“ . . . . .	412
	Register „Files“ . . . . .	413
<b>4.8.3</b>	Spezifizieren eines Projekts . . . . .	414
<b>4.8.4</b>	Definieren der globalen Kommunikation. . . . .	416

<b>4.8.5</b>	Definieren des Scheduling im Betriebssystemeditor . . . . .	419
	Einstellungen des Betriebssystems. . . . .	420
	Tasks und Prozesse . . . . .	421
	Überwachungsoption . . . . .	430
	Betriebsmodi . . . . .	431
<b>4.8.6</b>	Verwaltung externer Projektdateien . . . . .	432
<b>4.8.7</b>	Hybride Projekte . . . . .	436
<b>4.8.8</b>	Projekteinstellungen . . . . .	438
	Knoten „ASAM-2MC“ . . . . .	442
	Knoten „Build“ . . . . .	443
	Knoten „OS Configuration“ . . . . .	445
	Knoten „Code Generation“ . . . . .	446
	Knoten „Integer Arithmetic“ . . . . .	448
	Knoten „Experiment Code“ . . . . .	451
	Knoten „Production Code“ . . . . .	452
	Knoten „Optimization“ . . . . .	452
	Knoten „Statemachine“ . . . . .	453
<b>4.8.9</b>	Festlegen der Implementierung für die Festkomma- Arithmetik . . . . .	455
	Formeln . . . . .	456
	Globale Veränderungen in Implementierungen . . . . .	462
	Implementierungstypen . . . . .	463
<b>4.8.10</b>	Experimentieren mit Projekten . . . . .	469
	Online- und Offline-Experimentieren. . . . .	470
	Experimentieren mit quantisiertem Gleitkomma-Code . . . . .	473
<b>4.8.11</b>	Generieren von Applikationsdaten . . . . .	476
<b>4.9</b>	Container . . . . .	476
<b>4.9.1</b>	Arbeiten mit Containern. . . . .	477
<b>4.9.2</b>	Container und Netzwerke. . . . .	481
<b>4.10</b>	Bearbeiten von Elementen . . . . .	481
	Instanzen und Vorkommen. . . . .	482
<b>4.10.1</b>	Elementkonfiguration . . . . .	482
<b>4.10.2</b>	Abhängige Elemente . . . . .	489
<b>4.11</b>	Bearbeiten von Daten . . . . .	493
<b>4.11.1</b>	Editoren für skalare Typen. . . . .	495
	Numerischer Editor. . . . .	495
	Logischer Editor . . . . .	495
	Enumeration-Editor . . . . .	496
<b>4.11.2</b>	Der Editor für zusammengesetzte Typen (Tabelleneditor). . . . .	496
	Array-Editor . . . . .	497



	1D-Tabelleneditor . . . . .	498
	2D-Tabelleneditor . . . . .	500
	Feste Tabellen . . . . .	500
	Gruppentabellen und Verteilungen . . . . .	502
<b>4.11.3</b>	Datensätze . . . . .	503
<b>4.12</b>	Bearbeiten von Implementierungen . . . . .	513
<b>4.12.1</b>	Implementierung von Komponenten/Projekten . . . . .	514
<b>4.12.2</b>	Implementierung von skalaren nichtlogischen Elementen . . . . .	518
	Individuelle Implementierung spezifizieren . . . . .	521
	Implementierungstypen verwenden . . . . .	531
<b>4.12.3</b>	Implementierung von methoden-/prozesslokalen Variablen. . . . .	533
<b>4.12.4</b>	Implementierungen für temporäre Variablen. . . . .	535
<b>4.12.5</b>	Implementierung von Arrays, Matrizen und Tabellen. . . . .	535
<b>4.12.6</b>	Implementierung von logischen Elementen. . . . .	537
<b>4.12.7</b>	Implementierung von Enumerationen . . . . .	539
<b>4.12.8</b>	Methoden- und Prozessimplementierungen . . . . .	540
<b>4.12.9</b>	Implementierung von Argumenten und Rückgabewert einer Methode . . . . .	542
<b>4.12.10</b>	Implementierung von Implementation-Casts. . . . .	542
<b>4.12.11</b>	Operatorimplementierungen . . . . .	543
	Automatische Konvertierung von Operator- implementierungen . . . . .	549
<b>4.13</b>	Bearbeiten des Layouts einer Komponente . . . . .	552
<b>4.13.1</b>	Bearbeiten des Layouts einer Klasse . . . . .	553
<b>4.13.2</b>	Bearbeiten des Layouts anderer Komponenten . . . . .	558
<b>4.14</b>	Arithmetische Dienste . . . . .	559
<b>4.14.1</b>	Funktionsweise eines Arithmetischen Dienstes . . . . .	560
<b>4.14.2</b>	Definition von Arithmetischen Diensten . . . . .	561
	Funktionsschlüssel . . . . .	561
	Zulässige Arithmetische Dienste . . . . .	562
	Zulässige Typen. . . . .	564
	Funktionsdeklaration . . . . .	565
<b>4.14.3</b>	Erstellen und Speichern von Arithmetischen Diensten . . . . .	568
<b>4.14.4</b>	Verwendung von Arithmetischen Diensten in ASCET . . . . .	568
	Auswahl eines Satzes . . . . .	569
	Mögliche Fehlerzustände . . . . .	571
<b>4.14.5</b>	Der Schnittstelleneditor für Arithmetische Dienste . . . . .	572
	Funktionsumfang des AS-Editors . . . . .	573
	Starten des Editors . . . . .	574
	Die Benutzeroberfläche des AS-Editors . . . . .	575

	Die Editoroberfläche . . . . .	579
	Den AS-Editor verwenden. . . . .	582
<b>5</b>	Signale und Symbole . . . . .	597
<b>5.1</b>	Der Signalbetrachter . . . . .	597
<b>5.2</b>	Der Symboleditor . . . . .	600
<b>6</b>	Experimentieren . . . . .	603
<b>6.1</b>	Die Experimentierumgebung . . . . .	603
<b>6.1.1</b>	Beschreibung der Menüfunktionen. . . . .	604
<b>6.1.2</b>	Aufrufen und Einrichten der Experimentierumgebung . . . . .	607
<b>6.1.3</b>	Der Ereignisgenerator . . . . .	609
	Ereignisoptionen. . . . .	612
<b>6.1.4</b>	Der Datengenerator . . . . .	616
<b>6.1.5</b>	Das Mess-System . . . . .	625
<b>6.1.6</b>	Das Verstellsystem . . . . .	627
<b>6.1.7</b>	Durchführen von Offline-Experimenten. . . . .	629
<b>6.1.8</b>	Laden und Speichern von Umgebungen . . . . .	638
<b>6.1.9</b>	Der Data-Logger. . . . .	641
<b>6.1.10</b>	Experimentierumgebungs-Optionen . . . . .	655
	Navigieren zwischen Komponenten . . . . .	655
	Anzeigeoptionen . . . . .	656
<b>6.1.11</b>	Manipulieren von Daten . . . . .	657
<b>6.2</b>	Verstellfenster . . . . .	660
<b>6.2.1</b>	Allgemeine Beschreibung der Menüfunktionen. . . . .	661
<b>6.2.2</b>	Dateneditoren für Verstellgrößen . . . . .	665
<b>6.2.3</b>	Arbeiten mit Verstellfenstern . . . . .	665
	Numerischer Editor. . . . .	668
	Logischer Editor . . . . .	672
	Enumeration-Editor . . . . .	673
	Array-Editor . . . . .	673
	1D-Tabelleneditor . . . . .	677
	2D-Tabelleneditor . . . . .	682
	1D-Grafikeditor . . . . .	683
	2D-Grafikeditor . . . . .	688
	3D-Grafikeditor . . . . .	692
<b>6.2.4</b>	Verstellen von Stützstellen im Tabelleneditor. . . . .	695
<b>6.3</b>	Messfenster . . . . .	697
<b>6.3.1</b>	Messfenster auswählen . . . . .	697
<b>6.3.2</b>	Allgemeine Beschreibung von Menüfunktionen . . . . .	698
<b>6.3.3</b>	Arbeiten mit Messfenstern . . . . .	701

<b>6.3.4</b>	Die verschiedenen Messfenster . . . . .	705
	Numerische Anzeige . . . . .	705
	Oszilloskop . . . . .	707
	Recorder . . . . .	725
	Horizontale und Vertikale Balkenanzeige . . . . .	727
	Bitanzeige . . . . .	729
	Monitor . . . . .	729
<b>7</b>	Automatische Dokumentation . . . . .	731
<b>7.1</b>	Erzeugen von Dokumentation . . . . .	731
<b>7.2</b>	Ausgabeformate für Dokumentationsdateien . . . . .	735
	ASCII-Format . . . . .	735
	RTF-Format . . . . .	735
	HTML-Format . . . . .	735
	Postscript-Format . . . . .	736
<b>7.3</b>	Sichten . . . . .	736
	Allgemeines . . . . .	737
	Optionen für die Dokumentation . . . . .	742
	Optionen für die Editoren . . . . .	743
	Optionen für Elementtypen . . . . .	745
<b>7.4</b>	Notizen . . . . .	747
	Index . . . . .	751



# 1 Einführung

---

Im Band „Benutzerhandbuch“ werden alle Programmteile und Arbeitsfenster, die Menüstruktur und die Bedienung des Programms detailliert beschrieben. Die Funktion aller Menüs und Menüfunktionen wird erklärt. Bevor Sie sich mit diesem Teil beschäftigen, sollten Sie mit ASCET bereits vertraut sein, d.h. im Band „Schnelleinstieg“ das Kapitel „ASCET – Grundlagen“ auf Seite 49 und das Tutorial durchgearbeitet haben. Die Reihenfolge der Unterkapitel entspricht weitestgehend der Abfolge von Arbeitsschritten, die bei der Spezifikation von Komponenten und Projekten notwendig sind.

Alle in diesem Band beschriebenen Inhalte sind nach dem gleichen Prinzip aufgebaut. Vor jedem Hauptkapitel steht eine kurze Einleitung, danach eine Beschreibung der Benutzeroberfläche und Menüpunkte, gefolgt von einer ablaforientierten Beschreibung einzelner Arbeitsschritte.

## 1.1 Typischer Arbeitsablauf

---

Bedingt durch die Programmstruktur von ASCET ist der Anwender angehalten, seinen Arbeitsablauf an gewissen Stellen nach einem vorgegebenen Schema durchzuführen. Es gibt mehrere Dialogabfolgen, die unveränderbar sind. Daraus resultierend ergibt sich ein sinnvoller und typischer Arbeitsablauf, der auch bei der Dokumentation berücksichtigt wurde und als Leitfaden verwendet wird. Einen groben Überblick über die Reihenfolge der einzelnen Arbeitsschritte entnehmen Sie bitte der folgenden Liste:

- ASCET einrichten
- Datenbank und Datenbankeinträge anlegen
- Eigene Funktionen einfügen
- Komponenten spezifizieren und simulieren
- Projekte spezifizieren und simulieren
- Datensätze erstellen und bearbeiten
- Implementierungen erstellen und bearbeiten
- Signale und Symbole verwenden
- Experimentieren
- Dokumentation der Ergebnisse

## 1.2 Programmstart

---

### **ASCET starten:**

---

- Die Installation von ASCET legt standardmäßig eine Symbolgrafik auf Ihrem Desktop ab. Ein Doppelklick auf dieses Symbol startet das Programm.
- Alternativ dazu können Sie ASCET auch über die Programmgruppe im **Start**-Menü starten.

Im Weiteren finden Sie eine detaillierte Beschreibung der Benutzeroberfläche des Komponentenmanagers, mit der ASCET standardmäßig startet.

#### **Hinweis**

*Das Verhalten der Software kann individuell angepasst werden. Sie können in ASCET Standardpfade festlegen, verschiedene Farben und Schriftarten setzen, Vorgaben für Import und Export von Daten machen und anderes mehr. Nähere Informationen hierzu finden Sie im Kapitel 2.2 "Komponentenmanager – Einrichten von ASCET".*

## 2 **Der Komponentenmanager**

---

In diesem Kapitel wird der Komponentenmanager, seine Menüstruktur und Bedienung, detailliert beschrieben. Dieses Fenster ersetzt das ASCET-Startfenster und den Datenbank-Browser aus früheren Versionen.

Die Funktion aller Menüs und Menüpunkte wird erklärt. Bevor Sie sich mit diesem Teil beschäftigen, sollten Sie mit ASCET bereits vertraut sein, d.h. das Kapitel „ASCET – Grundlagen“ und das „Tutorial“ im ASCET-Schnelleinstieg durchgearbeitet haben.

### 2.1 **Komponentenmanager – Benutzeroberfläche**

---

#### 2.1.1 **Allgemeine Beschreibung**

---

Beim Starten des Programms wird der ASCET-Komponentenmanager geöffnet. Von hier aus starten Sie die verschiedenen Editoren (siehe Kapitel 4 „Spezifikation von Komponenten und Projekten“). Des Weiteren können Sie mehrere Benutzerprofile verwalten und über einen Optionendialog rechner-, datenbank- und benutzerspezifische Einstellungen vornehmen; z.B. Definition von Speicherverzeichnissen, Einstellungen wie Bildschirmdarstellung oder das Startverhalten der Software (siehe „Komponentenmanager – Einrichten von ASCET“ auf Seite 37).

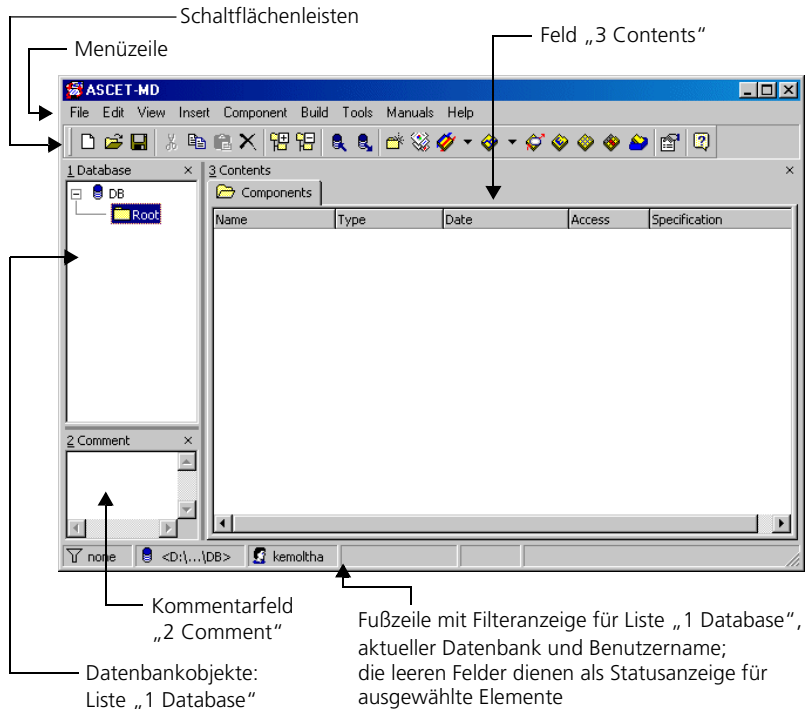
Hauptaufgabe des Komponentenmanagers ist es allerdings, alle Daten, die bei der Arbeit mit ASCET entstehen (Klassen, Module, Projekte, etc.), in einer Datenbank systematisch abzulegen und zu verwalten (siehe „Komponentenmanager – Daten verwalten“ auf Seite 76).

Beschreibungsgegenstand dieses Teilkapitels sind nur die Funktionen, die fester Bestandteil der Oberfläche des Komponentenmanagers sind. Die Editoren, die Experimentierumgebung sowie weitere Zusatzmodule werden in eigenen Kapiteln beschrieben.

#### 2.1.2 **Benutzeroberfläche Komponentenmanager**

---

Alle Datenbankeinträge werden von ASCET in einer Datenbank im Komponentenmanager verwaltet. Hier können Sie Hauptordner und Unterordner anlegen, einzelne Objekte verschieben, kopieren, importieren und exportieren, aber auch komplett neue Datenbanken anlegen. Neben der Organisation ihrer Daten ist in ASCET ein Bearbeiten der Komponenten und Projekte möglich. In früheren Versionen war dieses Bearbeiten nur in den Spezifikationseditoren möglich.



Die Ordner und Einträge der aktuellen Datenbank werden in der Liste „1 Database“ angezeigt. Die Wurzel der Baumstruktur bildet der Name der Datenbank. Jede Datenbank enthält einen oder mehrere Ordner, die ihrerseits weitere Ordner und Datenbankeinträge enthalten. Sie werden in einer Hierarchie in der Liste „1 Database“ organisiert und angezeigt.

Das Feld „2 Comment“ ist ein Textfeld, das Anmerkungen oder interne Kommentare zum aktuellen Ordner oder Eintrag enthält. Typischerweise wird dieses Feld für Entwicklernotizen und interne Informationen zu Versionen verwendet. Diese Informationen werden in der automatisch erzeugten Dokumentation nicht mit aufgeführt. Sie sind nicht zu verwechseln mit den im Notizeneditor eingegebenen Informationen (siehe Kapitel 7.4 „Notizen“ auf Seite 747).

### Hinweis

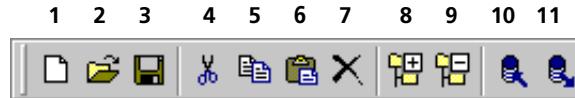
Beim Export (s. Kapitel 2.3.3 auf Seite 94) wird der Inhalt des Feldes „2 Comment“ **nur** für Hauptordner und Datenbankeinträge mit exportiert. Anmerkungen zu Unterordnern werden **nicht** exportiert.



Das Feld „3 Contents“ zeigt in verschiedenen Registern Informationen zum ausgewählten Ordner oder Datenbankeintrag an. Abhängig von der Auswahl, Ordner oder Datenbankeintrag, sowie der Art des Datenbankeintrags ändert sich im Feld „3 Contents“ der angezeigte Inhalt. Details finden Sie in Kapitel 2.1.3 „Ansichtskonzepte im Komponentenmanager“.

### *Bedienelemente in den Schaltflächenleisten*

#### **Standard-Schaltflächenleiste:**



1. New (erzeugt eine neue Datenbank)
2. Open (öffnet eine Datenbank)
3. Save (sichert die Datenbank)
4. Cut (schneidet das ausgewählte Element aus)
5. Copy (kopiert das ausgewählte Element)
6. Paste (fügt das Element aus der Zwischenablage ein)
7. Delete (löscht das ausgewählte Element)
8. Expand all (klappt die Baumstruktur in der Liste „1 Database“ weitestmöglich auf)
9. Collapse all (klappt die Baumstruktur in der Liste „1 Database“ vollständig zu)
10. Import
11. Export

#### **Schaltflächenleiste „Insert“:**



1. Insert Folder (fügt einen Ordner ein)
  2. Insert Project (fügt ein Projekt ein)
  3. Insert Module - <Type> (fügt ein Modul ein)
  4. Insert Class - <Type> (fügt eine Klasse ein)
- Mit den Pfeilen (3a und 4a) kann der Typ des Objekts - Blockdiagramm, C-Code, ESDL - gewählt werden.

5. Insert State Machine (fügt einen Zustandsautomaten ein)
6. Insert Container (fügt einen Container ein)
7. Insert Boolean Table (fügt eine Boolesche Tabelle ein)
8. Insert Conditional Table (fügt eine Bedingungstabelle ein)
9. Insert Container (fügt einen Container ein)

### Schaltflächenleiste „Tools“:

1 2



1. Options (öffnet das Optionsfenster)
2. ? (öffnet das Fenster „About ASCET“ mit Informationen zu den installierten Produkten der ASCET-Produktfamilie)

### Symbolgrafiken für Datenbankobjekte

---



1. Datenbank
2. Projekt
3. Modul
4. Klasse
5. CT-Block
6. Zustandsautomat
7. Enumeration
8. Boolesche Tabelle
9. Bedingungstabelle
10. Symbol
11. Signal
12. Container
13. ASAM-MCD-2MC-Projekt

- **File**

- *New Database* (<CTRL> + <N>)  
Datenbank neu anlegen.
- *Open Database* (<CTRL> + <O>)  
Datenbank öffnen.
- *Close Database*  
Datenbank schließen.
- *Save Database* (<CTRL> + <S>)  
Speichern der Änderungen in der Datenbank.
- *Save Database as...*  
Erzeugt eine Sicherungskopie von der gesamten Datenbank unter einem von Ihnen angegebenen Pfad.
- *Import* (<CTRL> + <M>)  
Importiert Datenbankeinträge aus einer Exportdatei und zeigt sie in der Liste „1 Database“.
- *Import directory*  
Importiert den kompletten Inhalt eines Verzeichnisses.
- *Export* (<CTRL> + <E>)

### **Hinweis**

*Diese Funktion ist nur verfügbar, wenn ASCET-MD installiert ist. Ist dies der Fall, ist sie in der Liste „1 Database“ auch als Kontextmenü verfügbar.*

Exportieren ausgewählter Datenbankeinträge oder kompletter Ordner aus der Liste „1 Database“ in eine oder mehrere Exportdateien (\* .exp).

- *1 D:\..\DB*  
(und weitere ähnliche Einträge)  
Liste der zuletzt geöffneten Datenbanken.
- *Exit* (<ALT> + <F4>)  
ASCET beenden.

- **Edit**

### Hinweis

Die Funktionen **Cut**, **Copy**, **Paste**, **Delete** und **Rename** sind in der Liste „1 Database“ auch als Kontextmenü verfügbar. Bis auf **Find** und **Query** stehen die Funktionen nur zur Verfügung, wenn ASCET-MD installiert ist.

- *Cut* (<CTRL> + <X>)  
Schneidet den markierten Datenbankeintrag aus der Datenbank aus.
- *Copy* (<CTRL> + <C>)  
Kopiert den markierten Datenbankeintrag in die Zwischenablage der Datenbank.
- *Paste* (<CTRL> + <V>)  
Fügt einen ausgeschnittenen oder kopierten Datenbankeintrag aus der Zwischenablage in die Datenbank ein.
- *Delete* (<DEL>)  
Endgültiges Löschen eines markierten Datenbankeintrags. Wird ein Ordner gelöscht, werden auch alle darin enthaltenen Elemente gelöscht.

### Hinweis

Verwenden Sie die Funktion **Edit** → **Delete** vorsichtig, sie kann **nicht** rückgängig gemacht werden.

- *Rename* (<F2>)  
Umbenennen eines markierten Datenbankeintrags.
- *Find* (<CTRL> + <F>)  
Suchen einer Zeichenkette in C-Code- oder ESDL-Komponenten.
- *Replace* (<CTRL> + <H>)  
Ersetzen einer Zeichenkette in C-Code- oder ESDL-Komponenten.
- *Query*. (<CTRL> + <Q>)  
Durchsucht die Datenbank unter verschiedenen Gesichtspunkten (siehe Abschnitt „Durchsuchen einer Datenbank“ auf Seite 139).

- **View**

- *Expand all*  
Der gesamte Inhalt (Ordner und Datenbankeinträge) der Datenbank wird angezeigt.
- *Collapse all*  
Nur der Name der Datenbank wird angezeigt.
- *Filter* → <component type>  
Ermöglicht die Filterung der Datenbankeinträge im Feld „1 Database“.
- *Show /Hide*  
Blendet verschiedene Fensterelemente ein oder aus;  
*Database List* → das Feld „1 Database“,  
*Comment* → das Feld „2 Comment“,  
*Contents* → das Feld „3 Contents“,  
*Monitor* → das Monitorfenster
- *Update* (<F5>)  
Aktualisieren des Komponentenmanagers.

- **Insert**

### Hinweis

Das Menü **Insert** ist in der Liste „1 Database“ auch als Kontextmenü verfügbar. Es ist nur verfügbar, wenn ASCET-MD installiert ist.

- *Folder* (<INSERT>)  
Einfügen eines neuen Ordners.
- *Project*  
Einfügen eines neuen Projekts.
- *Module*  
Einfügen eines neuen Moduls;  
*Block Diagram* → als Blockdiagramm,  
*ESDL* → in ESDL,  
*C Code* → in C-Code.
- *Class*  
Einfügen einer neuen Klasse (Untermenüs wie bei Modulen).
- *Continuous Time Block*  
Einfügen eines neuen CT-Blocks (Untermenüs wie bei Modulen).

- *State Machine*  
Einfügen eines neuen Zustandsautomaten.
- *Enumeration*  
Einfügen einer neuen Enumeration.
- *Boolean Table*  
Einfügen einer neuen Booleschen Tabelle.
- *Conditional Table*  
Einfügen einer neuen Bedingungstabelle (s. Kapitel 4.7).
- *Icon*  
Einfügen eines neuen Symbols.
- *Signal*  
Einfügen eines neuen Signals.
- *Container*  
Einfügen eines neuen Containers.

- **Component**

#### Hinweis

Die Funktion **Edit Item** ist in der Liste „1 Database“ auch als Kontextmenü verfügbar.

**Edit Layout, Replace References, Become another Item** und **Reproduce as** sind nur verfügbar, wenn ASCET-MD installiert ist.

- *Edit Item* (<RETURN>)  
Startet den Editor für den markierten Datenbankeintrag.
- *Edit Layout*  
Bearbeiten eines Komponentenlayouts.
- *Enumeration*  
nur aktiv, wenn im Feld „1 Database“ eine Enumeration ausgewählt wurde und der Fokus im Feld „3 Contents“ liegt. Siehe Seite 28.
- *Disallow Import*  
Verbietet das Überschreiben von Datenbankeinträgen beim Import.
- *Access Rights*  
Änderungen von Zugriffsberechtigungen für einen Ordner oder einen Datenbankeintrag.

- *Password*  
Aktivieren und Deaktivieren des Passwortschutzes (nur für Ordner).
- *Show References*  
Anzeigen der Referenzen auf eine Komponente.
- *Replace References*  
Ersetzen der Referenzen auf einen Datenbankeintrag.
- *Become another Item*  
Ersetzen eines Datenbankeintrags.
- *Reproduce As*  
Kopieren der Struktur einer Komponente  
*Block Diagram* → in ein Blockdiagramm,  
*ESDL* → in ESDL,  
*C Code* → in C-Code.
- *Notes*  
Bearbeiten der Notizen einer Komponente.
- *Reset Operator Implementation*  
Entfernt die Operatorimplementierungen  
*Flat* → der ausgewählten Komponenten,  
*Recursive* → der ausgewählten Komponenten und der darin referenzierten Komponenten.
- **Build**
  - *Touch All*  
Alle Komponenten der Datenbank werden als *bearbeitet* markiert. Damit wird eine Gesamtcompilierung des Projekts erzwungen.
  - *Clean All*  
Der in der Datenbank gespeicherte Code des Projekts wird verworfen.
- **Tools**
  - *Documentation* → *Contents*  
Öffnet das Fenster „Documentation Contents“. Von hier aus wird die Auswahl der zu dokumentierenden Einträge vorgenommen.
  - *Documentation* → *Options*  
Einstellung der Optionen für die Dokumentationserzeugung.

– *Database* → *Performance Utilities*

Bei Anwahl öffnet sich das Dialogfenster „Database Info for: <Database>“. Es enthält vier, teilweise zu kombinierende Datenbankverwaltungsfunktionen:

**Optimize** fasst die durch vielfaches Editieren entstandenen Datenbank-Bruchstücke wieder zusammen (Defragmentierung).

**Convert** bereinigt die Referenzierungen der Datensätze untereinander („X benutzt Y“) nach Änderungen der Datenbankstruktur und beschleunigt so den Zugriff.

**Repair** baut die Datenbank komplett neu auf; dabei werden auch die Funktionen „Optimize“ und „Convert“ durchgeführt.

**Check** kontrolliert die Strukturen und Referenzierungen in der Datenbank und protokolliert die Ergebnisse im Monitorfenster.

– *Database* → *Convert*

Konvertieren der Datenbank;

*Modify components to force impl/data update* → alle Komponenten erhalten den Status *modified*, sodass beim nächsten Zugriff auf eine Komponente deren Implementierung neu überprüft und – wenn nötig – angepasst wird;

*All Names to ANSI-C* → sämtliche Namen in der Datenbank werden nach ANSI-C konvertiert;

*System Constants to Constants* → Umwandeln von Systemkonstanten in Konstanten (s. Kapitel „Die Art von Elementen“ im ASCET Referenzhandbuch);

*Variables to Volatile* → weist allen Variablen der Datenbank das Attribut *volatile* zu (s. Seite 117 und Seite 486);

*Parameters to Nonvolatile* → weist allen Parametern der Datenbank das Attribut *non-volatile* zu;

*Components/Elements to default memory location* → trägt in den Implementierungen aller Elemente der Datenbank den Speicherbereich `Default` ein;

*Operator Implementations to Impl. Casts* → wandelt vorhandene Operatorimplementierungen in Implementation-Casts um (s. Seite 549);

*Reset Operator Implementations* → entfernt alle Operatorimplementierungen aus der Datenbank.



- *Database → List Operator Implementations*  
Zeigt im ASCET-Monitorfenster eine Liste der Operatorimplementierungen an.
- *Database → Compare Database*  
Vergleicht die aktuelle Datenbank mit einer ausgewählten zweiten Datenbank.
- *Arithmetic Service Editor*  
Öffnet den Editor für Arithmetische Dienste für ein Target (siehe Kapitel 4.14);  
*PC → Target PC*;  
weitere Untermenüs kommen hinzu, wenn ASCET-RP oder ASCET-SE installiert sind.
- *Network Settings*  
Öffnet das Fenster „Network settings for ETAS hardware (Page 1)“; nur für ASCET-RP von Bedeutung.
- *Views*  
Öffnet das Fenster „Views“, in dem Sichten verwaltet werden (siehe Kapitel 7.3 auf Seite 736).
- *Options*  
Öffnet das Fenster „Options“, in dem verschiedene Optionen eingestellt werden (s. Kapitel 2.2 auf Seite 37).

- **Manuals**

### Hinweis

Wenn Sie keine Handbücher installiert haben, enthält das Menü keine Funktionen.

- *English user's guide*  
Öffnet das englische Benutzerhandbuch im Acrobat-Reader.
- *English reference*  
Öffnet das englische Referenzhandbuch im Acrobat-Reader.
- *German user's guide*  
Öffnet das deutsche Benutzerhandbuch im Acrobat-Reader.
- *German reference*  
Öffnet das deutsche Referenzhandbuch im Acrobat-Reader.

- *Open manuals folder*

Öffnet das Verzeichnis `ETAS\ETASManuals` im Windows-Explorer.

Wenn Sie ASCET-RP oder ASCET-SE installieren, werden weitere Menüfunktionen für die entsprechenden Handbücher eingefügt.

- **Help**

- *Contents*

Öffnet den Inhalt der Online-Hilfe zu ASCET.

- *Index*

Öffnet den Index der Online-Hilfe zu ASCET.

- *Loaded Packages*

Zeigt im Monitorfenster eine Liste aller installierten Integrationsmodule (siehe Kapitel „Systeminformation“ im ASCET-Schnelleinstieg).

- *Loaded Targets*

Zeigt im Monitorfenster eine Liste der installierten ASCET-SE an.

- *Product Disclaimer*

Öffnet das Disclaimer-Fenster mit einer Gefahrenmeldung.

- *Problem Report*

Startet das Tool „Problem Report“ (siehe Kapitel 2.1.4).

- *Hotkey Assignment*

Öffnet ein Fenster mit einer Liste der Tastaturkürzel.

- *About*

Öffnet ein Fenster mit Informationen zu den installierten Produkten der ASCET-Produktfamilie.

- *Support*

Öffnet ein Fenster mit den Adressen der ASCET-Hotlines.

- *License Info*

Öffnet ein Fenster mit Informationen zur Lizenzierung (siehe Kapitel „Lizensierung“ im ASCET-Schnelleinstieg).

## Kontextsensitive Menüfunktionen

Die nachfolgend beschriebenen Funktionen der Menüs **Edit** und **Component** haben je nach Fokus verschiedene Bedeutungen. In diesem Abschnitt ist für jedes Register im Feld „3 Contents“ aufgelistet, welche Bedeutungen die Menüfunktionen haben, wenn der Fokus in diesem Feld liegt.

### Hinweis

*Die in diesem Abschnitt **nicht** beschriebenen Funktionen der Menüs haben entweder unabhängig vom Fokus immer dieselbe Bedeutung, oder sie sind deaktiviert.*

#### Register „Element“:

- **Edit**
  - *Copy* (<CTRL> + <C>)  
Kopiert die ausgewählten Elemente in die Zwischenablage der Datenbank.
  - *Paste* (<CTRL> + <V>)  
Fügt die ausgewählten Elemente aus der Zwischenablage der Datenbank in die Komponente ein.
  - *Delete* (<Del>)  
Löscht die gewählten Elemente aus der Komponente.
  - *Rename* (<F2>)  
Umbenennen des Elements.
- **Component**
  - *Edit Item* (<RETURN>)  
Öffnet den Elementeditor für das gewählte Element.

#### Register „Data“:

- **Edit**
  - *Copy* (<CTRL> + <C>)  
Kopiert die aktuellen Daten des ausgewählten Elements in die Zwischenablage der Datenbank.
  - *Paste* (<CTRL> + <V>)  
Fügt die Daten aus Zwischenablage der Datenbank in das ausgewählte Element ein.

- *Delete* (<Del>)
 

Löscht die gewählten Elemente aus der Komponente.
- *Rename* (<F2>)
 

Umbenennen des Elements.

- **Component**

- *Edit Item* (<RETURN>)
 

Öffnet den Dateneditor für das gewählte Element.

**Register „Implementation“:**

- **Edit**

- *Copy* (<CTRL> + <C>)
 

Kopiert die aktuelle Implementierung des ausgewählten Elements in die Zwischenablage der Datenbank.
- *Paste* (<CTRL> + <V>)
 

Fügt die Implementierung aus Zwischenablage der Datenbank in das ausgewählte Element ein.

- *Delete* (<Del>)
 

Löscht die gewählten Elemente aus der Komponente.
- *Rename* (<F2>)
 

Umbenennen des Elements.

- **Component**

- *Edit Item* (<RETURN>)
 

Öffnet den Implementierungseditor für das gewählte Element.

**Im Feld „1 Database“ ist eine Enumeration gewählt:**

- **Edit**

- *Delete* (<Del>)
 

Löscht den gewählten Enumerator aus der Komponente.
- *Rename* (<F2>)
 

Umbenennen des Enumerators.

- **Component**

- *Enumeration* → *Add* (<INSERT>)
 

Enumerator hinzufügen.
- *Enumeration* → *Shift Up* (<CTRL> + <U>)
 

Enumerator in der Liste nach oben verschieben.

- Enumeration → Shift Down (<CTRL> + <D>)  
Enumerator in der Liste nach unten verschieben.

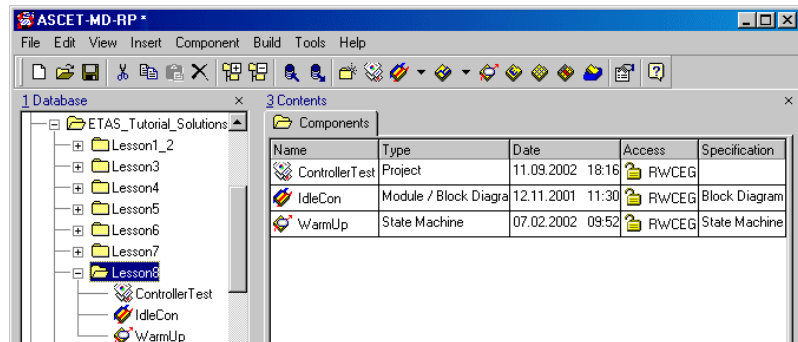
### 2.1.3 Ansichtskonzepte im Komponentenmanager

ASCET bietet ein erweitertes Ansichtskonzept für die Datenbankeinträge. Ordner, Komponenten, Projekte, Container und Enumerationen können im Feld „3 Contents“ unter verschiedenen Gesichtspunkten angezeigt werden. Für Komponenten und Projekte stehen in den Registern im Feld „3 Contents“ eine Element-, eine Daten- und eine Implementierungsansicht zur Verfügung, für Komponenten außerdem noch eine Layoutansicht. Für Ordner gibt es die Ordneransicht, die den Inhalt sowie Informationen zu den enthaltenen Objekten darstellt. Die Containeransicht entspricht weitgehend der Ordneransicht.

Jedes Register bietet bestimmte Bearbeitungsmöglichkeiten, die sowohl über die Menüs **Edit** und **Component** als auch über das Kontextmenü im Feld „3 Contents“ zugänglich sind.

#### Auswählen der Ordneransicht:

- Markieren Sie einen Ordner im Feld „1 Database“.

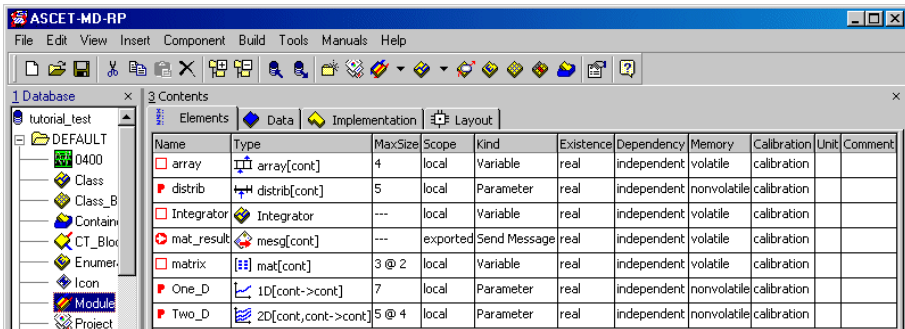


Der Inhalt des Ordners sowie Name, Typ, Erstellungszeitpunkt, Zugriffsrechte und Art der Erstellung der enthaltenen Objekte werden angezeigt.

Wie Sie in diesem Register arbeiten, ist im Abschnitt „Bearbeiten von Komponenten und Projekten im Komponentenmanager“ auf Seite 86 erklärt.

## Auswählen der Elementansicht einer Komponente/eines Projekts:

- Markieren Sie im Feld „1 Database“ eine Komponente oder ein Projekt.
- Klicken Sie im Feld „3 Contents“ auf das Register „Elements“, um die Elementansicht zu erhalten.

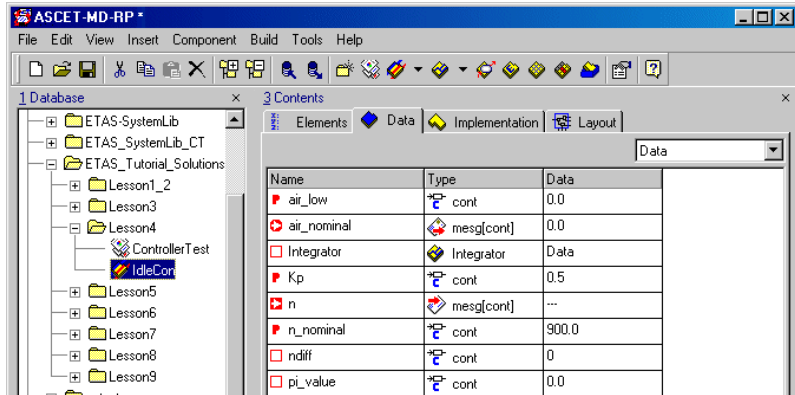


Alle Elemente der Komponente/des Projekts werden mit Name, Typ, Geltungsbereich, Art und Einheit angezeigt. Wurde zu einem Element ein Kommentar eingegeben (s. Seite 486), wird dieser ebenfalls angezeigt. Für Arrays, Matrizen und Kennlinien/-felder wird auch die maximale Größe angezeigt.

Wie Sie die Elemente bearbeiten, ist im Abschnitt „Bearbeiten von Komponenten und Projekten im Komponentenmanager“ auf Seite 86 beschrieben.

## Auswählen der Datenansicht:

- Markieren Sie im Feld „1 Database“ eine Komponente/ein Projekt.
- Klicken Sie im Feld „3 Contents“ auf das Register „Data“, um die Datenansicht zu erhalten.

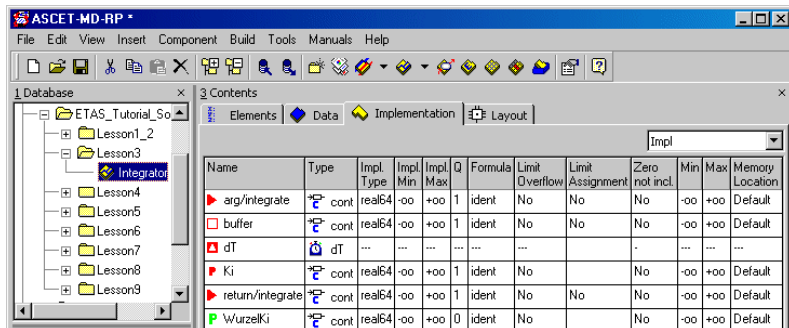


Namen, Typ und Daten der Elemente werden angezeigt. Im Kombikästchen kann ein anderer Datensatz gewählt werden.

Wie Sie hier die Daten bearbeiten, ist im Abschnitt „Bearbeiten von Komponenten und Projekten im Komponentenmanager“ auf Seite 86 beschrieben.

### Auswählen der Implementierungsansicht:

- Markieren Sie im Feld „1 Database“ eine Komponente/ein Projekt.
- Klicken Sie im Feld „3 Contents“ auf das Register „Implementation“, um die Implementierungsansicht zu erhalten.



Alle Informationen der derzeitigen Implementierung werden angezeigt. Im Kombikästchen kann eine andere Implementierung gewählt werden.

Wie Sie hier die Implementierungen bearbeiten, ist im Abschnitt „Bearbeiten von Komponenten und Projekten im Komponentenmanager“ auf Seite 86 beschrieben.

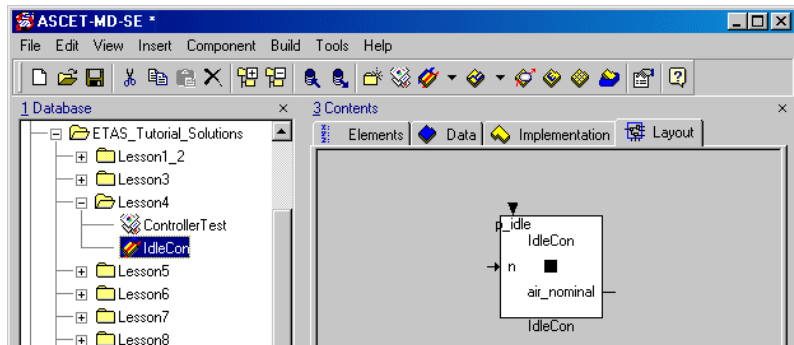
### Auswählen der Layoutansicht einer Komponente:

- Markieren Sie im Feld „1 Database“ eine Komponente.

#### Hinweis

*Für Projekte gibt es kein Layout.*

- Klicken Sie im Feld „3 Contents“ auf das Register „Layout“, um die Layoutansicht zu erhalten.

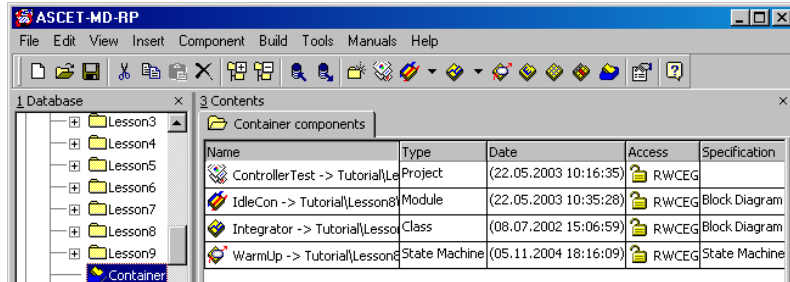


Das Layout der Komponente wird angezeigt. Wie Sie hier das Layout bearbeiten, ist im Abschnitt „Bearbeiten von Komponenten und Projekten im Komponentenmanager“ auf Seite 86 beschrieben.



## Auswählen der Containeransicht:

- Markieren Sie im Feld „1 Database“ einen Container.

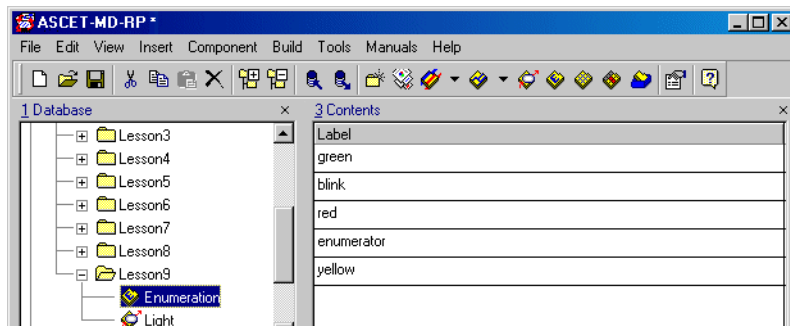


Der Inhalt des Containers sowie Name, Typ, Erstellungszeitpunkt, Zugriffsrechte und Art der Erstellung der enthaltenen Objekte werden angezeigt.

Wie Sie in diesem Register arbeiten, ist in Kapitel 4.9.1 „Arbeiten mit Containern“ erklärt.

## Auswählen der Enumerationsansicht:

- Markieren Sie eine Enumeration im Feld „1 Database“.
- Im Feld „3 Contents“ werden die Enumeratoren aufgelistet.



Wie Sie die Enumeration bearbeiten, ist im Abschnitt „Erstellen einer Enumeration:“ auf Seite 80 beschrieben.

## 2.1.4 ETAS Support-Funktion „Problem Report“

---

ASCET stellt Ihnen eine Support-Funktion zur Verfügung, um Probleme, die während der Arbeit mit dem Programm aufgetreten sind, an ETAS weiterzuleiten. Wenn Sie die Supportfunktion nutzen, packt ASCET den gesamten Inhalt des „log“-Verzeichnisses (alle \*.log-Dateien) zusammen mit einer textlichen Beschreibung in ein Archiv und speichert es im Verzeichnis `ETAS\LogFiles\` als Datei `EtasLogFiles01.zip`. Bei weiteren Archivdateien wird der Dateiname automatisch hochgezählt (bis max. 19), so dass alte Archivdateien nicht sofort überschrieben werden.

### **Hinweis**

*Es können nur max. 20 Archivdateien angelegt werden; danach werden die alten Dateien bei 00 beginnend wieder überschrieben.*

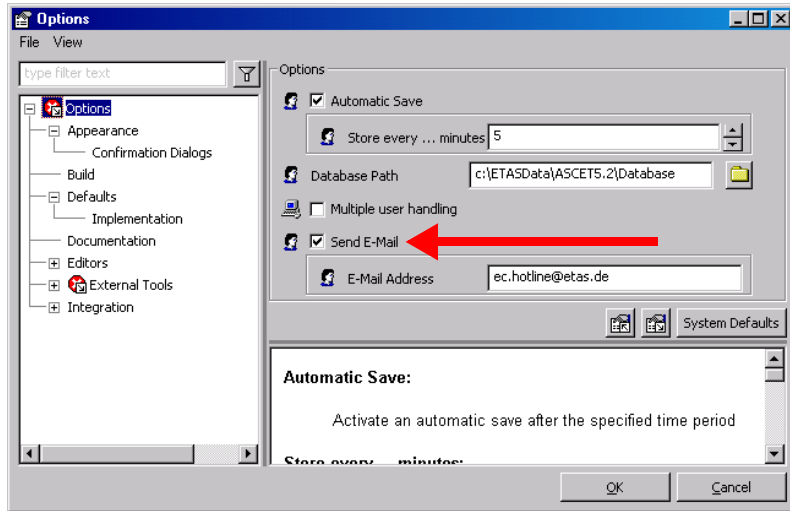
Sie können ASCET so einrichten, dass dieses Archiv automatisch an den ETAS Hotline-Service geschickt wird. Dazu benötigen Sie ein E-Mail-Programm, das MAPI-kompatibel ist (z.B. MS Exchange oder Eudora). Bei anderen E-Mail-Programmen müssen Sie das Archiv selbst als Dateianhang verschicken.

### **„Problem Report“ Support-Funktion einrichten:**

---

- Wählen Sie im Komponentenmanager die Menüfunktion **Tools** → **Options**.  
Das Fenster „Options“ öffnet sich.

- Aktivieren Sie im Knoten „Options“ die Option **Send E-Mail**.



Damit ist das automatische Verschicken der E-Mail an die Hotline eingeschaltet. Wenn Ihr E-Mail-Programm nicht MAPI-kompatibel ist, ist diese Funktion ohne Bedeutung.

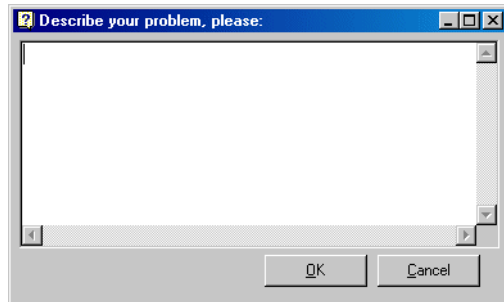
- Passen Sie gegebenenfalls im Feld „E-Mail Address“ die E-Mail-Adresse des Hotline-Service an. Die derzeitige Adresse lautet: `ec.hotline@etas.de`

## Problembeschreibung verschicken:

---

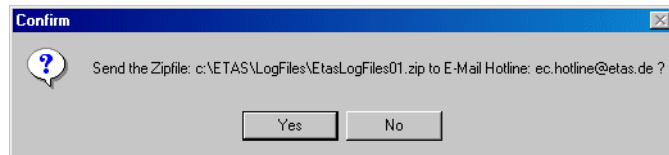
- Wählen Sie im Komponentenmanager **Help** → **Problem Report**, um Ihr Problem an ETAS zu schicken.

Ein Dialogfenster öffnet sich und fragt nach einer Beschreibung des Problems.



- Geben Sie den Text ein und klicken Sie auf **OK**.

Ihre Lizenz wird überprüft. Wenn Sie eine lizenzierte ASCET-Version verwenden, wird der Bericht in ein Dateiarchiv gepackt. Wenn Sie den automatischen Versand aktiviert haben, wird das Archiv sofort verschickt. Andernfalls werden Sie gefragt, ob das Archiv verschickt werden soll.



- Klicken Sie **Yes**, um die E-Mail zu verschicken; andernfalls klicken Sie **No**.

Wenn Ihr E-Mail-Programm nicht MAPI-kompatibel ist, müssen Sie die Archivdatei selbst als Dateianhang verschicken.


## 2.2 Komponentenmanager – Einrichten von ASCET

Dieses Kapitel beschreibt die Konfigurationsoptionen von ASCET und die Arbeit mit Benutzerprofilen.


### Hinweis

*In diesem Handbuch sind die Optionen von ASCET-MD beschrieben. Zusatzprogramme wie ASCET-RP können eigene Optionen hinzufügen; diese sind im Handbuch des betreffenden Zusatzprogramms beschrieben.*

In ASCET können Sie verschiedene Optionen setzen. Es gibt zwei Arten von Optionen: allgemeine und benutzerspezifische Optionen. Beide Arten werden im Fenster „Options“ verwaltet und in XML-Dateien abgelegt. Wenn ASCET nach der Installation das erste Mal gestartet wird, sind diese XML-Dateien noch nicht vorhanden. Sie werden automatisch angelegt und mit den Standardeinstellungen des Systems oder – wenn eine ältere ASCET-Version installiert war – mit den Einstellungen der älteren Version belegt.

*Allgemeine Optionen* (Symbol ) sind für die ASCET-Installation auf Ihrer Workstation spezifisch. Sie umfassen das Zielverzeichnis für die Codegenerierung, verschiedene Pfadangaben, Einstellungen für die Code-Vorschau sowie die Auswahl von Einzel- oder Mehrbenutzermodus für den Programmablauf.

Allgemeine Optionen werden im Datenverzeichnis in der Datei `stationSettings.xml` abgelegt.

*Benutzerspezifische Optionen* (Symbol ) umfassen verschiedene Dinge, von Einstellungen für Diagramme, Export-/Import-Dateipfaden bis zu diversen allgemeinen Optionen, die später in diesem Kapitel genauer beschrieben werden. Dadurch besteht die Möglichkeit, eine an Ihre Bedürfnisse angepasste Benutzerumgebung zu erzeugen. Änderungen der Benutzeroptionen werden für jeden Benutzer getrennt abgelegt, wenn die Benutzerauswahl beim Start aktiv ist. Neue Benutzerprofile starten mit den Standardeinstellungen.

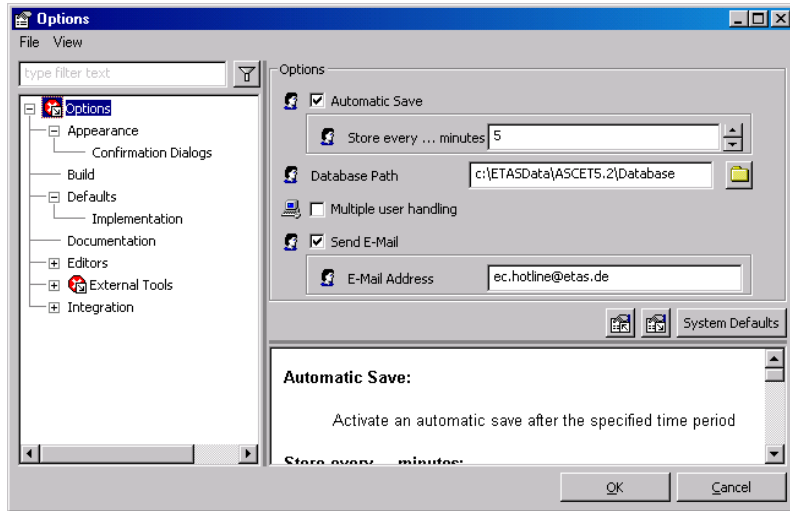
Benutzerspezifische Optionen werden im Verzeichnis des jeweiligen Benutzers (`ETASData\ASCET5.2\User\) in der Datei userSettings.xml abgelegt.`



### Hinweis

*Bevor Sie benutzerspezifische Optionen setzen, sollten Sie entscheiden, ob Ihre Installation von ASCET im Einzel- oder Mehrbenutzerbetrieb läuft, um sicherzugehen, dass Ihre Einstellungen im richtigen Benutzerprofil abgelegt werden.*

Optionen mit ungültigen Werten werden durch einen roten Kreis mit weißem X markiert:  .

**Benutzeroberfläche des ASCET-Optionsfensters:** Das ASCET-Optionsfenster enthält folgende Elemente:



- links oben ein Filter für die Optionen
- links eine Baumansicht mit den Optionsgruppen als Knoten  
Das Symbol  bzw.  an einem Knoten bedeutet, dass der Knoten bzw. einer seiner Unterknoten eine ungültige Option enthält.
- rechts oben das Anzeigefeld für die Optionen des gewählten Knotens
- rechts unten eine Kurzbeschreibung der angezeigten Optionen
- die Schaltflächen **Import Options for selected Node from XML File**, **Export Options of selected Node into XML File** und **System Defaults**
- das Menü **File** mit folgenden Menüfunktionen
  - **Import**  
Importiert die Optionen für den gewählten Knoten aus einer XML-Datei.
  - **Export**  
Exportiert die Optionen des gewählten Knoten in eine XML-Datei.
- das Menü **View** mit folgenden Menüfunktionen
  - **Expand All**  
Klappt die Baumansicht so weit wie möglich auf.

- **Collapse All**  
Klappt die Baumansicht so weit wie möglich zu.
- **Show/Hide Description**  
Zeigt/versteckt die Kurzbeschreibung der angezeigten Optionen.

### Setzen von Optionen für ASCET:

---

- Wählen Sie im Komponentenmanager **Tools → Options**

oder



- klicken Sie auf die Schaltfläche **Options**.

Das ASCET-Optionsfenster (Dialogfenster „Options“) öffnet sich.

Die Optionen sind nach Gebieten geordnet; jedes Gebiet ist links im Fenster „Options“ als Knoten in der Baumansicht dargestellt.

- Wählen Sie den gewünschten Knoten, um die Einstellungen für ASCET zu setzen.

Einige Optionen erscheinen als Optionsfeld, Eingabefeld oder Kombikästchen. Andere Optionen haben neben dem Namen eine Schaltfläche, mit der ein weiteres Dialogfenster geöffnet wird.



- Nehmen Sie die neue Einstellung entsprechend den jeweiligen Steuerelementen vor.

Geänderte Optionen sind mit einem \* markiert, Optionen mit einer anderen als der Standardeinstellung erscheinen in Fettschrift.

Sie können nacheinander in verschiedenen Knoten des Fensters „Options“ Optionen setzen.

- Klicken Sie auf die Schaltfläche **System Defaults**, um die Standardeinstellungen für den aktuellen Knoten wiederherzustellen.

- Wenn Sie alle Einstellungen vorgenommen haben, klicken Sie auf **OK**.

Die Einstellungen werden übernommen, das Fenster „Options“ wird geschlossen.

### Hinweis

*Einige neue Einstellungen wirken nicht sofort. Wenn Sie beispielsweise die Listenschriftart der ASCET-Bedienoberfläche ändern, werden dadurch bereits geöffnete Fenster nicht beeinträchtigt. Sie müssen das betreffende Fenster schließen und neu öffnen, damit die Einstellungen in diesem Fenster wirken.*

### **Exportieren von Optionen:**

Wenn Sie die Optionen eines Knotens eingerichtet haben, können Sie sie in eine XML-Datei exportieren. Öffnen Sie das ASCET-Optionsfenster.



- Wählen Sie den Knoten, dessen Optionen Sie exportieren wollen.
- Setzen Sie ggf. die Optionen dieses Knotens.
- Klicken Sie auf die Schaltfläche **Export Options of selected Node into XML File**

oder

- wählen Sie **File** → **Export**.

Das Windows-Dateiauswahlfenster öffnet sich. Das Format \* .xml ist fest eingestellt.

- Setzen Sie Pfad und Namen der Exportdatei.
- Klicken Sie auf **Speichern**.

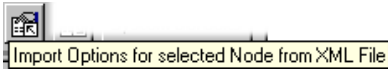
Die Optionen des gewählten Knotens werden in die angegebene XML-Datei geschrieben.

### **Importieren von Optionen:**

Exportierte Optionen können wieder importiert werden. Auf diese Weise können z.B. einheitliche Optionen für mehrere Anwender definiert werden.

- Öffnen Sie das ASCET-Optionsfenster.
- Wählen Sie den Knoten, dessen Optionen Sie importieren wollen.





- Klicken Sie auf die Schaltfläche **Import Options for selected Node from XML File**

oder

- wählen Sie **File** → **Import**.

Ein Bestätigungsfenster öffnet sich.

- Aktivieren Sie die Option **Remember my Decision**, wenn Sie *jede* Abfrage dieser Art auf dieselbe Weise beantworten wollen (siehe auch Seite 47).

Das Fenster „Import Options“ wird dann nicht mehr angezeigt. Sie können die Einstellung im ASCET-Optionsfenster, Knoten „Confirmation Dialogs“, wieder rückgängig machen (s. „Optionen für Bestätigungsfenster“ auf Seite 47.)

- Bestätigen Sie das Überschreiben der vorhandenen Optionen mit **OK**.

Der Windows-Dateiauswahldialog öffnet sich. Das Format \*.xml ist fest eingestellt.

- Wählen Sie die XML-Datei, die die gewünschten Optionen enthält.
- Klicken Sie auf **Öffnen**.

Die in der angegebenen XML-Datei enthaltenen Optionen werden in das Optionsfenster übernommen.

### Filtern der Optionen:

---

Sie können die Optionen filtern und so die Anzeige übersichtlicher gestalten.

- Geben sie in das Textfeld des Filters einen Text ein.

Das kann ein oder mehrere Wörter, aber auch ein teil eines Wortes sein.

- Klicken Sie auf die Schaltfläche **Apply Filter String**.

Im Anzeigefeld werden nur Optionen angezeigt, deren Name, Wert oder Kurzbeschreibung den eingegebenen Text enthält. In der Baumansicht werden nur Knoten ange-



zeigt, die zum Filter passende Optionen enthalten. Zusätzlich werden ggf. übergeordnete Knoten angezeigt, damit die Hierarchie intakt bleibt.

### 2.2.1 Grundlegende Optionen

---

Im obersten Knoten des Optionsfensters, dem Knoten „Options“ (siehe Abbildung auf Seite 39), setzen Sie einige grundlegende Optionen.

Option	Wert	Beschreibung
Automatic Save	aktiviert / deaktiviert	Schaltet die automatische Speicherfunktion ein/aus.
Store every ... minutes	Zahl	Zeitintervall für automatische Speicherung
Database Path	Pfad	Hauptverzeichnis der ASCET-Datenbanken
Multiple user handling	aktiviert / deaktiviert	Schaltet die Benutzerauswahl ein/aus.
Send E-Mail	aktiviert / deaktiviert	Schaltet die automatische E-Mail-Übertragung ein/aus.
E-Mail Address	Adresse	E-mail-Adresse der Hotline

Mit der Option **Automatic Save** können Sie die automatische Speicherfunktion aktivieren und ein gewünschtes Zeitintervall eingeben. Je nach eingegebenem Zeitintervall werden Ihre Daten dann gespeichert.

Einige Anwender-Operationen, die einen Eintrag in die Datenbank bedeuten, werden aus Performance-Gründen nur in einem Cache gehalten. Über **Automatic Save** kann nun erzwungen werden, dass der Cache der Datenbank zyklisch gespeichert wird.

#### Setzen der automatischen Sicherung:

---

- Aktivieren Sie die Option **Automatic Save**. Die automatische Sicherung ist eingeschaltet.
- Geben Sie im Feld „Store every ... minutes“ einen Wert ein

oder



- benutzen Sie die Pfeil-Schaltflächen, um die Zeitspanne zwischen zwei Sicherungen festzulegen.

Der Cache der Datenbank wird regelmäßig im angegebenen Zeitintervall gespeichert.

Mit der Option **Multiple user handling** legen Sie fest, ob Ihre ASCET-Installation Benutzerprofile verwalten soll. Details hierzu finden Sie in Kapitel 2.2.11 „Arbeiten mit Benutzerprofilen“.

Mit der Option **Send E-Mail** können Sie das automatische Versenden von E-Mails an die ETAS Hotline einschalten.

#### **Automatisches Versenden von E-Mails einschalten:**

---

- Aktivieren Sie die Option **Send E-Mail**.
- Geben Sie im Feld „E-Mail Address“ die aktuelle Adresse der Hotline ein.
- Klicken Sie **OK**, um die Einstellung zu übernehmen.

Wenn Sie nun **Help** → **Problem Report** aufrufen, wird automatisch eine Mail an die Hotline geschickt. Diese Funktion ist im Detail in Kapitel 2.1.4 auf Seite 34 beschrieben.

### 2.2.2 Optionen zum Erscheinungsbild

---

Im Knoten „Appearance“ setzen Sie Optionen, die das Aussehen von ASCET oder von einzelnen Editoren beeinflussen. Gleichartige Optionen sind in Unterknoten zusammengefasst.

<b>Option</b>	<b>Wert</b>	<b>Beschreibung</b>
Extended Infos in Database Browser	aktiviert/ deaktiviert	Schaltet die Anzeige von Informationen über Zugriffsrechte im Komponentenmanager ein/aus.
Show Disclaimer	aktiviert/ deaktiviert	Schaltet die Anzeige des Disclaimers beim Start von ASCET ein/aus.
List Font	Schriftart	Schrift für Listeneinträge
Text Font	Schriftart	Schrift für Text auf der Bedienoberfläche
Edit Font Mapping	Anweisungen	Abbildung von Windows-Schriftarten auf Postscript-Schriftarten

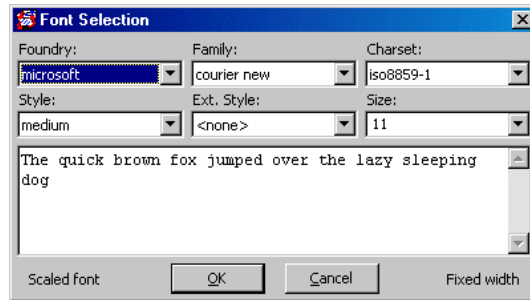
## Setzen von Schriftarten:

---



- Klicken Sie auf die Schaltfläche neben der Schriftart, die Sie ändern wollen.

Das Fenster „Font Selection“ öffnet sich.



- Wählen Sie im Kombikästchen „Foundry“ die Herkunft der Schrift.
- Wählen Sie im Kombikästchen „Family“ die Schrift.
- Wählen Sie im Kombikästchen „Charset“ den Zeichensatz.
- Wählen Sie im Kombikästchen „Style“ Grad und Neigung der Schrift.
- Wählen Sie im Kombikästchen „Size“ die Schriftgröße.

Die verfügbaren Größen hängen von der ausgewählten Schrift ab.

Das Textfeld unter den Kombikästchen zeigt einen Beispielsatz in der aktuell ausgewählten Schriftart. Im Kombikästchen „Ext. Style“ sowie rechts und links der Schaltflächen **OK** und **Cancel** sind weitere Informationen zur gewählten Schrift angegeben.

- Klicken Sie **OK**, um die Auswahl zu übernehmen und das Fenster „Font Selection“ zu schließen.

Die Schriftauswahl wirkt sich nicht auf vorhandene Fenster aus.

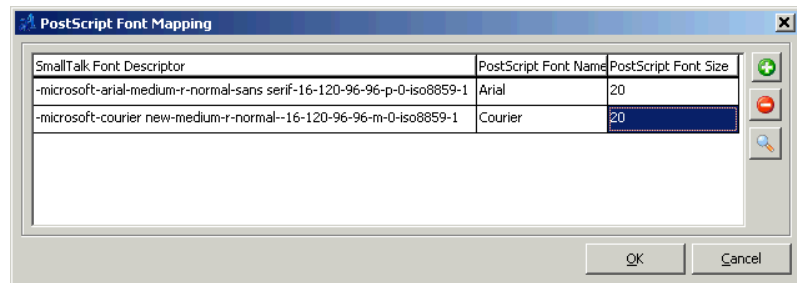
## Schriftarten abbilden:

Mit der Option „Edit Font Mapping“ können Sie die Windows-Schriftarten auf Postscript-Schriftarten abbilden. Beim Speichern eines Blockdiagramms als Postscript-Datei werden die Windows-Schriften durch die gewählten Post-Script-Schriften ersetzt.



- Klicken Sie im Unterknoten „PostScript“ auf die Schaltfläche.

Das Fenster „PostScript Font Mapping“ öffnet sich. Es zeigt in einer Tabelle die vorhandenen Abbildungsanweisungen für Schriften.



### Spalte

SmallTalk Font Descriptor

PostScript Font Name

PostScript Font Size

### Bedeutung

Windows-Schrift (SmallTalk)

PostScript-Schrift

Größe der PostScript-Schrift

1. Abbildungsanweisung erzeugen/löschen



- Klicken Sie im Fenster „PostScript Font Mapping“ auf die Schaltfläche **Add new Mapping Entry**, um eine neue Abbildungsanweisung zu erzeugen.

### Hinweis

Sie können eine Windows-Schrift auf mehrere PostScript-Schriften abbilden. Bei der Generierung von PostScript-Dateien wird jedoch nur die erste Abbildungsanweisung für eine Schrift verwendet, die anderen bleiben wirkungslos.



- Klicken Sie im Fenster „PostScript Font Mapping“ auf die Schaltfläche **Remove selected Mapping Entry**, um eine Abbildungsanweisung zu entfernen.

## 2. Abbildungsanweisung bearbeiten

- Doppelklicken Sie im Fenster „PostScript Font Mapping“ in eine Zelle der Spalte „SmallTalk Font Descriptor“.

Das Fenster „Font Selection“ (siehe Seite 44) öffnet sich.

- Stellen Sie im Fenster „Font Selection“ die Windows-Schriftart ein, die sie auf eine PostScript-Schrift abbilden wollen, und klicken Sie **OK**.
- Doppelklicken Sie im Fenster „PostScript Font Mapping“ in eine Zelle der Spalte „PostScript Font Name“ und geben Sie die PostScript-Schrift ein, die die Windows-Schrift ersetzen soll.
- Doppelklicken Sie im Fenster „PostScript Font Mapping“ in eine Zelle der Spalte „PostScript Font Size“ und geben Sie die Größe für die PostScript-Schrift ein.




- Klicken Sie im Fenster „PostScript Font Mapping“ auf die Schaltfläche **Preview selected Mapping Entry**, um den PostScript-Befehl anzuzeigen, der mit den gewählten Einstellungen generiert wird.

## 3. Arbeit abschließen

- Schließen Sie das Fenster „PostScript Font Mapping“ mit **OK**, um die Einstellungen zu übernehmen.
- Schließen Sie das Fenster „PostScript Font Mapping“ mit **Cancel**, um die Einstellungen zu verwerfen.

## Optionen für Bestätigungsfenster

Bei der Arbeit mit ASCET erscheint an verschiedenen Stellen ein Bestätigungsfenster, sie Sie auf Wunsch ausblenden können. Im Knoten „Confirmation Dialogs“ des Dialogfensters „Options“ können Sie ausgeblendete Bestätigungsfenster (Symbol ) wieder anzeigen. Ebenso können Sie hier Bestätigungsfenster ausblenden.

Dieser Knoten ist primär dazu gedacht, Bestätigungsfenster wieder anzuzeigen, die Sie während der laufenden Arbeit ausgeblendet haben. An dieser Stelle Bestätigungsfenster auszublenden, die Sie noch nicht im Programm gesehen haben, wird *ausdrücklich nicht empfohlen*.

Spalte „Dialog“	Spalte „Decision“	Beschreibung
Change Database Path	Show / Yes / No	Bestätigungsfenster für den Wechsel des Datenbankpfads beim Öffnen einer nicht im Standardverzeichnis befindlichen Datenbank.
Close ASCET	Show / OK	Bestätigungsfenster beim Beenden von ASCET.
Close Boolean Table	Show / OK	Bestätigungsfenster beim Schließen einer Booleschen Tabelle mit fehlerhaften Einträgen.
Close existing Editor	Show / OK	Bestätigungsfenster beim wiederholten Öffnen eines Editors für dieselbe Komponente.
Deassign Application Mode <inactive>	Show / OK	Bestätigungsfenster, wenn im OS-Editor der Typ einer Init-Task mit Betriebsmodus <code>inactive</code> geändert wird.
Edit complex Element	Show / Hide	Legt fest, ob der Elementeditor für komplexe Elemente automatisch geöffnet wird (Show) oder nicht (Hide).
Edit CT Element	Show / Hide	Legt fest, ob der Elementeditor für CT-Blöcke automatisch geöffnet (Show) wird oder nicht (Hide).
Edit Implementation Cast	Show / Hide	Legt fest, ob beim Erzeugen eines neuen Implementation-Casts der Elementeditor geöffnet wird (Show) oder nicht (Hide).
Edit primitive Element	Show / Hide	Legt fest, ob beim Erzeugen eines neuen Elements (Variable, Parameter, Kennlinie/-feld) der Elementeditor geöffnet wird (Show) oder nicht (Hide).

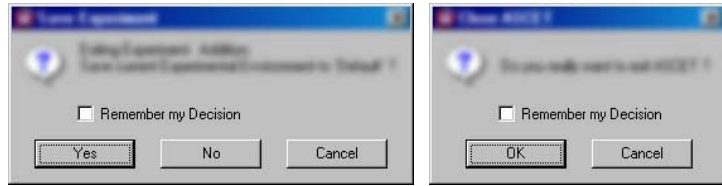
Spalte „Dialog“	Spalte „Decision“	Beschreibung
Export Components	Show / OK	Bestätigungsfenster beim Export mit aktivierter Option <b>One File for Each Item</b> .
Export Message Configuration	Show / OK	Bestätigungsfenster beim Export von Meldungskonfigurationen (siehe Kapitel 2.4.2)
Import Options	Show / OK	Bestätigungsfenster beim Import von Optionen
Import Views	Show / OK	Bestätigungsfenster beim Import von Sichten
Overwrite Files/Folders	Show / OK	Bestätigungsfenster beim Import bereits vorhandener Komponenten
Resize Drawing Area	Show / Yes / No	Bestätigungsfenster, wenn die Größe des Zeichenbereichs beim Öffnen eines Blockdiagramms geändert werden soll.
Save Experiment	Show / Yes / No	Bestätigungsfenster für das Abspeichern beim Verlassen des Experiments.
Set Layout to Default	Show / OK	Bestätigungsfenster, wenn das Layout einer aufgenommenen Komponente als Standardlayout gesetzt wird (s. Seite 246).
Set Options to Default	Show / OK	Bestätigungsfenster beim Zurücksetzen von Optionen.
Set Target Settings to Default	Show / OK	Bestätigungsfenster, wenn ein Projekt geöffnet wird, dessen Target oder Compiler ASCET nicht bekannt ist.
Start Offline Simulation	Show / OK	Bestätigungsfenster, wenn ein Offline-Experiment gestartet wird, ohne dass im Ereignisgenerator ein Ereignis ist.
Store Changes in Graphic	Show / Yes / No	Bestätigungsfenster, wenn ein Blockdiagramm mit ungesicherten Änderungen geschlossen wird.

Die möglichen Werte einer Option hängen vom Bestätigungsfenster ab.

- `Show` ist immer vorhanden; wenn `Show` gewählt ist, wird das Bestätigungsfenster angezeigt.
- `Yes` und `No` sind vorhanden, wenn das Bestätigungsfenster die Schaltflächen **Yes** und **No** enthält.



- OK ist vorhanden, wenn das Bestätigungsfenster die Schaltfläche **OK** enthält.

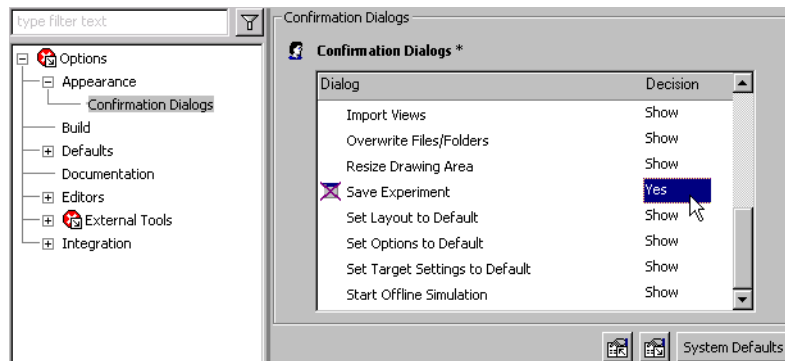


Yes, No und OK blenden das Fenster aus; die weitere Wirkung entspricht der Wirkung der gleichnamigen Schaltfläche.

- Hide ist bei den Optionen vorhanden, die das automatische Öffnen des Elementeditors für neue Elemente steuern. Hide verhindert das automatische Öffnen des Elementeditors.

### Bestätigungsfenster anzeigen/verstecken:

- Öffnen Sie den Knoten „Confirmation Dialogs“ im ASCET-Optionsfenster.
- Doppelklicken Sie im Feld „Confirmation Dialogs“ auf den Eintrag der Spalte „Decision“ neben der gewünschten Option.



- Wählen Sie aus dem Kombikästchen einen Eintrag.

Save Experiment  
 Set Layout to Default  
 Set Options to Default



Das Bestätigungsfenster wird ab sofort gemäß der gewählten Einstellung behandelt.

### 2.2.3 Build-Optionen

Im Knoten „Build“ setzen Sie Optionen, die die Codegenerierung in ASCET beeinflussen.

Option	Wert	Beschreibung
Code Preview Path	Pfad	Verzeichnis der generierten C-Code- oder HTML-/XML-Dateien
Code Generation Path	Pfad	Zielverzeichnis für die Codegenerierung, hier werden die bei der Codegenerierung erzeugten Dateien abgelegt. Standard: ETAS\Ascet5.2\cgen
Target Root Path	Pfad	Target-Hauptverzeichnis; hier werden die Unterverzeichnisse für die verschiedenen Targets angelegt.
Specific Path	Pfad	Einstellbarer Pfad für Projektdateien
Write Project Files on Build	aktiviert / deaktiviert	Legt fest, ob Projektdateien bei der Codegenerierung auf die Festplatte geschrieben werden.
Keep files in Code Generation Directory	aktiviert / deaktiviert	Gibt an, ob das Codegenerierungsverzeichnis (Option „Code Generation Path“) beim Verlassen von ASCET behalten oder gelöscht werden soll.

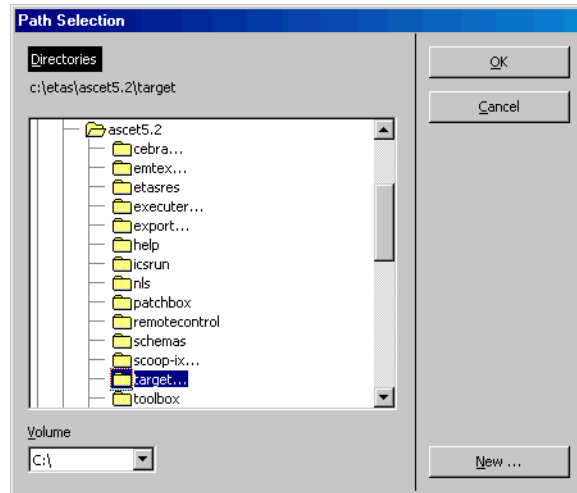
## Setzen eines Pfades:

---



- Wenn Sie einen Pfad ändern wollen, klicken Sie auf die Schaltfläche neben der Pfadoption, die Sie ändern wollen.

Das Fenster „Path Selection“ öffnet sich.



- Wählen Sie bei Bedarf im Kombikästchen „Volume“ ein Laufwerk.
- Wählen Sie in der Liste „Directories“ ein vorhandenes Verzeichnis

oder

- erzeugen Sie mit der Schaltfläche **New** ein neues Verzeichnis.
- Klicken Sie auf **OK**.

Das gewählte Verzeichnis wird im Optionsfenster angezeigt.

## 2.2.4 Standardoptionen

Im Knoten „Defaults“ können Sie Standardeinstellungen für neu angelegte Projekte (auch Standardprojekte) sowie für Implementierungen setzen. Die Standardeinstellungen für Implementierungen sind im Unterknoten „Implementation“ zusammengefasst.

Option	Wert	Beschreibung
Use Arithmetic Service	aktiviert / deaktiviert	Schaltet für neue Projekte die Verwendung arithmetischer Dienste ein/aus.
Use first available Service set	aktiviert / deaktiviert	Weist neuen Projekten den ersten Satz arithmetischer Dienste in der Datei <code>services.ini</code> zu.
Use Project Options Template	aktiviert / deaktiviert	Schaltet die Verwendung einer XML-Datei als Vorlage für die Projekteinstellungen neuer Projekte ein/aus.
Project Options Template	Datei	Name und Pfad der XML-Datei, die als Vorlage für die Projekteinstellungen genutzt wird (nur verfügbar, wenn <b>Use Project Options Template</b> aktiviert ist).

### Standardeinstellung für Arithmetische Dienste:

Um die Verwendung arithmetischer Dienste für neu angelegte Projekte zu aktivieren, gehen Sie wie folgt vor.

- Aktivieren Sie im Knoten „Defaults“ die Option **Use Arithmetic Service**.  
Damit ist die Verwendung arithmetischer Dienste für neu angelegte Projekte aktiviert.
- Aktivieren Sie die Option **Use first available Service Set**.  
Ein neu angelegtes Projekt verwendet den ersten Satz arithmetischer Dienste der Datei `services.ini`.
- Klicken Sie **OK**, um die Einstellung zu übernehmen.

Details zu arithmetischen Diensten finden Sie in Kapitel 4.14.

## Implementierungsoptionen

Option	Wert	Beschreibung
Limit to maximum bit length	aktiviert/deaktiviert	Legt fest, ob das Ergebnis einer Operation im Fall eines Überlaufs begrenzt werden soll.
Implementation Master	Model / Implementation	Master für die Implementierung
Resolution Handling	Automatic / Reduce Resolution / Keep Resolution	Legt fest, wie die Auflösung im Fall eines Überlaufs behandelt wird.
Minimum cont Data Type	real64/ real32/ uint8/ int8/ uint16/ int16/ uint32/ int32	minimaler Typ für Implementierungen von <code>cont</code> -Größen
Minimum log Data Type	bit/ uint8/ sint8/ uint16/ int16/ uint32/ int32	minimaler Typ für Implementierungen von <code>log</code> -Größen
Minimum sdisc Data Type	uint8/ int8/ uint16/ int16/ uint32/ int32	minimaler Typ für Implementierungen von <code>sdisc</code> -Größen
Minimum udisc Data Type	uint8/ int8/ uint16/ int16/ uint32/ int32	minimaler Typ für Implementierungen von <code>udisc</code> -Größen
Default cont Data Type	real64/ real32/ uint8/ int8/ uint16/ int16/ uint32/ int32	Standardtyp für Implementierungen von <code>cont</code> -Größen
Default log Data Type	bit/ uint8/ int8/ uint16/ int16/ uint32/ int32	Standardtyp für Implementierungen von <code>log</code> -Größen
Default sdisc Data Type	uint8/ int8/ uint16/ int16/ uint32/ int32	Standardtyp für Implementierungen von <code>sdisc</code> -Größen
Default udisc Data Type	uint8/ int8/ uint16/ int16/ uint32/ int32	Standardtyp für Implementierungen von <code>udisc</code> -Größen

## Einstellen von Standardoptionen für Implementierungen:

---

- Aktivieren Sie die Option **Limit to maximum bit length**, wenn das Ergebnis einer Operation im Fall eines Überlaufs begrenzt werden soll.

- Wählen Sie im Kombikästchen „Implementation Master“ die Masterseite für neu angelegte Implementierungen.

Wenn Sie eine neue Implementierung erzeugen, wird die gewählte Seite im Implementierungseditor automatisch als Master gesetzt.

- Wählen Sie im Kombikästchen „Minimum <type> Data Type“ einen minimalen Implementierungsdatentyp für Größen vom Typ <type>.
- Wählen Sie im Kombikästchen „Default <type> Data Type“ den Standard-Implementierungsdatentyp für Größen vom Typ <type>.

<type> steht für `cont`, `log`, `sdisc` oder `udisc`.

Die gewählten Typen sind voreingestellt, wenn Sie die Implementierungen entsprechender Größen bearbeiten.

Wenn Sie eine Datenbank aus einer früheren ASCET-Version öffnen, wird für `cont`-Größen `real64`, für `sdisc`-Größen `int8`, für `udisc`-Größen `uint8` und für `log`-Größen `int8` ausgewählt.

- Klicken Sie **OK**, um die Einstellung zu übernehmen.

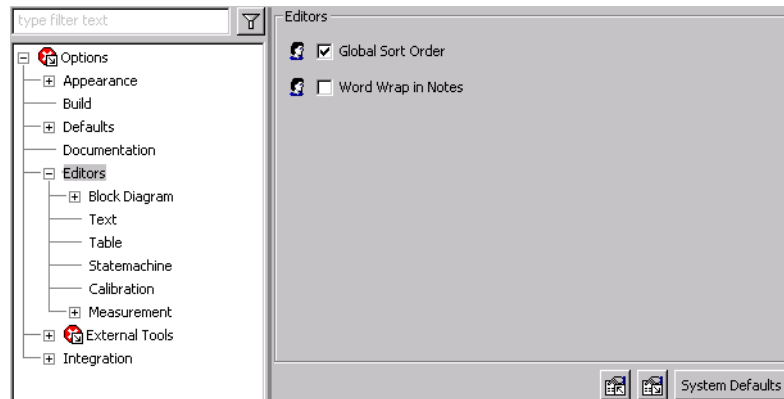
## 2.2.5 Optionen für die automatische Dokumentation

Im Knoten „Documentation“ können Sie globale Einstellungen für die automatische Dokumentation (Details siehe Kapitel 7) vornehmen.

Option	Wert	Beschreibung
Documentor Working Path	Pfad	Dokumentationsverzeichnis: Zielverzeichnis für die erzeugte Dokumentation.
Scale EPS Graphic	aktiviert / deaktiviert	Legt fest, ob die mit <b>View</b> → <b>Save as Postscript</b> erzeugte EPS-Grafik auf die Papiergröße (s. Seite 58) skaliert wird.

## 2.2.6 Optionen für Editoren

Im Knoten „Editors“ können Sie Einstellungen für verschiedene Editoren sowie für Mess- und Verstellfenster vornehmen. Einstellungen, die nur zu einem Editor gehören, sind in Unterknoten zusammengefasst.



Option	Wert	Beschreibung
Global Sort Order	aktiviert / deaktiviert	Schaltet die Verwendung einer gemeinsamen Sortierreihenfolge in der Elementliste aller Editoren ein/aus.
Word Wrap in Notes	aktiviert / deaktiviert	Zeilenumbruch im Notizeneditor

## Optionen für Blockdiagramme

---

Die Optionen im Knoten „Block Diagram“ betreffen alle Blockdiagramme.

Option	Wert	Beschreibung
Activate flexible layout	aktiviert / deaktiviert	Legt fest, ob das Layout einer aufgenommenen Komponente im Blockdiagrammeditor geändert werden kann oder nicht.
Size of Undo Buffer	Zahl	Anzahl Arbeitsschritte im Undo-Puffer von Blockdiagramm- und Zustandsautomaten-Editor
Drawing Area Size (px)	2000@2000 / 5000@5000 / 10000@10000	Größe des Zeichenbereichs in Pixeln.
Selection Marker Size (Pixel)	Zahl	Größe der Auswahlmarkierungen in Blockdiagrammen
Show Watermark	aktiviert / deaktiviert	Legt fest, ob der Name der gewählten Sicht als Wasserzeichen im Blockdiagramm gezeigt wird.
Watermark Size (px)	Zahl	Schriftgröße für das Wasserzeichen
Show Page Number	aktiviert / deaktiviert	Zeigt die Seitenzahl im Zeichenbereich an.
BDE Graphical Comments	Schriftart	Schrift für Benutzerkommentare in Blockdiagrammen
BDE Graphical Names	Schriftart	Schrift für Namen von grafischen Elementen in Blockdiagrammen

### Flexibles Layout aktivieren (global):

---

- Aktivieren Sie im Knoten „Block Diagram“ die Option **Activate flexible layout**.

Sie können nun das Layout aufgenommenener Komponenten im Blockdiagramm bzw. im Register „Graphics“ des Projekteditors bearbeiten.

Wie Sie Schriftarten setzen, ist auf Seite 44 beschrieben.



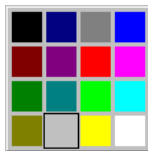
**Knoten „Colors“:** Die Farbeinstellungen für Blockdiagramme sind im Optionsfenster im Knoten „Colors“ zusammengefasst.

Option	Wert	Beschreibung
Background Color	Farbe	Hintergrundfarbe im Blockdiagrammeditor und im Projekteditor, Register „Graphics“
Grid Color	Farbe	Farbe des Rasters in Blockdiagrammen
Comment Line Color	Farbe	Farbe von Kommentarlينien in Blockdiagrammen oder Projekten
Selection Marker Color	Farbe	Farbe für die Auswahlmarkierungen in Blockdiagrammen
Watermark Color	Farbe	Farbe für das Wasserzeichen, d.h. den Namen der ausgewählten Sicht, sowie für Seitenzahlen und Markierung der Druckseiten.

### Festlegen von Farben:

---

Sie können verschiedene Farbeinstellungen für Blockdiagramme sowie für den Eingangsbereich (Eingänge bei Booleschen Tabellen, s. Kapitel 4.6, und Bedingungsbereich bei Bedingungstabellen, s. Kapitel 4.7) und Ausgangsbereich (Ausgänge bei Booleschen Tabellen und Aktionsbereich bei Bedingungstabellen) festlegen.



- Wählen Sie das Kombikästchen der Option, die Sie ändern wollen.  
Ein Farbauswahl-Fenster wird angezeigt. Die aktuelle Farbe ist markiert.
- Wählen Sie die gewünschte Farbe.  
Damit die Farbauswahl wirksam wird, müssen Sie ein vorhandenes Fenster schließen und wieder öffnen.

**Knoten „Sequencing“:** In diesem Knoten sind Optionen für Sequenzauf-rufe (siehe Kapitel 4.1.6) zusammengefasst.

Option	Wert	Beschreibung
Sequence Shift Offset	Zahl	Wert für die Verschiebung der Nummern von Sequenzaufrufen
Sequence Step Size	Zahl	Schrittweite bei der automatischen Skalierung der Nummern von Sequenzaufrufen
Use gaps	aktiviert / deaktiviert	Füllt bei automatischer Nummerierung von Sequenzaufrufen die Lücken der vorhandenen Nummernfolge.

### Setzen von Sequenzierungsoptionen:

- Stellen Sie im Feld „Sequence Step Size“ den Wert für die Schrittweite bei der automatischen Skalierung der Nummern von Sequenzaufrufen ein.
- Stellen Sie im Feld „Sequence Shift Offset“ den Wert für die automatische Verschiebung der Nummern von Sequenzaufrufen ein.
- Aktivieren Sie die Option **Use gaps**, wenn bei der automatischen Nummerierung von Sequenzaufrufen Lücken in der vorhandenen Nummernfolge aufgefüllt werden.

**Knoten „Paper Size“:** Dieser Knoten enthält die Optionen, die den Druck von Blockdiagrammen steuern.

Option	Wert	Beschreibung
Paper Size	A5 (148x210)/ A4 (210x297)/ Letter (216x279)/ Legal (216x355)/ Userdefined	Legt die Größe des Druckbereichs fest. Wenn <code>Userdefined</code> gewählt ist, kann die Größe in den beiden folgenden Optionen beliebig eingestellt werden.
User Size Width (mm)	Zahl	Benutzerdefinierter Druckbereich: Breite
User Size Height (mm)	Zahl	Benutzerdefinierter Druckbereich: Höhe
Paper Orientation	Landscape / Portrait	Orientierung des Druckbereichs.

### Optionen für Texteditoren

---

Die Optionen im Knoten „Text“ betreffen C-Code- und ESDL-Editor, jedoch nicht den Zustands- oder Übergangseditor.

Option	Wert	Beschreibung
Tabulator Size	Zahl	Größe des Tabulators in den Texteditoren
Code Font	Schriftart	Schrift für Texte in ESDL- und C-Code-Editoren

Wie Sie Schriftarten setzen, ist auf Seite 44 beschrieben.

### Optionen für Tabelleneditoren

---

Die Optionen im Knoten „Table“ betreffen den Booleschen Tabelleneditor und den Editor für Bedingungsstabellen.

Option	Wert	Beschreibung
Input Color	Farbe	Hintergrundfarbe für den Eingangsbereich einer Tabelle
Output Color	Farbe	Hintergrundfarbe für den Ausgangsbereich einer Tabelle

Wie Sie Farben festlegen, ist auf Seite 57 beschrieben.

### Optionen für Zustandsautomaten

---

Im Knoten „Statemachine“ legen Sie Standardeinstellungen für Zustandsautomaten fest.

Option	Wert	Beschreibung
Use ESDL as default for state machine	aktiviert / deaktiviert	Standard für Aktionen von Zuständen und Übergängen
Animated States Color	Farbe	Farbe von animierten Zuständen beim Experimentieren

#### Einstellen von Standardoptionen für Zustandsautomaten:

---

- Aktivieren Sie die Option **Use ESDL as default for state machine**.
- Wählen Sie die Farbe für animierte Zustände gemäß Seite 57.

- Klicken Sie **OK**, um die Einstellung zu übernehmen.

Wenn Sie in einem Zustandsautomaten einen neuen Zustand oder Übergang anlegen, wird ESDL als Standard für die Übergangs- und Zustandseditoren verwendet.

Wie Sie Farben festlegen, ist auf Seite 57 beschrieben.

### *Optionen für Verstellfenster*

---

Im Knoten „Calibration“ legen Sie Standardeinstellungen für Verstellfenster fest. Diese und weitere Einstellungen können auch für jedes Verstellfenster einzeln gesetzt werden.

Die Bedeutung der Optionen entnehmen Sie den Kurzbeschreibungen.

### *Optionen für Messfenster*

---

Im Knoten „Measurement“ legen Sie Standardeinstellungen für Messfenster fest. Diese und weitere Einstellungen können auch für jedes Messfenster einzeln gesetzt werden.

Die Bedeutung der Optionen entnehmen Sie den Kurzbeschreibungen.

**Knoten „Datalogger“:** Dieser Knoten enthält Optionen für den Data-Logger (s. Kapitel 6.1.9).

Option	Wert	Beschreibung
Auto Increment Data-Logger File Name	aktiviert / deaktiviert	Legt fest, ob die Aufzeichnungsdatei des Dataloggers fortlaufend nummeriert (aktiviert) oder bei jeder Aufzeichnung überschrieben wird.

## 2.2.7 Experiment-Optionen

---

Im Knoten „Experiment“ sind Optionen für Online- und Offline-Experimente enthalten.

Option	Wert	Beschreibung
Initialize variables at OS start	aktiviert / deaktiviert	Legt fest, ob die Variablen bei jedem Start der Simulation (Offline-Experiment) bzw. des Betriebssystems (Online-Experiment) initialisiert werden.
Initialize parameters at OS start	aktiviert / deaktiviert	Legt fest, ob die Parameter bei jedem Start der Simulation (Offline-Experiment) bzw. des Betriebssystems (Online-Experiment) initialisiert werden.

## 2.2.8 Optionen für externe Werkzeuge

Im Knoten „External Tools“ sind Optionen für externe Werkzeuge enthalten. Einige externe Werkzeuge haben eigene Unterknoten, z.B. die Compiler und der Texteditor, der aus C-Code- und ESDL-Editor oder aus dem ASCET-Monitorfenster aufgerufen werden kann. Andere Optionen sind direkt im Knoten „External Tools“ untergebracht.

Option	Wert	Beschreibung
Perl Compiler	Datei	Perl-Compiler
External Make Utility	Datei	externe Make-Datei
Output Redirector Utility	Datei	Umlenken der Ausgabe
C Preprocessor	Datei	C-Code-Präprozessor
C Code Beautifier	Datei	reformatiert den generierten Code nach Benutzervorgaben

### Setzen der Pfade für externe Werkzeuge:



- Klicken Sie im Knoten „External Tools“ auf die Schaltfläche neben der Option, die Sie ändern wollen.

Die Windows-Dateiauswahl erscheint.

- Wählen Sie das Verzeichnis, in dem sich das Werkzeug befindet.
- Wählen Sie die gewünschte Datei an und klicken Sie auf **Öffnen**.

Dateiname und -Pfad werden rechts neben der Schaltfläche angezeigt.

### Optionen für den ASCII Editor

Option	Wert	Beschreibung
Use external ASCII Editor	aktiviert/ deaktiviert	Schaltet die Verwendung eines externen Texteditors ein/aus.
ASCII Editor	Datei	Name und Pfad des externen Texteditors

Option	Wert	Beschreibung
Commandline Arguments	Zeichenkette	Kommandozeilen-Argumente für den externen Texteditor; erstes Argument: Dateiname, zweites Argument: Zeilennummer
Begin Region Marker	Zeichenkette	Markierung für denTexteditor, die den Beginn einer Region kennzeichnet.
End Region Marker	Zeichenkette	Markierung für denTexteditor, die das Ende einer Region kennzeichnet.

Die Kombikästchen für Kommandozeilen-Argumente und Region-Marker bieten voreingestellte Werte für einige Texteditoren an.

### *Compiler-Optionen*

Im Knoten „Compiler“ finden Sie Unterknoten für die verfügbaren Compiler. Standardmäßig sind das Borland-C V4.5, Borland-C V5.5 und Microsoft Visual C++. Wenn Sie ASCET-RP oder ASCET-SE installieren, kommen weitere hinzu. Die Optionen sind – bis auf die letzte – für alle Compiler verfügbar.

Option	Wert	Beschreibung
Tool Root Path	Pfad	Installationsverzeichnis des Compilers
Private directive	Anweisung	Compiler-Anweisung für private Funktionen
Public directive	Anweisung	Compiler-Anweisung für öffentliche Funktionen
Unique identifier	String	Eindeutiger Identifier für den Compiler
Object file extension	Dateiendung	Dateierweiterung für Objektdateien
Result file extension	Dateiendung	Dateierweiterung für ausführbare Dateien
Supports long filenames	aktiviert / deaktiviert	Zeigt an, ob der Compiler lange Dateinamen unterstützt.
Inline directive	Anweisung	Compiler-Anweisung für Inline-Funktionen
Change CYGWIN Registry Settings		<i>Nur GNU-C V2.95.3 (PowerPC)</i> ; ändert die Registry-Einträge für Cygwin, um korrekte Zusammenarbeit mit dem GNU-Compiler zu gewährleisten.
Supports precompiled header	aktiviert / deaktiviert	Nicht einstellbar. Zeigt an, ob der Compiler vorcompilierte Header unterstützt.

## 2.2.9 Optionen für die Integration

Im Knoten „Integration“ finden Sie Optionen für Import/Export, Datenaustausch usw. Zusammengehörige Optionen sind in Unterknoten zusammengefasst.

Option	Wert	Beschreibung
Tool Server Activation	aktiviert / deaktiviert	Für den internen Gebrauch; sollte immer aktiviert sein.
Use .NET based Tool API	aktiviert / deaktiviert	Schaltet zwischen der aktuellen .NET-basierten Tool-API und der (veralteten) MS J++ Tool-API um.

### *Exportoptionen*

Im Knoten „Export“ können Sie bestimmen, in welcher Art und in welchem Umfang die gewählten Datenbankobjekte exportiert werden. So können Sie einzelne Datensätze als eigene Datei ablegen und einstellen, dass die referenzierten Einträge mit exportiert werden sollen.

Desweiteren können Sie einstellen, ob beim Export von Datenbankelementen eine Beschreibungsdatei angelegt wird. Ebenso können Format und Inhalt dieser Datei bestimmt werden.

Option	Wert	Beschreibung
Default Export Path	Pfad	Standardverzeichnis für Export
Include Referenced Items	aktiviert / deaktiviert	Schaltet rekursiven Export ein/aus.
One File for each Item	aktiviert / deaktiviert	Schaltet Export in einzelne Dateien ein/aus.
Write ASCII file on Binary Export	aktiviert / deaktiviert	Schaltet die Erzeugung von zusätzlichen ASCII-Beschreibungsdateien ein/aus.
Write XML file on Binary Export	aktiviert / deaktiviert	Schaltet die Erzeugung von zusätzlichen XML-Beschreibungsdateien ein/aus.
Create HTML File	aktiviert / deaktiviert	Erzeugt eine HTML-Datei für die XML-Beschreibungsdatei.
Write Default Project Info	aktiviert / deaktiviert	Nimmt Informationen zum Standardprojekt in die XML-Beschreibungsdatei auf.
Write Version Info	aktiviert / deaktiviert	Nimmt Versionsinformationen in die XML-Beschreibungsdatei auf.
Write Graphic Specification	aktiviert / deaktiviert	Nimmt grafische Informationen in die XML-Beschreibungsdatei auf.

Option	Wert	Beschreibung
Sort XML Tags	aktiviert/ deaktiviert	Schaltet die alphabetische Sortierung der Tags in der XML-Beschreibungsdatei ein/aus.
Create complete Hierarchy	aktiviert/ deaktiviert	Legt fest, ob beim AMD-Export die exportierten Dateien in einer Verzeichnisstruktur abgelegt werden, die die Ordnerstruktur der ASCET-Datenbank nachbildet.
Export Experiment Environment	aktiviert/ deaktiviert	Legt fest, ob die Experimentierumgebungen zusammen mit der Komponente exportiert werden.
Encryption Key	Schlüssel	Schlüssel zur Dateiverschlüsselung beim AMD-Export

### Setzen von Export-Optionen:

- Wählen Sie im Feld „Default Export Path“ das Standard-Zielverzeichnis für den Export.
- Aktivieren Sie die Option **Include Referenced Items**, um Datenbankeinträge rekursiv (mit allen referenzierten Einträgen) zu exportieren.
- Aktivieren Sie die Option **One File for each Item**, um jeden exportierten Datenbankeintrag in eine eigene Datei zu schreiben.
- Aktivieren Sie die Option **Write ASCII file for Binary Export**, wenn Sie eine zusätzliche ASCII-Beschreibungsdatei erzeugen wollen.
- Aktivieren Sie die Option **Write XML file for Binary Export**, wenn Sie eine zusätzliche XML-Beschreibungsdatei erzeugen wollen.  
Die Optionen im Feld unter **Write XML file for Binary Export** sind nun verfügbar.
  - Aktivieren oder deaktivieren Sie die Optionen, um die Inhalte der XML-Beschreibungsdatei festzulegen.
- Aktivieren oder deaktivieren Sie die Optionen im Feld „AMD Format“, um die Inhalte der AMD-Beschreibungsdatei festzulegen.

Informationen zum Exportieren von Datenbankeinträgen siehe Kapitel 2.3.3 „Exportieren von Ordnern und Datenbankeinträgen“ auf Seite 94.



## Importoptionen

---

Im Knoten „Import“ setzen Sie die Optionen für das Importieren von Datenbankeinträgen. Optionen für die automatische Reparatur bei AMD-Import sind im Unterknoten „Autofixes“ (Seite 67) zusammengefasst.

Option	Wert	Beschreibung
Default Import Path	Pfad	Standardverzeichnis für Import
Discard Existing Implementation	aktiviert/ deaktiviert	Schaltet Ersetzen aller bestehenden Implementierungen durch die importierten Implementierungen ein/aus.
Keep Folder Path in Database	aktiviert/ deaktiviert	Schaltet die Verwendung des in der Datenbank angelegten Pfades beim Import von schon vorhandenen Objekten ein/aus.
Remove Version Information	aktiviert/ deaktiviert	Schaltet die Entfernung von Informationen aus einem Versionsmanagement-Tool ein/aus.
Keep Hierarchy	aktiviert/ deaktiviert	Legt fest, ob beim XML-Import die Verzeichnisstruktur der Exportdateien in der ASCET-Datenbank nachgebildet wird.
Import referenced Items	aktiviert/ deaktiviert	Schaltet rekursiven AMD-Import ein/aus.
Remap OIDs	aktiviert/ deaktiviert	Weist importierten Elementen neue Objekt-IDs zu und erzeugt so „Kopien“ von bereits vorhandenen Elementen.
Decryption Key	Schlüssel	Schlüssel zur Datei-Entschlüsselung beim AMD-Import

### Setzen von Importoptionen:

---

- Geben Sie im Feld „Default Import Path“ das Standardverzeichnis für Importe ein.

- Aktivieren Sie die Option **Discard Existing Implementations**, um *alle* bestehenden Implementierungen durch die importierten Implementierungen zu ersetzen.

Alle ehemals existierenden Implementierungen gehen verloren, nur die Implementierungen der importierten Komponente sind vorhanden.

Diese Option ist nur relevant, wenn eine bestehende Komponente importiert wird.

- Aktivieren Sie die Option **Keep Folder Path**, um den in der Datenbank angelegten Pfad beizubehalten.

Wenn Sie diese Option deaktivieren, wird beim Import einer bestehenden Komponente der in der Exportdatei vorhandene Pfadname verwendet.

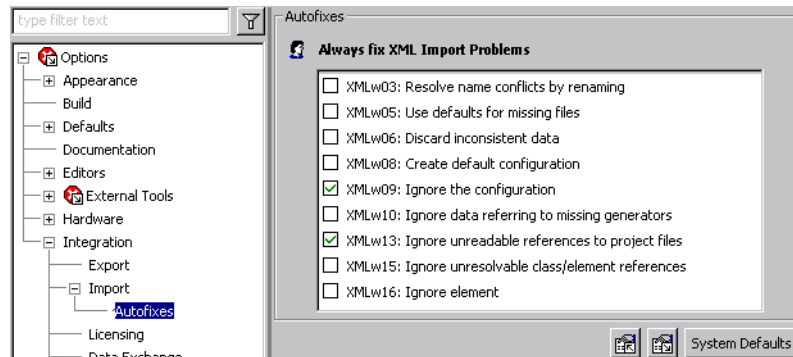
- Aktivieren Sie die Option **Remove Version Information**, um vorhandene Versionsinformationen zu entfernen.

Wenn Sie ein versioniertes Element importieren, ohne die Option zu aktivieren, können Sie es hinterher nicht in eine eigene Versionsmanagement-Datenbank ablegen, es sei denn, Sie haben Zugriff auf das Originalarchiv. Im Zweifelsfall empfiehlt es sich, die Option zu aktivieren.

- Verwenden Sie die Optionen im Feld „AMD Format“, um den AMD-Import einzurichten.

Informationen zum Importieren von Datenbankeinträgen siehe Abschnitt „Importieren von Ordnern und Datenbankeinträgen“ auf Seite 97.

**Knoten „Autofixes“:** Dieser Knoten enthält Lösungen für mögliche Probleme beim AMD-Import (siehe „Besonderheiten beim AMD-Import“). Sie können einstellen, welche Probleme ohne Anzeige automatisch behoben werden sollen.



### Lizenzierungsoptionen

Im Knoten „Licensing“ legen Sie die Ausleihdauer für Server-Lizenzen fest. Außerdem können Sie Einstellungen für das automatische Beenden von ASCET vornehmen.

Option	Wert	Beschreibung
Borrow license for days	Zahl	Ausleihdauer für Lizenzen (siehe ASCET-Schnelleinstieg, Kapitel „Lizenzierung“).
Activate idle time shut-down	aktiviert / deaktiviert	Legt fest, ob ASCET nach einer längeren Zeit ohne Aktivitäten beendet wird (aktiviert) oder nicht.
Idle period [min]	Zahl	Legt die Zeitdauer bis zum automatische Beenden von ASCET fest.

## Optionen für den Datenaustausch

Im Knoten „Data Exchange“ setzen Sie die Optionen für den Datenaustausch. Unterstützt werden die Austauschformate DCM V1.x und 2.x und Datenaustausch mit INCA und ASCET ab V4.

Option	Wert	Beschreibung
Data File Path	Pfad	Verzeichnis der Daten-Datei
Extension for Output File	dcm / dcm / kon	Dateierweiterung für die Austauschdatei
DCM Format version	DCM V1.x / DCM V2.x	Legt die Version des DCM-Formats fest.
Write Enums	aktiviert / deaktiviert	Legt fest, ob Enumerationen im TEXT-Format geschrieben werden (nur für DCM V1.x).
Write Sampling Points	aktiviert / deaktiviert	Legt fest, ob Abtastpunkte in der DCM V2.x-Syntax geschrieben werden (nur für DCM V1.x).
Case Sensitive Names	aktiviert / deaktiviert	Legt fest, ob Groß- und Kleinschreibung berücksichtigt wird. Ist die Option deaktiviert, wird z.B. beim Datenimport ein Parameter mit Namen CONT auf einen Parameter mit Namen cont abgebildet.
Hierarchical Names	aktiviert / deaktiviert	Schaltet zwischen vollständigen (aktiviert) und einfachen Elementnamen von globalen Elementen um.
Include Booleans	aktiviert / deaktiviert	Legt fest, ob Boolesche Größen mitberücksichtigt werden.
Boolean format	true / false / Integer	Legt das Format der Booleschen Größen fest (nur für DCM V2.x).
Show Log File for Load / Show Log File for Save	aktiviert / deaktiviert	Legt fest, ob die Protokolldatei für Lese-/ Schreibvorgänge angezeigt wird.
Log File for Load/ Log File for Save	Dateiname	Pfad und Name der Protokolldateien für Schreib- und Lesevorgänge

## Optionen für ausführbare Dateien

---

Im Knoten „HexFile“ befinden sich Einstelloptionen für die ausführbaren Dateien.

Option	Wert	Beschreibung
HexFile Format	IntelHex / MotorolaSRecord	Standardformat zum Exportieren einer Hex-Datei
IntelHex Record Size	16 / 32 / 64	erlaubte Anzahl der Bytes pro Feld für das IntelHex-Format
SRecord Count	aktiviert / deaktiviert	Bestimmt, ob an jedem Blockende (und vor einem eventuell folgenden Termination-Record) die Anzahl der Datenfelder geschrieben wird.
SRecord Format	16 / 24 / 32	Adressbreite in Bits im Motorola-Format
SRecord Size	16 / 24 / 32	erlaubte Anzahl der Bytes pro Feld für das Motorola-Format
SRecord Termination	aktiviert / deaktiviert	Legt fest, ob an jedem Blockende (und vor einem eventuell folgenden Termination-Record) ein Termination-Record gesetzt wird oder nicht.

Die Optionen **SRecord\*** sind nur von Bedeutung, wenn unter „HexFile Format“ der Eintrag **MotorolaSRecord** gewählt wurde.

### 2.2.10 Externe Optionen

---

Sie haben die Möglichkeit, benutzerdefinierte Optionen einzubinden. Dies geschieht über eine XML-Datei, die Sie Ihren Bedürfnissen entsprechend anpassen. Die Datei wird im ASCET-Installationsverzeichnis abgelegt; damit ASCET sie erkennt, muss sie die Endung **\*.aod.xml** haben.

Die Definition einer Option sieht grundsätzlich so aus (kursive Codeteile müssen für jede Option durch geeignete Werte ersetzt werden):

```
<OptionDeclaration
  xmlCategory="path"
  optionCategory="value"
  optionClass="type"
  attributeName="option name"
  optionFile="filename.xml">
  <Group>path</Group>
```

```

<Label>text</Label>
<Description>text</Description>
<Tooltip>text</Tooltip>
<InitialValue>value</InitialValue>
<DefaultValue>value</DefaultValue>
</OptionDeclaration>

```

Einige Optionstypen haben zusätzliche Elemente; diese sind in Tab. 2-2 beschrieben.

Die Bedeutung der Attribute und Elemente ist in Tab. 2-1 angegeben.

Attribut/Element	Bedeutung
xmlCategory <sup>a</sup>	Pfad, unter dem die Option in der XML-Datei aus optionFile abgelegt ist, z.B. Sample\Option.
optionCategory	Art, wie die Option gesichert wird (FILE – in Datei, FIXED – keine Sicherung).
optionClass	Typ der Option; mögliche Werte siehe Tab. 2-2.
attributeName	Name der Option in der XML-Datei aus optionFile. Muss in der Datei eindeutig sein.
optionFile <sup>a</sup>	XML-Datei, in der der Wert der Option abgelegt ist.
<Group>	Pfad (Knoten\Subknoten\...) der Option im ASCET-Optionsfenster; wahlweise neu oder vorhanden.
<Label>	Name der Option im ASCET-Optionsfenster.
<Description>	Kurzbeschreibung der Option.
<Tooltip>	Tooltip-Text der Option im ASCET-Optionsfenster.
<InitialValue>	Initialwert; wird nur verwendet, wenn noch kein Wert in der XML-Datei aus optionFile gespeichert ist.
<DefaultValue>	Standardwert, wird verwendet für <b>System Defaults</b> und als Initialwert, wenn <InitialValue> nicht gesetzt ist.

a: Die Werte mehrerer Optionen können in derselben XML-Datei und/oder unter demselben Pfad abgelegt werden.

**Tab. 2-1** Bedeutung der Attribute und Elemente

Folgende Typen stehen für die Definition benutzerdefinierter Optionen zur Verfügung:

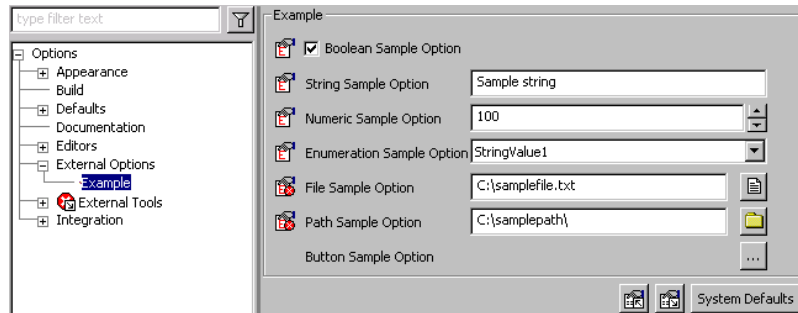
<b>optionClass</b>	<b>Erklärung</b>
EtasBooleanOption	Optionskästchen
EtasButtonOption	Schaltfläche, die einen Befehl ausführt <InitialValue> und <DefaultValue> haben den Wert ignored. zusätzliche Elemente für die Definition: <pre>           &lt;ButtonOption&gt;             &lt;Action&gt;path/executable file↵             &lt;/Action&gt;           &lt;/ButtonOption&gt;           </pre>
EtasEnumerationOption	Kombikästchen zur Auswahl einer Zeichenkette zusätzliche Elemente für die Definition: <pre>           &lt;EnumerationOption&gt;             &lt;StringValues&gt;               &lt;StringValue&gt;string1 ↵               &lt;/StringValue&gt;               ...               &lt;StringValue&gt;stringN ↵               &lt;/StringValue&gt;             &lt;/StringValues&gt;             &lt;Values&gt;               &lt;Value&gt;value1&lt;/Value&gt;               ...               &lt;Value&gt;valueN&lt;/Value&gt;             &lt;/Values&gt;           &lt;/EnumerationOption&gt;           </pre>

<b>optionClass</b>	<b>Erklärung</b>
EtasFileOption	<p>Textfeld für Pfad und Name einer Datei sowie Schaltfläche für Dateiauswahlfenster</p> <hr/> <p>zusätzliche Elemente für die Definition:</p> <pre> &lt;FileOption&gt;   &lt;DialogTitle&gt;text          ↵     &lt;/DialogTitle&gt;   &lt;SearchPath&gt;path&lt;/SearchPath&gt;   &lt;SearchMask&gt;mask&lt;/SearchMask&gt;   &lt;InvalidCharacters&gt;characters     &lt;/InvalidCharacters&gt;   &lt;FilterTypes&gt;     &lt;FilterType filter&gt;       &lt;Description&gt;text      ↵     &lt;/Description&gt;     &lt;/FilterType&gt;     ...   &lt;/FilterTypes&gt; &lt;/FileOption&gt; </pre>
EtasFloatOption	<p>Eingabefeld mit Pfeil-Schaltflächen für Float-Werte</p> <hr/> <p>zusätzliche Elemente für die Definition:</p> <pre> &lt;FloatOption&gt;   &lt;MinValue&gt;value&lt;/MinValue&gt;   &lt;MaxValue&gt;value&lt;/MaxValue&gt; &lt;/FloatOption&gt; </pre>
EtasNumericOption	<p>Eingabefeld mit Pfeil-Schaltflächen für Integer-Werte</p> <hr/> <p>zusätzliche Elemente für die Definition:</p> <pre> &lt;NumericOption&gt;   &lt;MinValue&gt;value&lt;/MinValue&gt;   &lt;MaxValue&gt;value&lt;/MaxValue&gt; &lt;/NumericOption&gt; </pre>
EtasPathOption	Textfeld für Verzeichnispfad mit Schaltfläche für Fenster „Path Selection“
EtasStringOption	Textfeld für die Eingabe einer Zeichenkette

**Tab. 2-2** Typen für benutzerdefinierte Optionen (kursive Codeteile müssen für jede Option durch geeignete Werte ersetzt werden)



Auf Ihrer Installations-CD finden Sie die Beispieldatei `externalOptionsExample.aod.xml`. Diese Datei definiert die abgebildeten Optionen.



### 2.2.11 Arbeiten mit Benutzerprofilen

In ASCET können mehrere Benutzerprofile eingerichtet und verwaltet werden. Über diese Funktionalität erhält der Anwender die Möglichkeit, spezifische, nur für ihn geltende, Einstellungen in ASCET vorzunehmen. Wenn z.B. mehrere Anwender auf einem Rechner arbeiten, kann jeder Anwender eine an seine Anforderungen angepasste Konfiguration von ASCET in seiner persönlichen Benutzerumgebung ablegen. Das sind z.B. Optionen wie Bildschirm-schriftart/-größe, diverse Vorgaben für die verschiedenen Editoren, Standardeinstellungen für Elemente usw.

#### **Hinweis**

*Auch wenn nur ein Anwender auf dem Rechner arbeitet, kann er sich mehrere Benutzerprofile einrichten. Er kann sich so z.B. ein optimiertes Benutzerprofil für das Arbeiten im Büro und eines für das Fahrzeug erstellen.*

Alle Einstellungen im jeweiligen Benutzerprofil sind global und gelten unabhängig davon, wie die konkrete Aufgabe aussieht.

Während der Programmausführung kann nur das Benutzerprofil der aktuellen Benutzerumgebung verändert werden. Das Bearbeiten von Benutzerprofilen ist nur möglich, wenn zuvor diese Funktionalität freigeschaltet wurde. Über die Funktion **Tools → Options** (Knoten „Options“, siehe Kapitel 2.2.1 auf Seite 42) können Sie bestimmen, ob die Bearbeitung von Benutzerprofilen während der Programmausführung möglich ist oder nicht. Die Auswahl eines anderen Benutzerprofils ist nur durch einen Neustart von ASCET möglich.

#### Verwenden der Benutzerauswahl

Durch die Aktivierung der Benutzerauswahl kann der Anwender einstellen, ob das Dialogfenster zur Auswahl eines Benutzerprofils erscheint.

Ist die Benutzerauswahl nicht aktiviert, verwendet das Programm automatisch die Benutzerumgebung des im System angemeldeten Benutzers.

Wenn die Benutzerauswahl aktiviert ist, wird der Anwender beim (nächsten) Programmstart aufgefordert, sich anzumelden, indem er ein bestehendes Benutzerprofil auswählt oder ein neues Profil zu ASCET hinzufügt. Diese Arbeitsschritte werden im folgenden beschrieben.

### Aktivieren der Benutzerauswahl:

---

- Wählen Sie im Komponentenmanager **Tools → Options**

oder



- klicken Sie auf die Schaltfläche **Options**.  
Das Fenster „Options“ öffnet sich.
- Aktivieren Sie im Knoten „Options“ die Option **Multiple user handling**.

Beim nächsten Start von ASCET werden Sie gebeten, entweder einen existierenden Namen auszuwählen oder einen neuen Namen einzugeben.

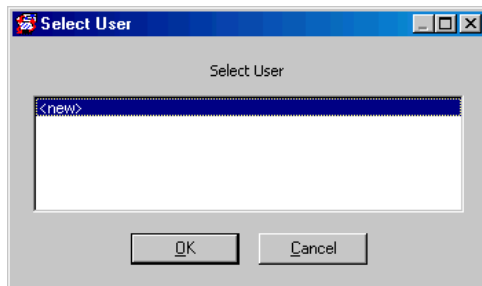
### Hinzufügen eines neuen Benutzerprofils:

---

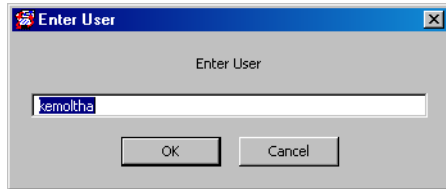
- Starten Sie ASCET mit aktivierter Benutzerauswahl im Dialogfenster „Options“.

Das Dialogfenster „Select User“ öffnet sich. Es enthält sämtliche verfügbaren Benutzer.

Beim ersten Aktivieren der Benutzerauswahl enthält die Benutzerliste nur einen Eintrag: <new>.



- Wählen Sie <new> und klicken Sie auf **OK**.  
Das Dialogfenster „Enter User“ fragt nach dem Namen für das neue Benutzerprofil.



- Geben Sie einen Namen ein und klicken Sie auf **OK**.  
Ein Profilname darf aus bis zu acht Zeichen bestehen.  
ASCET wird jetzt mit einer Kopie des Standard-Benutzerprofils (Standardeinstellungen) gestartet; der Komponentenmanager öffnet sich. Der Name des aktuellen Benutzerprofils wird in der Fußzeile des Fensters angezeigt.

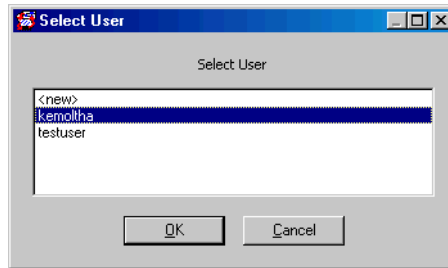
Die während einer ASCET-Sitzung vorgenommenen Benutzereinstellungen werden in dem zu Beginn des Programms ausgewählten Benutzerprofil gespeichert. Sie sind wieder gültig, wenn der entsprechende Benutzername beim Start ausgewählt wird.

Wie Sie die Benutzerumgebung Ihren individuellen Ansprüchen anpassen, lesen Sie in den Abschnitten 2.2.1 bis 2.2.10.

## Aktivieren eines bestehenden Benutzerprofils:

---

- Starten Sie ASCET mit aktiver Benutzerauswahl im Dialogfenster „Options“.  
Das Dialogfenster „Select User“ öffnet sich.



- Wählen Sie den entsprechenden Namen aus der Liste aus und klicken Sie auf **OK**.  
Der Komponentenmanager wird geöffnet. Die Benutzereinstellungen des ausgewählten Profils werden aktiv.

### Hinweis

*Benutzerprofile können nicht bei laufendem Programm gewechselt werden. Sie müssen die aktuelle Sitzung beenden und das Programm neu starten, um Benutzer zu wechseln.*

## 2.3 Komponentenmanager – Daten verwalten

---

Wie schon erwähnt, besteht die Hauptaufgabe des Komponentenmanagers darin, alle Daten, die bei der Arbeit mit ASCET entstehen (Klassen, Module, Projekte etc.), in einer Datenbank systematisch abzulegen. Der Komponentenmanager bietet Ihnen hierbei die Möglichkeit, die Objekte der Datenbank in einer übersichtlichen Oberfläche zu verwalten. Ähnlich wie im Windows-Explorer können Sie hier Ordner und Unterordner anlegen, einzelne Objekte verschieben, kopieren, importieren und exportieren, aber auch komplett neue Datenbanken anlegen. Sie können also Ihre Daten, ähnlich wie über das Dateisystem gewohnt, organisieren.

Neben der Organisation ihrer Daten ist in ASCET ein Bearbeiten der Komponenten und Projekte möglich.

## Datenbankeinträge

Eine ASCET-Datenbank enthält verschiedene Typen von Datenbankeinträgen. Dieser Abschnitt gibt einen Überblick über die in ASCET verfügbaren Typen von Datenbankeinträgen.

**Komponenten:** Die Spezifikation eines eingebetteten Softwaresystems in ASCET setzt sich aus Komponenten zusammen. Eine Komponente ist ein modulares Stück Funktionalität, das Algorithmen und Daten enthält. Die verschiedenen Algorithmen innerhalb einer Komponente können unabhängig voneinander ausgeführt werden. Die Komponenten eines Embedded Control Systems können zu Projekten zusammengefasst werden.

Auf physikalischer Ebene werden Komponenten entweder grafisch als Blockdiagramme und Zustandsautomaten oder mittels ESDL-Code spezifiziert. Alternativ können sie auch in C-Code angegeben werden. Komponenten, die grafisch oder in ESDL spezifiziert werden, sind implementierungsunabhängig, d. h. sie können verwendet werden, um Code für verschiedene Plattformen zu generieren. In C-Code spezifizierte Komponenten sind immer plattformabhängig. Sie schließen targetspezifisches Verhalten ein.

Die in ASCET verfügbaren Komponententypen und ihre Verwendung sind im Kapitel „Komponenten“ im ASCET-Referenzhandbuch beschrieben. Informationen zum Arbeiten mit Komponenten gibt es in der Beschreibung der jeweiligen Editoren.

**Projekte:** Ein Projekt spezifiziert die Funktionalität eines Embedded Control Systems. Es enthält alle Komponenten zusammen mit den nötigen Protokollen, Betriebssystem- und Codegenerierungseinstellungen. Ein Projekt legt die Kommunikation zwischen Komponenten und die Reihenfolge, in der Algorithmen abgearbeitet werden, fest.

Das Projektkonzept ist im Kapitel „Projekte“ im ASCET-Referenzhandbuch erläutert; eine Beschreibung, wie mit Projekten zu arbeiten ist, befindet sich im Kapitel „Der Projekteditor“ auf Seite 398.

**Symbole:** Wenn eine Komponente in einer anderen Komponente liegt, erscheint sie als Block im Diagramm der umgebenden Komponente. Sie können solch einer eingebundenen Komponente ein Symbol zuweisen, um die Lesbarkeit eines komplizierten Diagramms zu verbessern.

ASCET liefert einige Symbole. Zusätzlich können eigene definiert und bearbeitet werden. Importieren, Erstellen und Bearbeiten von Symbolen ist im Abschnitt „Der Symboleditor“ auf Seite 600 beschrieben.

**Signale:** Zum Erproben von ASCET-Modellen unter realistischen Bedingungen können Daten aus echten Messungen in ASCET importiert und als Eingabe beim Prüfen von ASCET-Modellen verwendet werden. Solche Daten werden in Signal-Einträgen gespeichert.

Zusätzliche Informationen zum Arbeiten mit Signalen finden sich im Kapitel „Der Signalbetrachter“ auf Seite 597.

**Container:** Container dienen als Behälter für Projekte, Klassen und Module. Ihr Zweck ist es, Modelle und Datenbanken zu strukturieren und verschiedene Datenbankobjekte unter eine gemeinsame Versionskontrolle zu stellen. Weitere Informationen finden Sie im Kapitel „Container“ auf Seite 476.

**Enumerationen:** Enumerationen sind eindeutige Typen mit Werten aus einer Menge von bekannten Konstanten, die Aufzählungskonstanten (Enumeratoren) genannt werden.

**ASAM-MCD-2MC-Projekte:** Die ASAM-MCD- 2MC-Datei stellt die Schnittstelle zwischen ASCET-MD und anderen Programmen (INCA, ASCET-RP und andere) dar, die das Standard-ASAM-MCD-Format erkennen.

### 2.3.1 Datenbankeinträge verwalten

---

In ASCET werden Datenbankeinträge in Ordnern organisiert. Eine Datenbank kann eine beliebige Anzahl von Hauptordnern enthalten, die ihrerseits andere Ordner enthalten können. Datenbankeinträge müssen in Ordnern gespeichert werden; sie können nicht auf der Hauptebene erzeugt werden. Datenbankobjekte werden über ihre Objekt-IDs unterschieden. Innerhalb eines Ordners ist Namensgleichheit zwar nicht möglich, aber Objekte aus verschiedenen Ordnern können identische Namen haben.

Dieser Abschnitt erläutert die Standardoperationen an Ordnern und Datenbankeinträgen.

#### **Anlegen eines Ordners:**

---

- Wählen Sie im Komponentenmanager in der Liste „1 Database“ den Namen der Datenbank oder den Ordner, der den neuen Ordner enthalten soll.

Sie können Ordner nur unterhalb des Datenbanknamens anlegen.

- Wählen Sie **Insert** → **Folder**

oder



- klicken Sie auf die Schaltfläche **Insert Folder**

oder

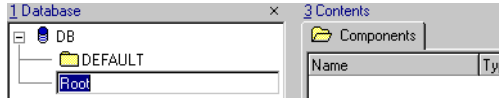
- drücken Sie die Taste <INSERT>.

Ein neuer Ordner erscheint in der Fläche „1 Database“.

Wenn Sie den neuen Ordner in der obersten

Hierarchiestufe anlegen (siehe Abbildung), wird der Name `Root` vorgegeben, andernfalls der Name `Folder`.

- Bearbeiten Sie den Ordernamen. Sie können einfach den hervorgehobenen Namen überschreiben und dann `<ENTER>` drücken.



### Hinweis

*Bei Namen von Ordnern oder Komponenten wird nicht zwischen Groß- und Kleinschreibung unterschieden. Ordernamen, die sich nur durch die Verwendung von Groß-/Kleinbuchstaben unterscheiden, sind nicht erlaubt.*

Jede Datenbank hat mindestens einen Ordner, der automatisch beim Erstellen einer Datenbank angelegt wird.

### Erstellen eines Datenbankeintrags:

- Wählen Sie in der Liste „1 Database“ den Ordner, der den neuen Eintrag enthalten soll.
- Wählen Sie **Insert** → **<component type>** bzw. **Insert** → **<component type>** → **<item type>**

oder

- klicken Sie auf die entsprechende Schaltfläche der Insert-Schaltflächenleiste.

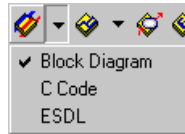
Der neue Eintrag wird mit einem Standardnamen und -Layout erzeugt.



- Bearbeiten Sie den Namen und drücken Sie `<ENTER>`.

Wenn Sie eine Komponente vom Typ Module, Class oder Continuous Time Block anlegen, können Sie auswählen, ob der Eintrag als Blockdiagramm, in ESDL- oder in C-Code realisiert werden soll. Diese Eintragstypen sind über die Menüfunktionen **<item type>** bzw. über die Schaltflächen **Insert Module - <item type>** und **Insert Class - <item type>** verfügbar. Der Standard-Eintragstyp für die Schaltflächen ist einstellbar.

### Standard-Eintragstyp für Klassen/Module wählen:



- Klicken Sie auf die Pfeil-Schaltfläche neben **Insert Module - <item type>** oder **Insert Class - <item type>**.

Eine Auswahlliste wird aufgeklappt. Der aktuelle Standardtyp ist mit einem Häkchen gekennzeichnet.

- Wählen Sie den Standardtyp für Module bzw. Klassen aus.

Wenn Sie mit den Schaltflächen Komponenten erzeugen, haben diese den Standard-Eintragstyp.

### Erstellen einer Enumeration:

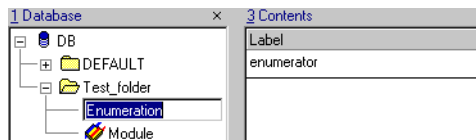
- Wählen Sie in der Liste „1 Database“ einen Ordner, der die Enumeration enthalten soll.
- Wählen Sie **Insert → Enumeration**

oder



- klicken Sie auf die Schaltfläche **Insert Enumeration**.

Die Enumeration wird mit einem Standardnamen erzeugt und im Feld „3 Contents“ wird der erste Enumerator angezeigt.



- Klicken Sie irgendwo in das Feld „3 Contents“, um es in den Fokus zu bringen.

Die auf Seite 28 beschriebenen Änderungen der Menüfunktionen werden wirksam.



- Wählen Sie **Component** → **Enumeration** → **Add**

oder

- drücken Sie auf <INSERT>

oder

- klicken Sie mit der rechten Maustaste in das Feld „3 Contents“ und wählen Sie **Add** → **<submenu>**, um einen Enumerator hinzuzufügen.

Die Funktionen in **Add** → **<submenu>** fügen den neuen Enumerator an folgenden Stellen ein:

**as first** an erster Stelle

**as last** an letzter Stelle

**before selection** vor dem markierten Enumerator

**after selection** nach dem markierten Enumerator

- Wählen Sie **Edit** → **Rename**

oder

- drücken Sie auf <F2>, um einen ausgewählten Enumerator umzubenennen.

- Wählen Sie **Edit** → **Delete**

oder

- drücken Sie auf <DEL>, um einen ausgewählten Enumerator zu löschen.

- Wählen Sie **Component** → **Enumeration** → **Shift Up**, um einen ausgewählten Enumerator in der Liste nach oben zu verschieben.

- Wählen Sie **Component** → **Enumeration** → **Shift Down**, um einen ausgewählten Enumerator nach unten zu verschieben.

Alle Menüfunktionen sind auch als Kontextmenü im Feld „3 Contents“ verfügbar.

### **Bearbeiten eines Datenbankeintrags:**

---

- Wählen Sie in der Liste „1 Database“ den zu bearbeitenden Eintrag.

- Doppelklicken Sie auf den Eintrag  
oder
- wählen Sie **Component** → **Edit Item**  
oder
- drücken Sie die <RETURN>-Taste.  
Der Editor für den gewählten Eintrag wird geöffnet.

### **Umbenennen eines Ordners oder Datenbankeintrags:**

---

- Wählen Sie aus der Liste „1 Database“ den Ordner oder Datenbankeintrag aus, den Sie umbenennen wollen.
- Wählen Sie **Edit** → **Rename**  
oder
- drücken Sie <F2>.  
Der Ordnername wird hervorgehoben.
- Bearbeiten Sie den Namen und drücken Sie <ENTER>.

### **Löschen eines Ordners oder Datenbankeintrags:**

---

#### **Hinweis**

*Ordner und Komponenten werden direkt von der Datenbank gelöscht und können nicht wieder hergestellt werden. Wenn Sie einen Ordner löschen, wird der gesamte Inhalt ebenfalls gelöscht. Benutzen Sie das Kommando „Löschen“ daher immer mit Bedacht.*

- Wählen Sie in der Liste „1 Database“ den Ordner oder Datenbankeintrag aus, den Sie löschen wollen.
- Wählen Sie **Edit** → **Delete**  
oder
- klicken Sie auf die Schaltfläche **Delete**  
oder



- drücken Sie die <DEL>-Taste.  
Wenn eine Komponente gelöscht werden soll, oder wenn ein zu löschender Ordner andere Ordner oder Einträge enthält, öffnet sich ein Bestätigungsfenster.  
Leere Ordner werden ohne Bestätigung entfernt.
- Bestätigen Sie den Löschvorgang mit **OK**.

Sie können Datenbankeinträge oder ganze Ordner und ihre Inhalte kopieren und verschieben. Wenn im Zielverzeichnis bereits ein Objekt mit identischem Namen wie das Original existiert, erhält das eingefügte Objekt den Namen des Originals mit der Erweiterung „\_1“.

### Hinweis

Hauptordner können mit **Cut/Copy/Paste** *nur* auf Hauptordnerebene kopiert und verschoben werden.

Unterordner können mit **Cut/Copy/Paste** *nicht* auf Hauptordnerebene angehoben werden.

### Kopieren eines Datenbankeintrags oder Ordners:

- Wählen Sie den zu kopierenden Eintrag oder Ordner.
- Wählen Sie **Edit** → **Copy**

oder



- klicken Sie auf die Schaltfläche **Copy**

oder

- drücken Sie <CTRL> + <C>.

Der Datenbankeintrag oder Ordner wird in die Zwischenablage kopiert.

### Ausschneiden eines Datenbankeintrags oder Ordners:

- Wählen Sie den zu kopierenden Eintrag oder Ordner.
- Wählen Sie **Edit** → **Cut**

oder

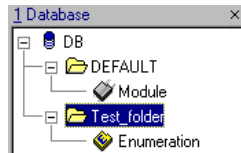


- klicken Sie auf die Schaltfläche **Cut**

oder

- drücken Sie <CTRL> + <X>.

Der Datenbankeintrag oder Ordner wird in die Zwischenablage verschoben. Das Symbol im Komponentenmanager erscheint nun schwarzweiß.



### Hinweis

Wenn Sie die Datenbank schließen, solange der Eintrag oder Ordner noch in der Zwischenablage ist, wird die Zwischenablage geleert, das Objekt aber **nicht** gelöscht.

### Einfügen eines Datenbankeintrags oder Ordners:

- Wählen Sie im Komponentenmanager den Zielordner, der den in der Zwischenablage befindlichen Datenbankeintrag oder Ordner aufnehmen soll.

- Wählen Sie **Edit** → **Paste**

oder



- klicken Sie auf die Schaltfläche **Paste**

oder

- drücken Sie <CTRL> + <V>.

Der Datenbankeintrag oder Ordner wird in den gewählten Zielordner eingefügt. Wenn er ausgeschnitten wurde (s. Seite 83), wird der entsprechende Eintrag aus dem Quellordner gelöscht.

Beim Verschieben von Einträgen oder Ordnern innerhalb der Datenbank bleiben deren Namen im Zielordner normalerweise unverändert. Nur wenn der Zielordner bereits einen Eintrag oder Ordner mit gleichem Namen enthält, wird der neue Eintrag automatisch umbenannt, um einen Namenskonflikt zu vermeiden.

Wenn Sie eine bestehende Komponente vom Blockdiagramm in C-Code oder ESDL-Code (oder umgekehrt) umwandeln wollen, können Sie die Struktur der Originalkomponente kopieren, so dass sie nicht noch einmal spezifiziert werden muss. Beim Kopieren der Struktur einer Komponente wird die gesamte Schnittstelle automatisch in der Zielkomponente angelegt.

### **Kopieren der Struktur einer Komponente:**

---

- Wählen Sie im Komponentenmanager die zu kopierende Komponente.
- Wählen Sie **Component** → **Reproduce As** → **<item type>**, um den zu erstellenden Komponententyp festzulegen.  
Die neue Komponente wird im gleichen Ordner wie das Original angelegt. Sie erhält den Namen des Originals, erweitert um „\_1“.  
Die neue Komponente enthält die gleiche Schnittstelle wie das Original, aber für die Komponente wird keine Funktionalität spezifiziert.
- Geben Sie der neuen Komponente gegebenenfalls einen anderen Namen, um Verwechslungen zu vermeiden.
- Öffnen Sie mit einem Doppelklick auf den neuen Eintrag den Komponenteneditor und spezifizieren Sie die gewünschte Funktionalität.

### **Speichern der aktuellen Datenbank:**

---

Wenn Sie Änderungen an der ASCET-Datenbank vornehmen, werden diese im Zwischenspeicher (RAM) abgelegt. Um die Änderungen dauerhaft vorzunehmen, müssen sie in der Datenbank auf der Festplatte abgespeichert werden.

- Wählen Sie **File** → **Save Database**

oder



- klicken Sie auf die Schaltfläche **Save**

oder

- drücken Sie **<CTRL> + <S>**.

Der Inhalt des Zwischenspeichers wird auf der Festplatte gespeichert.

Informationen zum automatischen Speichern der Datenbank in regelmäßigen Abständen finden Sie im Abschnitt „Grundlegende Optionen“ auf Seite 42.

### 2.3.2 Arbeiten in den Ansichtskonzepten

---

Das Arbeiten in den Ansichtskonzepten erfolgt wahlweise über Menüfunktionen oder über Kontextmenüs. Eine Beschreibung der Menüfunktionen finden Sie im Abschnitt „Die Menüfunktionen“ auf Seite 19.

#### *Bearbeiten von Komponenten und Projekten im Komponentenmanager*

---

In ASCET ist es möglich, im Komponentenmanager die Eigenschaften von Komponenten und Projekten zu ändern. Durch das Auswählen des gewünschten Elements im Feld „3 Contents“ kann entsprechend der gewählten Ansicht die Konfiguration, der Datensatz, die Implementierung oder das Layout bearbeitet werden.

Ausnahmen bilden hier Methoden-Argumente und -Rückgabewerte. Diese Elemente können nur in den Spezifikationseditoren geändert werden.

#### **Bearbeiten von Datenbankeinträgen:**

---

- Markieren Sie im Feld „1 Database“ einen Ordner.  
Im Feld „3 Contents“ wird die Ordneransicht angezeigt.
- Markieren Sie einen Eintrag im Register „Components“.
- Wählen Sie **Component** → **Edit Item**  
oder
- drücken Sie <RETURN>  
oder
- doppelklicken Sie auf den markierten Eintrag.  
Der Editor für den gewählten Eintrag wird geöffnet.
- Wählen Sie aus dem Kontextmenü **Rename**  
oder
- drücken Sie auf <F2>, um den gewählten Eintrag umzubenennen.
- Wählen Sie aus dem Kontextmenü **Select All**, um alle Einträge zu markieren.

- Wählen Sie aus dem Kontextmenü **Delete**
- oder
- drücken Sie <DEL>, um die markierten Einträge zu löschen.
  - Wählen Sie aus dem Kontextmenü **Sort by** → **<Spalte>**

oder

- klicken Sie auf den Namen einer Spalte, um die Anzeige nach der betreffenden Spalte zu sortieren.

Name	Type	D
Ref_Class	Class / Block Diagram	2
Enumeration	Enumeration	2
Module	Module / Block Diagram	2
State_Machine	State Machine	2

In aufsteigender Sortierung kommen Zahlen vor Großbuchstaben, Groß- vor Kleinbuchstaben.

Abermaliges Sortieren kehrt die Reihenfolge um.

### **Bearbeiten von Elementen:**

---

- Markieren Sie im Feld „1 Database“ eine Komponente oder ein Projekt.
- Klicken Sie im Feld „3 Contents“ auf das Register „Elements“.
- Sie erhalten die Elementansicht.
- Klicken Sie auf den Namen einer Spalte, um die Anzeige nach der betreffenden Spalte zu sortieren.
- Markieren Sie im Register „Elements“ ein Element.

- Wählen Sie **Component** → **Edit Item**

oder

- drücken Sie <ENTER>

oder

- doppelklicken Sie auf das markierte Element.  
Der Elementeditor wird geöffnet.

Die Elementkonfiguration ist in Kapitel 4.10 „Bearbeiten von Elementen“ ausführlich beschrieben.

- Wählen Sie **Edit** → **Rename**

oder

- drücken Sie <F2>, um das markierte Element umzubenennen.
- Wählen Sie aus dem Kontextmenü **Select All**, um alle Einträge zu markieren.



- Wählen Sie **Edit** → **Delete**

*oder*

- drücken Sie <DEL>, um die markierten Elemente zu löschen.

- Wählen Sie **Edit** → **Copy**

*oder*

- drücken Sie <CTRL> + <C>, um die markierten Elemente in die Zwischenablage zu kopieren.

- Wählen Sie **Edit** → **Paste**

*oder*

- drücken Sie <CTRL> + <V>, um Elemente aus der Zwischenablage in die Komponente zu kopieren.

Ist ein gleichnamiges Element bereits vorhanden, wird an den Namen des eingefügten Element automatisch ein Zähler (n) angehängt.

### **Bearbeiten der Daten:**

---

- Markieren Sie im Feld „1 Database“ eine Komponente oder ein Projekt.

- Klicken Sie im Feld „3 Contents“ auf das Register „Data“.

Sie erhalten die Datenansicht.

- Klicken Sie auf den Namen einer Spalte, um die Anzeige nach dieser Spalte zu sortieren.

- Markieren Sie das gewünschte Element.

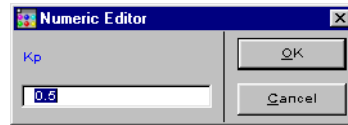
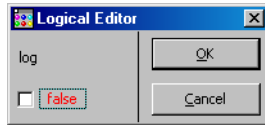
- Wählen Sie **Component** → **Edit Item**

*oder*

- drücken Sie <ENTER>

*oder*

- doppelklicken Sie auf das markierte Element.  
Je nach Art des gewählten Elements öffnet sich ein passender Dateneditor.  
Wenn Sie z.B. ein logisches oder skalares Element gewählt haben, wird das Fenster „Logical Editor“ oder „Numeric Editor“ geöffnet.



Die Bearbeitung von Daten und Datensätzen ist in Kapitel 4.11 ausführlich beschrieben.

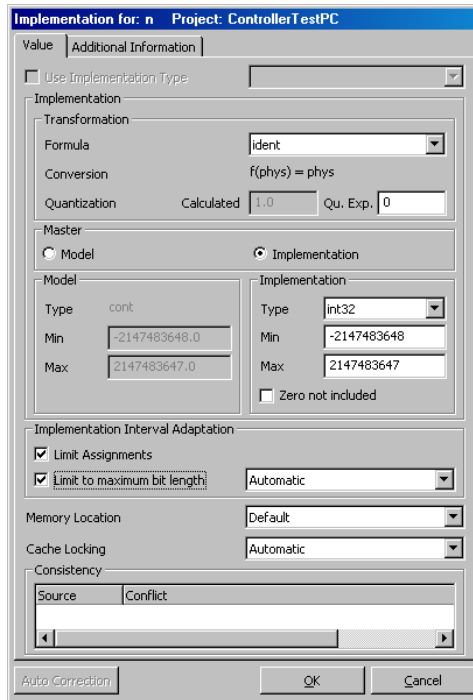
- Wählen Sie **Edit** → **Rename**  
*oder*
- drücken Sie <F2>, um das markierte Element umzubenennen.
- Wählen Sie **Edit** → **Copy**  
*oder*
- drücken Sie <Ctrl> + <C>, um die aktuellen Daten des markierten Elements in die Zwischenablage zu kopieren.
- Wählen Sie aus dem Kontextmenü **Select All**, um alle Einträge zu markieren.
- Wählen Sie **Edit** → **Paste**  
*oder*
- drücken Sie <Ctrl> + <V>, um die Daten aus der Zwischenablage in die markierten Elemente zu kopieren.
- Wählen Sie **Edit** → **Delete**  
*oder*
- drücken Sie <Del>, um die markierten Elemente zu löschen.

#### **Bearbeiten der Implementierung:**

---

- Markieren Sie im Feld „1 Database“ eine Komponente oder ein Projekt.

- Klicken Sie im Feld „3 Contents“ auf das Register „Implementation“ (Implementierungsansicht).
- Klicken Sie auf den Namen einer Spalte, um die Anzeige nach dieser Spalte zu sortieren.
- Markieren Sie das gewünschte Element.  
Dabei kann es sich um ein Basiselement oder eine enthaltene Komponente handeln.
- Wählen Sie **Component** → **Edit Item**  
oder  
• drücken Sie <ENTER>  
oder  
• doppelklicken Sie auf das gewählte Element.  
Der entsprechende Implementierungseditor wird geöffnet (die Abbildung zeigt den Implementierungseditor für Basiselemente).



Die Bearbeitung von Implementierungen ist in Kapitel 4.12 ausführlich beschrieben.

- Wählen Sie **Edit** → **Rename**  
*oder*
- drücken Sie <F2>, um das markierte Element umzubenennen.
- Wählen Sie **Edit** → **Copy**  
*oder*
- drücken Sie <CTRL> + <C>, um die aktuelle Implementierung des markierten Elements in die Zwischenablage zu kopieren.
- Wählen Sie aus dem Kontextmenü **Select All**, um alle Einträge zu markieren.
- Wählen Sie **Edit** → **Paste**  
*oder*
- drücken Sie <Ctrl> + <v>, um die Implementierung aus der Zwischenablage in die markierten Elemente zu kopieren.

#### Hinweis

*Sie können Implementierungen nur zwischen Elementen des gleichen Typs kopieren. Es ist **nicht** möglich, die Implementierung eines Elementtyps (z.B. einer Kennlinie) in ein Element eines anderen Typs (z.B. Skalar, Matrix, ...) zu kopieren.*

- Wählen Sie **Data** → **Delete**  
*oder*
- drücken Sie <Del>, um die markierten Elemente zu löschen.

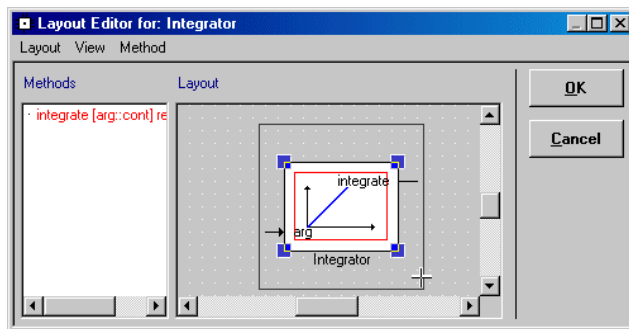
#### Bearbeiten des Layouts:

- Markieren Sie im Feld „1 Database“ die gewünschte Komponente.

#### Hinweis

*Für Projekte gibt es das Register „Layout“ nicht.*

- Klicken Sie im Feld „3 Contents“ auf das Register „Layout“.  
Sie erhalten die Layoutansicht.
- Wählen Sie **Component** → **Edit Layout**  
oder
- drücken Sie <ENTER>  
oder
- doppelklicken Sie auf das angezeigte Layout  
Der Layout-Editor wird geöffnet.



Die Bearbeitung des Layouts ist in Kapitel 4.13 ausführlich beschrieben.

### Auswahl eines weiteren Daten- oder Implementierungssatzes

Wurden weitere Daten- oder Implementierungssätze für eine Komponente angelegt, können Sie diese über ein Kombikästchen auswählen. Dieses Kombikästchen ist nur in der Daten- und Implementierungsansicht verfügbar.

### **Auswählen weiterer Daten- oder Implementierungssätze:**



- Klicken Sie im Register „Data“ oder „Implementation“ auf das Kombikästchen.
- Wählen Sie einen Daten- oder Implementierungssatz.  
Die Werte der einzelnen Elemente werden entsprechend geändert.

### 2.3.3 Exportieren von Ordnern und Datenbankeinträgen

---

In ASCET können Ordner und Einträge zwischen Datenbanken ausgetauscht werden. Sie können ganze Ordner, individuelle Einträge oder eine beliebige Auswahl von Einträgen und Ordnern exportieren. Es wird unterschieden zwischen Exportieren von Einträgen/Ordnern (*einfacher Export*) und von Einträgen/Ordnern zusammen mit allen Referenzen (*rekursiver Export*).

#### **Hinweis**

*Beim Export wird der Inhalt des Feldes „2 Comment“ (s. Seite 16) nur für Hauptordner und Datenbankeinträge mit exportiert. Anmerkungen zu Unterordnern werden nicht exportiert.*

#### *Binärer Export*

---

Das Exportformat \*.exp ist ein sehr platzsparendes Binärformat, das schnell erzeugt und gelesen werden kann. Jedoch kann das Erzeugen von solchen Exportdateien eine speicherintensive Aufgabe sein. Beim Exportieren großer Ordner können die Inhalte über mehrere Dateien verteilt werden, um den Export/Import-Vorgang zu beschleunigen. Sie können den Inhalt großer Ordner auf mehrere Dateien verteilen, indem Sie automatisch eine Datei pro exportiertem Eintrag anlegen.

Wenn im Fenster „Options“ die Option **Write XML file on Binary Export** oder **Write ASCII file on Binary Export** aktiviert wurde (siehe „Exportoptionen“ auf Seite 63), werden außer der Exportdatei Beschreibungsdateien im entsprechenden Format erzeugt. Diese können mit einem Internet-Browser bzw. einem Texteditor angeschaut werden.

#### **Hinweis**

*Für die Erzeugung von XML-Beschreibungsdateien ist derzeit der Microsoft Internet Explorer V5.5 (mit MSXML Parser V3.0) oder V6 erforderlich.*

#### *AMD-Export*

---

Ab der Version ASCET V5.2 gibt es das XML-basierte Exportformat \*.amd. Wenn Sie dieses Exportformat wählen, wird für jede exportierte Komponente eine Reihe von \*.amd-Dateien angelegt, die die komplette Modellbeschreibung enthalten. Die Namen der Dateien sind wie folgt aufgebaut:

*<Komponentenname>.<Informationstyp>.amd*

Tab. 2-3 listet die Informationstypen und Inhalte.

Informationstyp	Inhalt
main	Namen und Eigenschaften der Elemente der Komponente/des Projekts (s. Seite 481)
data	Daten der Elemente der Komponente/des Projekts (s. Seite 493)
experiments	in der Komponente/dem Projekt definierte Experimentierumgebungen (s. Seite 603)
implementation	Implementierungen der Elemente der Komponente/des Projekts (s. Seite 513)
specification	Einzelheiten zur Spezifikation (z.B. Schnittstelleninformationen, Aufbau des Blockdiagramms, Code einer ESDL- oder C-Code-Komponente)
project <sup>a</sup>	projektspezifische Daten wie Betriebssystem-Konfiguration (s. Seite 419), Projekteinstellungen (s. Seite 438)
project.formulas <sup>a</sup>	im Projekt definierte Formeln (s. Seite 456)
project.implementationTypes <sup>a</sup>	im Projekt definierte Implementierungstypen (s. Seite 463)

a. nur für Projekte

### Tab. 2-3 Typen der AMD-Exportdateien

Jede AMD-Datei enthält eine Signatur, anhand derer beim Import überprüft wird, ob die Datei zwischen Export und Import geändert wurde.

Auf Wunsch können die AMD-Dateien gleich beim Export in einer Zip-Datei zusammengefasst und komprimiert werden. Diese Zip-Datei hat die Erweiterung \*.axl.

#### Sonstige Exportformate

Außer den bisher genannten Exportformaten steht Ihnen noch ein XML-Format (\*.xml) zur Verfügung, das der optionalen XML-Beschreibungsdatei beim binären Export entspricht. Dieses XML-Format ist grundlegend verschieden vom AMD-Format; es kann auch nicht wieder in ASCET importiert werden.

Für den Export von ASAM-MCD-2MC-Projekten steht das Exportformat \*.a2l zur Verfügung.

#### Hinweis

Wenn Sie das \*.a2l-Format für etwas anderes als den Export eines ASAM-MCD-2MC-Projekts verwenden, produzieren Sie einen Fehler.

## Export durchführen

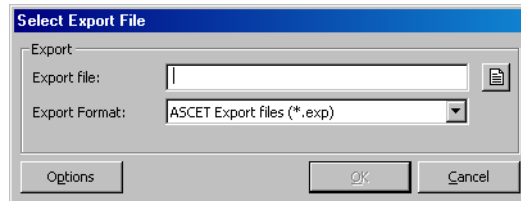
Bevor Sie Ordner oder Einträge exportieren, stellen Sie den jeweiligen Exportmodus (einfach oder rekursiv) und die Dateiverteilung in den Exportoptionen ein (siehe „Exportoptionen“ auf Seite 63).

### Exportieren von Ordnern oder Einträgen:

- Wählen Sie in der Liste „1 Database“ des Komponentenmanagers die Ordner oder Einträge, die Sie exportieren wollen.
- Wählen Sie **File** → **Export**  
oder
- wählen Sie **Export** aus dem Kontextmenü  
oder
- klicken Sie auf die Schaltfläche **Export**  
oder
- drücken Sie <CTRL> + <E>.



Das Fenster „Select Export File“ öffnet sich.




- Wählen Sie im Kombikästchen „Export Format“ ein Exportformat.

ASAP2 files (*.a2l)	nur ASAM-MCD-2MC-Projekte
ASCET Model Data files (*.amd)	AMD-Export (s. Seite 94)
ASCET compressed Model Data files (*.axl)	komprimierter AMD-Export
ASCET Export files (*.exp)	binärer Export (s. Seite 94)
ASCET XML files old format (*.xml)	XML-Export, nicht importierbar



- Geben Sie im Feld „Export File“ Pfad und Name der Exportdatei direkt oder mit Hilfe der Schaltfläche  an.

### Hinweis

Wenn die Exportoption **One File for each Item** aktiviert ist, heißt das Feld „Export folder“, und die Schaltfläche sieht so aus: 

Jetzt ist die Schaltfläche **OK** verfügbar.

- Klicken Sie auf die Schaltfläche **Options**, wenn Sie die Exportoptionen anpassen wollen.

Das Fenster „Options“ öffnet sich im Knoten „Export“ (s. Abschnitt „Exportoptionen“ auf Seite 63).

- Setzen Sie die Optionen für den Export.
- Klicken Sie auf **OK**, um das Optionsfenster zu schließen.

- Klicken Sie im Fenster „Select Export File“ auf **OK**, um den Export zu starten.

Die Objekte werden im gewählten Format in die angegebene Datei im gewählten Ordner exportiert.

Beim AMD-Export wird für jedes exportierte Objekt eine Reihe von \*.amd-Dateien angelegt.

### Hinweis

Wenn Sie Datenbankeinträge auf der Basis „Eine Datei pro Element“ exportieren, oder beim AMD-Export inklusive referenzierter Elemente sollten Sie die Einträge in ein leeres Verzeichnis exportieren, um die Übersicht zu behalten. ASCET exportiert automatisch in das unter „Default Export Path“ angegebene Verzeichnis.

## 2.3.4 Importieren von Ordnern und Datenbankeinträgen

Jedes Objekt in einer ASCET-Datenbank hat eine Identifikation und ist der Datenbank nur über diese Kennung bekannt. Datenbankeinträge referenzieren sich nur anhand dieser Zeichen und nicht an den Namen, die im Komponentenmanager angezeigt werden.

Sie können aus einer oder mehreren Exportdateien der folgenden Exportformate in die Datenbank importieren.

binäre Exportdatei (s. Seite 94)	ASCET Export files (*.exp;*.prj)
AMD-Exportdatei (s. Seite 94)	ASCET Model Data files (*.main.amd;*.main.xml)
komprimierte AMD-Exportdatei	ASCET compressed Model Data files (*.axl;*.zip)
ASAM-MCD-2MC-Projekt	ASAP2 files (*.a21)

Das XML-Exportformat (s. Seite 95) kann *nicht* importiert werden.

Die erforderlichen Ordner und Einträge werden automatisch angelegt. Sie müssen allerdings ggf. das Überschreiben vorhandener Objekte bestätigen.

In den Exportdateien sind neben den Komponenten auch deren Pfadangaben aus der exportierenden Datenbank enthalten. Wenn ein zu importierender Eintrag in einem Ordner liegt, der auch in der Zieldatenbank existiert, wird der Eintrag in diesen Ordner geschrieben. Wenn ein Eintrag sich in einem Ordner befindet, den es in der Zieldatenbank nicht gibt, wird der Ordner automatisch angelegt. Wenn nötig, werden mehrere Ordnerstufen erstellt. Dabei wird die in der Exportdatei gespeicherte Ordnerhierarchie hergestellt.

Wird ein Eintrag importiert, der in der Zieldatenbank schon existiert, wird der vorhandene Eintrag überschrieben. Da Datenbankeinträge über ihre Objekt-IDs und nicht über ihre Namen identifiziert werden, schützt sie ein Umbenennen nicht vor einem Überschreiben.

Beim AMD-Import sorgt die Import-Option **Remap OID** dafür, dass ein vorhandenes Objekt nicht überschrieben, sondern eine Kopie mit einer neuen Objekt-ID angelegt wird.

Wenn ein importierter Eintrag einen bestehenden überschreibt, werden der Pfad in der Zieldatenbank und das Verhalten bestehender Implementierungen in den Importoptionen festgelegt. Wenn Sie die Option **Keep Folder Path in Database** deaktivieren, wird der importierte Eintrag unter dem Pfad aus der Exportdatei abgelegt, auch wenn er in der Zieldatenbank in einem anderen Ordner liegt. Um den Pfad der Zieldatenbank beizubehalten, muss die Option **Keep Folder Path in Database** aktiviert werden.

Es ist möglich, die Daten in der Datenbank vor dem Überschreiben beim Import zu schützen.

### **Verbieten von Überschreiben von Daten in der Datenbank:**

---

- Wählen Sie im Komponentenmanager die zu schützenden Einträge und Ordner.

- Wählen Sie **Component** → **Disallow Import**.

Die ausgewählten Einträge und Ordner werden gegen Überschreiben beim Importieren geschützt. In der Liste „1 Database“ werden die geschützten Objekte mit `<Disallow Import>` markiert.

Wenn ein Ordner geschützt ist, kann der Ordner weder überschrieben werden noch können neue Einträge in diesen Ordner importiert werden.

### *Import durchführen*

---

Bevor Sie Ordner oder Einträge importieren, stellen Sie die Import-Optionen in den Benutzereinstellungen ein (siehe „Importoptionen“ auf Seite 65).

### **Importieren aus Exportdateien:**

---

- Öffnen Sie im Komponentenmanager die Datenbank, in die Sie importieren wollen.
- Für AMD-Import oder den Import von ASAM-MCD-2M-Dateien wählen Sie einen Ordner.

- Wählen Sie **File** → **Import**

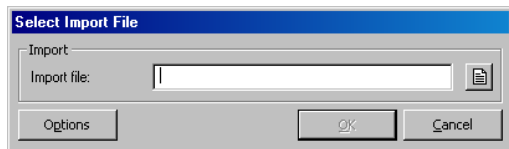
*oder*


- klicken Sie auf die Schaltfläche **Import**

*oder*

- drücken Sie `<CTRL> + <M>`.

Das Fenster „Select Import File“ öffnet sich.



- Geben Sie im Feld „Import File“ Pfad und Name der Datei, die Sie importieren wollen, direkt oder mit Hilfe der Schaltfläche  an.

### Hinweis

Sie können auch mehrere Dateien eingeben bzw. auswählen.

Das Format einer Datei muss nicht extra angegeben werden, es ist durch die Endung bestimmt.

Jetzt ist die Schaltfläche **OK** verfügbar.

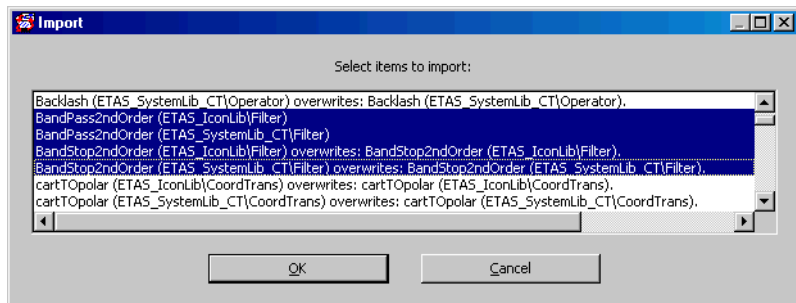
- Klicken Sie auf die Schaltfläche **Options**, wenn Sie die Importoptionen anpassen wollen.

Das Fenster „Options“ öffnet sich im Knoten „Import“ (s. Abschnitt „Importoptionen“ auf Seite 65).

- Setzen Sie die Optionen für den Import.
- Klicken Sie auf **OK**, um das Optionsfenster zu schließen.

- Klicken Sie im Fenster „Select Import File“ auf **OK**, um den Import zu starten.

Das Fenster „Import“ oder „Items available for import“ listet den Inhalt der Datei(en) auf. Ist eins der Objekte bereits in der Datenbank vorhanden (gleiche Objekt-ID), wird im Fenster angezeigt, welches Objekt überschrieben wird.



- Wählen Sie alle zu importierenden Einträge und klicken Sie auf **OK**.

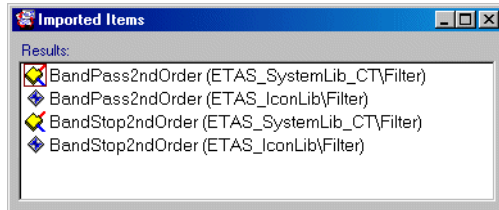
Wenn Sie bereits vorhandene Objekte importieren wollen, werden Sie aufgefordert, das Überschreiben der existierenden Datenbankobjekte zu bestätigen.

### Hinweis

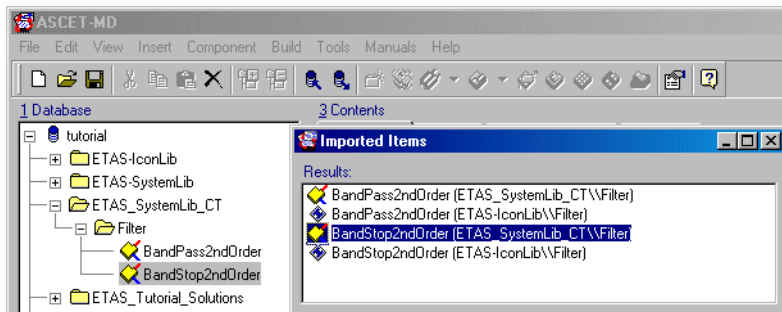
Wenn für den AMD-Import die Import-Option **Remap OID** aktiviert ist, wird ein vorhandenes Objekt nicht überschrieben, sondern eine Kopie mit einer neuen Objekt-ID angelegt.

- Bestätigen Sie die Meldung mit **OK**.

Die im Fenster „Import“ ausgewählten Einträge werden importiert. Abschließend werden die importierten Elemente im Fenster „Imported Items“ angezeigt.



- Wenn Sie im Fenster „Imported Items“ auf einen Eintrag klicken, wird er im Komponentenmanager ausgewählt.



## Importieren eines Verzeichnisinhalts

---

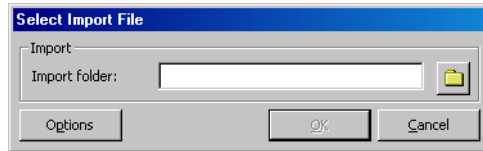
Um auf einfachem Weg *alle* Exportdateien eines ganzen Verzeichnisses zu importieren, gibt es die Menüfunktion **File** → **Import Directory**.


### Importieren eines Verzeichnisinhalts:

---

- Wählen Sie im Komponentenmanager **File** → **Import directory**.

Das Fenster „Select Import File“ öffnet sich.



- Geben Sie im Feld „Import folder“ den Pfad des Verzeichnisses, das Sie importieren wollen, direkt oder mit Hilfe der Schaltfläche  an. Jetzt ist die Schaltfläche **OK** verfügbar.

- Klicken Sie auf die Schaltfläche **Options**, wenn Sie die Importoptionen anpassen wollen.

Das Fenster „Options“ öffnet sich im Knoten „Import“ (s. Abschnitt „Importoptionen“ auf Seite 65).

- Setzen Sie die Optionen für den Import.
- Klicken Sie auf **OK**, um das Optionsfenster zu schließen.

- Klicken Sie im Fenster „Select Import File“ auf **OK**, um den Import zu starten.

Für die erste Importdatei öffnet sich ein Fenster „Import“ oder „Items available for import“, das den Inhalt der Datei auflistet. Ist eins der Objekte bereits in der Datenbank vorhanden (gleiche Objekt-ID), wird im Fenster angezeigt, welches Objekt überschrieben wird.

- Wählen Sie alle zu importierenden Einträge und klicken Sie auf **OK**.

Wenn Sie bereits vorhandene Objekte importieren wollen, werden Sie aufgefordert, das Überschreiben der existierenden Datenbankobjekte zu bestätigen.

- Bestätigen Sie die Meldung mit **OK**.

Die ausgewählten Einträge werden importiert. Anschließend öffnet sich ein Fenster „Import“ oder „Items available for import“ für die nächste Importdatei.

Nach Abschluss aller Importe werden die importierten Elemente im Fenster „Imported Items“ angezeigt.

### *Besonderheiten beim AMD-Import*

---

Beim Import aus AMD-Dateien werden verschiedene Prüfungen durchgeführt.

**Integritätsprüfung:** Anhand der Signaturen in den Dateien wird geprüft, ob die Dateien durch Dritte geändert wurden.

**Konsistenzprüfung:** Das System sucht nach folgenden Fehlern:

- ungültige Datei

Wenn eine der AMD-Dateien ungültig ist, d.h. unlesbar, nicht XML-konform oder nicht konform zum zugehörigen Schema ist, geschieht folgendes.

- Die Komponente wird nicht importiert.
- Das Fenster „Import Problems“ zeigt Datei und Problem an.
- Eine automatische Problemlösung gibt es nicht.

- fehlende Datei

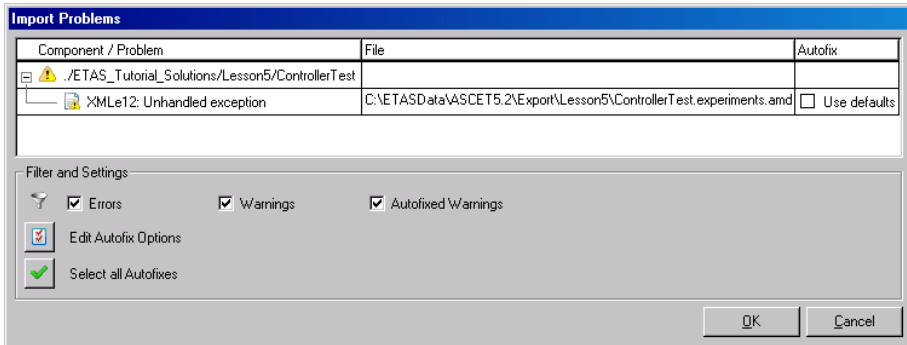
Eine ASCET-Komponente wird durch eine Reihe von Dateien spezifiziert (siehe „AMD-Export“). Wenn eine oder mehrere dieser Dateien fehlen, geschieht folgendes.

- Die Komponente wird nicht importiert.
- Das Fenster „Import Problems“ nennt die fehlenden Dateien.
- Der Anwender hat die Möglichkeit, für die fehlenden Dateien Standard-Dateien zu erzeugen.



- fehlende Komponenten beim Import  
 Wenn eine referenzierte Komponente nicht vorhanden oder nicht verfügbar ist, nennt das Fenster „Import Problems“ die fehlenden Dateien.
- fehlende Informationen in einer Datei  
 Wenn in einer AMD-Datei Informationen fehlen (z.B. wurde ein Element definiert, dessen Daten fehlen), die Datei selbst aber gültig (d.h. lesbar, konform zu XML und zum zugehörigen Schema) ist, geschieht folgendes.
  - Die Komponente wird nicht importiert.
  - Das Fenster „Import Problems“ nennt die fehlenden Informationen.
  - Der Anwender hat die Möglichkeit, für die fehlenden Informationen Standardwerte zu erzeugen.
- falsche Informationen in einer Datei  
 Wenn eine AMD-Datei falsche Informationen (z.B. Daten, die nicht zu Informationen in anderen AMD-Dateien der Komponente passen) enthält, geschieht folgendes.
  - Die Komponente wird nicht importiert.
  - Das Fenster „Import Problems“ nennt die falschen Informationen.
  - Eine automatische Problemlösung gibt es nicht.
- obsoletere Informationen in einer Datei  
 Wenn eine AMD-Datei mehr Informationen enthält als gebraucht werden (z.B. eine Datenkonfiguration für ein Element, das nicht existiert), geschieht folgendes.
  - Die Komponente wird nicht importiert.
  - Das Fenster „Import Problems“ nennt die obsoleten Informationen.
  - Der Anwender hat die Möglichkeit, die obsoletere Information zu ignorieren.



Das Fenster „Import Problems“ enthält folgende Elemente.



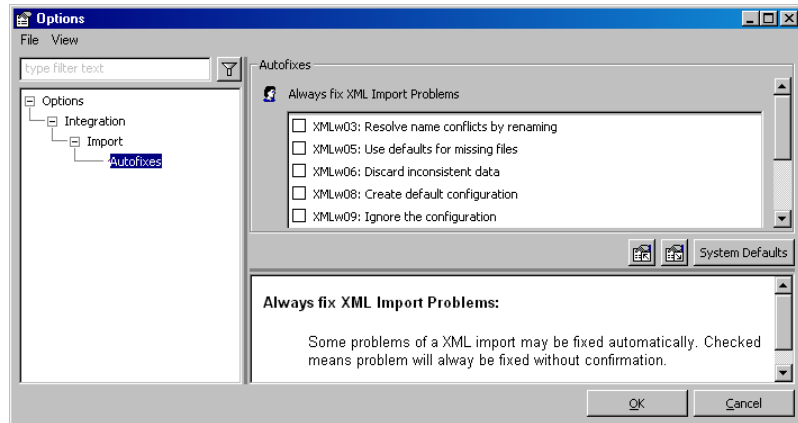
- Spalte „Component / Problem“  
Betroffene Komponente und Art des Problems, z.B. unhandled exception, Invalid AMD file (missing), Data for Element <name> missing oder Obsolete data for undefined element <name>.  
Probleme können vom Typ Warnung (Symbole und ) oder Fehler (Symbole und ) sein.
- Spalte „File“  
Pfad und Name der betroffenen Datei
- Spalte „Autofix“  
Automatische Problembehebung; die möglichen Lösungen müssen aktiviert werden, damit die entsprechende Aktion ausgeführt wird.  
Für Probleme ohne automatische Lösung ist die Spalte leer.
- Kontextmenü in der Tabelle
  - **Expand all Nodes** (<ALT>+<\*>)  
Klappt alle Knoten in der Spalte „Component / Problem“ auf.
  - **Select All Autofixes** (<CTRL>+<A>)  
Aktiviert alle automatischen Lösungen in der Spalte „Autofix“.
  - **Select Autofixes of this Type** (<ALT>+<A>)  
Nur verfügbar, wenn in in der Spalte „Autofix“ ein Eintrag gewählt ist. Aktiviert alle automatischen Lösungen des gewählten Typs.

- **Always Autofix Problems of this Type**  
Nur verfügbar, wenn in in der Spalte „Autofix“ ein Eintrag gewählt ist. Sorgt dafür, dass die Lösungen des gewählten Typs in Zukunft automatisch ausgeführt werden, ohne dass Sie im Fenster „Import Problems“ Optionen aktivieren müssen.
- **Save Issue List as XML** (<CTRL>+<S>)  
Speichert den Inhalt des Fenster in eine XML-Datei. Für einige Fehler können Sie so noch weitere Informationen bekommen.  
Diese XML-Datei ist hilfreich bei großen Importen, wenn Sie die dabei aufgetretenen Probleme automatisch analysieren lassen wollen.
- Filteroptionen
  - **Errors**  
Zeigt Probleme vom Typ Fehler an.
  - **Warnings**  
Zeigt Probleme vom Typ Warnungen an.
  - **Autofixed Warnings**  
Zeigt automatisch gelöste Probleme vom Typ Warnung an.
- Schaltfläche **Edit Autofix Options** ()  
Öffnet das Fenster „Options“, in dem Einstellungen zur automatischen Problembehebung vorgenommen werden können.
- Schaltfläche **Select all Autofixes** ()  
Aktiviert alle automatischen Lösungen in der Spalte „Autofix“.
- Schaltfläche **OK**  
Schließt das Fenster und führt die aktivierten automatischen Lösungen aus.
- Schaltfläche **Cancel**  
Schließt das Fenster, ohne die ausgewählten automatischen Lösungen auszuführen.

## Optionen zur automatischen Problembhebung setzen:

- Klicken Sie im Fenster „Import Problems“ auf die Schaltfläche **Options**.

Das Fenster „Options“ öffnet sich. Es enthält nur den Knoten „Import“.



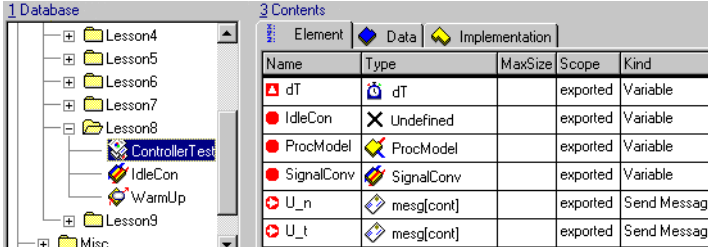
Die Liste „Always fix AMD Import Problems“ enthält Lösungen für die möglichen Probleme beim AMD-Import.

- Aktivieren Sie die Optionen für alle Probleme, die ohne Anzeige im Fenster „Import Problems“ automatisch behoben werden sollen.
- Schließen Sie das Fenster „Options“ mit **OK**.

Beim nächsten AMD-Import werden die markierten Lösungen automatisch ausgeführt, ohne dass Sie im Fenster „Import Problems“ Optionen aktivieren müssen.

## Importieren von Projekten

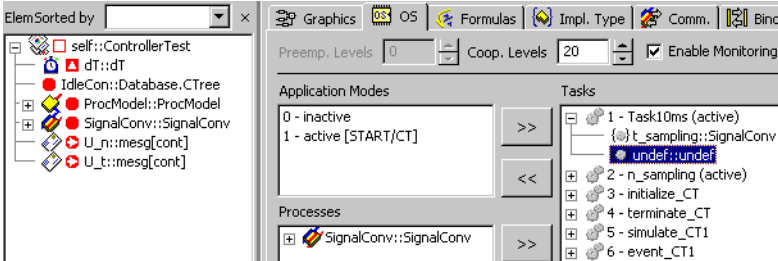
Projekte werden wie alle anderen Datenbankeinträge importiert. Wenn nach dem Import nicht alle zu einem Projekt gehörigen Module in der Datenbank vorhanden sind, werden fehlende Module in der Elementansicht des Komponentenmanagers angezeigt.



The screenshot shows the Component Manager interface. On the left, a tree view displays a project structure with folders Lesson4 through Lesson9 and Misc, and files ControllerTest, IdleCon, and WarmUp. On the right, the '3 Contents' table lists elements with their names, types, and scopes.

Name	Type	MaxSize	Scope	Kind
dT	dT		exported	Variable
IdleCon	Undefined		exported	Variable
ProcModel	ProcModel		exported	Variable
SignalConv	SignalConv		exported	Variable
U_n	msg[cont]		exported	Send Message
U_t	msg[cont]		exported	Send Message

Wird das Projekt im Projekteditor (Kapitel 4.8 „Der Projekteditor“ auf Seite 398) geöffnet, sind die Namen der fehlenden Module in der Liste „Elements“ enthalten. Im Betriebssystemeditor (Register „OS“, s. „Definieren des Scheduling im Betriebssystemeditor“ auf Seite 419) sind die fehlenden Prozesse aus dem Feld „Processes“ gelöscht, in der Taskliste jedoch sind sie als offene Referenzen enthalten (`undef::undef`).



The screenshot shows the Project Editor on the left and the OS editor on the right. The Project Editor shows a tree view with 'self::ControllerTest' and several missing modules: 'dT::dT', 'IdleCon::Database.CTree', 'ProcModel::ProcModel', 'SignalConv::SignalConv', 'U\_n::msg[cont]', and 'U\_t::msg[cont]'. The OS editor shows 'Preemp. Levels' set to 0 and 'Coop. Levels' set to 20. The 'Application Modes' section shows '0 - inactive' and '1 - active [START/CT]'. The 'Tasks' section shows a list of tasks, including '1 - Task10ms (active)', 't\_sampling::SignalConv', and 'undef::undef'. The 'Processes' section shows 'SignalConv::SignalConv'.

Sie können die fehlenden Module nachträglich importieren, auch wenn Sie das Projekt geöffnet hatten. Anschließend sind im Projekt keine offenen Referenzen mehr enthalten, auch die Prozesse werden automatisch wieder in die richtigen Tasks eingefügt.

## Importieren von Objekten aus anderen ASCET-Versionen

Exportdateien, die mit ASCET-SD vor der Version 4.0 erstellt wurden, können nicht in ASCET 5.2 importiert werden, da die aktuelle Version keine Schnittstellen zu den damaligen Datenbankformaten enthält. Wenn Sie versuchen, eine solche Datei wie auf Seite 99 beschrieben zu importieren, erhalten Sie folgende Fehlermeldung:

This file is compatible with ASCET-SD V3.0 or earlier. It cannot be imported directly. Convert it to a database using your old version of ASCET-SD, then open this database.

Umgekehrt können binäre Exportdateien aus ASCET V5.2 mit ASCET V5.0 oder 5.1 importiert werden.

#### **Importieren einer Exportdatei vor der Version ASCET-SD 4.0:**

---

Wenn Sie ASCET-SD 4.x noch haben, können Sie ganz alte Exportdateien dennoch mit ASCET 5.2 verwenden. Gehen Sie wie folgt vor:

- [Starten Sie ASCET-SD 4.x.](#)  
Diese Versionen enthalten noch Schnittstellen zu den ganz alten Datenbankformaten.
- [Importieren Sie die alte Exportdatei.](#)  
Die Datenbankobjekte werden dabei in das Format der entsprechenden Version von ASCET-SD konvertiert.
- [Exportieren Sie die importierten Objekte.](#)
- [Schließen Sie ASCET-SD 4.x.](#)
- [Starten Sie ASCET 5.2.](#)
- [Importieren Sie die eben aus ASCET-SD 4.x exportierten Datenbankobjekte wie auf Seite 99 beschrieben.](#)

### 2.3.5 [Arbeiten mit Datenbankeinträgen](#)

---

#### *Referenzen auf Einträge*

---

Eine Referenz auf einen Eintrag wird hergestellt, sobald ein Eintrag in einem anderen Eintrag benutzt oder eingebunden wird (z.B. wird ein Modul, das in einem Projekt eingebunden ist, von diesem Projekt referenziert). Beim Verschieben oder Umbenennen von Einträgen werden alle Referenzen auf den betreffenden Eintrag automatisch aktualisiert.

Der Komponentenmanager geht nicht wieder automatisch mit der Datenbank synchron, wenn Referenzen verändert wurden. Um eine Durchgängigkeit der Referenzen im Komponentenmanager zu gewährleisten, müssen sie explizit aktualisiert werden.

## Aktualisieren von Referenzen im Komponentenmanager:

---

- Wählen Sie im Komponentenmanager **View → Update**

oder

- drücken Sie <F5>.

Die Anzeige im Komponentenmanager wird mit der Datenbank abgeglichen und ggf. aktualisiert.

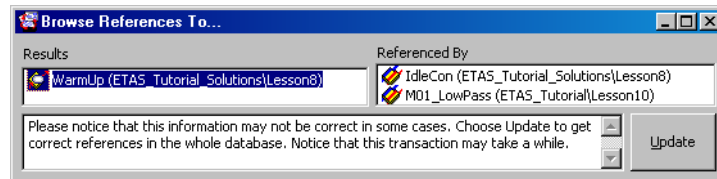
Sie können sich vor dem Löschen von Einträgen oder Ordnern eine Liste von Referenzen ansehen, um Durchgängigkeit zu sichern.

## Anzeigen der Referenzen auf einem Datenbankelement:

---

- Wählen Sie einen Datenbankeintrag.
- Wählen Sie **Component → Show References**.

Das Fenster „Browse References To“ öffnet sich.



Es werden die Namen aller Einträge angezeigt, die eine Referenz auf den gewählten Eintrag haben.

Referenzen können zyklisch sein; ein Eintrag kann einen Eintrag referenzieren, der wiederum den ersten Eintrag referenziert. Beim Löschen eines Eintrags aus der Datenbank sollten Sie immer sichergehen, dass keine referenzierende Komponente beschädigt wird. ASCET zeigt eine Warnung und eine Liste von Referenzen, wenn Sie versuchen, eine durch andere Komponenten referenzierte Komponente zu löschen.

Das Löschen von Datenbankeinträgen, die von anderen Komponenten referenziert werden, kann Ihre spezifizierten Komponenten zerstören. Dem kann vorgebeugt werden, indem die zu entfernende Komponente durch eine neue oder eine andere Komponente ersetzt wird.

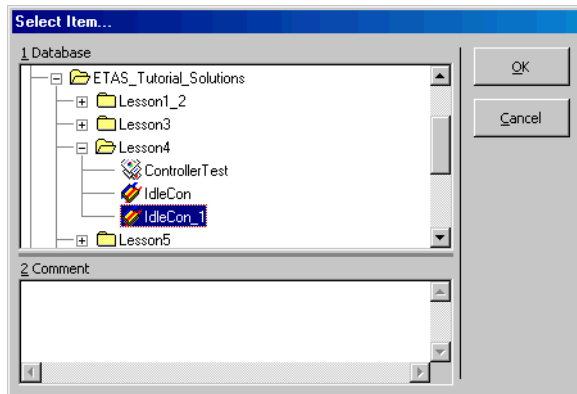
Wenn eine Komponente ersetzt wird, passen alle Datenbankeinträge, die die ersetzte Komponente referenzieren, ihre Referenzen an die ersetzende Komponente an. Die ersetzte Komponente wird nicht weiter von einem anderen Eintrag referenziert.

### Ersetzen der Referenzen auf einen Datenbankeintrag:

---

- Wählen Sie in der Liste „1 Database“ des Komponentenmanagers das zu ersetzende Element.
- Wählen Sie **Component** → **Replace References**.

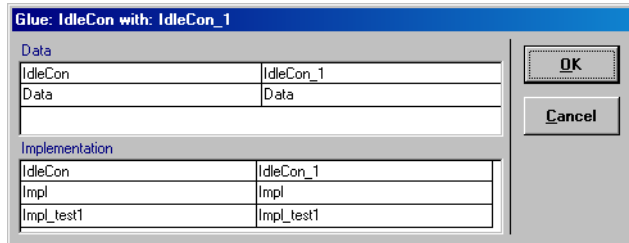
Das Fenster „Select Item“ öffnet sich.



- Wählen Sie den Eintrag, der den ersten Eintrag ersetzen soll, und klicken Sie auf **OK**.

- Bestätigen Sie das Kommando im Fenster „Confirm“ mit **OK**.

Das Fenster „Glue: <item> with: <item>“ wird geöffnet, in dem alle anfallenden Ersetzungen aufgelistet sind.



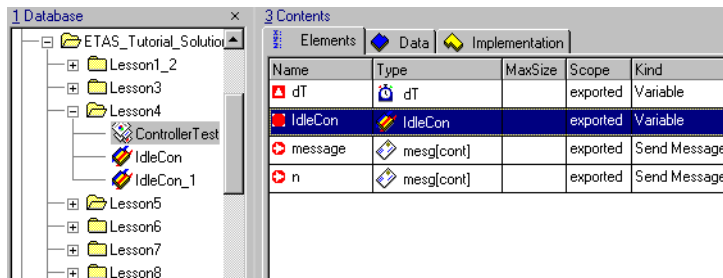
- Bestätigen Sie mit **OK**.

Die Referenzen auf den Eintrag werden in der gesamten Datenbank ersetzt.

Bei diesem Verfahren wird nur eine Referenz auf einen Eintrag durch eine Referenz auf einen anderen Eintrag ersetzt. Das Ersetzen von Referenzen betrifft die Einträge der Datenbank nicht in der Weise, dass sie hinterher nicht mehr existieren oder an anderer Stelle stehen.

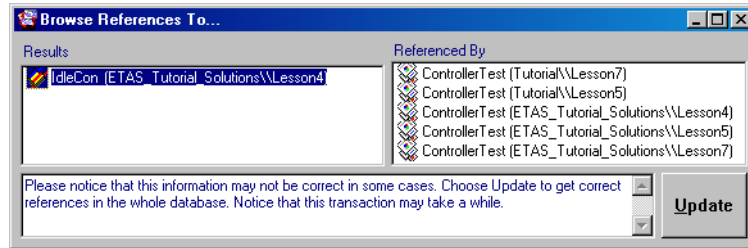
Ein Beispiel soll die Funktionsweise von **Component** → **Replace References** verdeutlichen.

Die Abbildung zeigt einen Ausschnitt aus der Elementansicht des Komponentenmanagers für das Projekt `ControllerTest` im Ordner `Lesson4` der Tutoriumsdatenbank. Dieses Projekt referenziert die Komponente `IdleCon`.



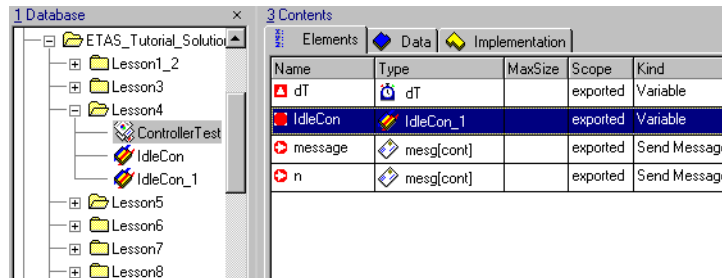


Dies kann nach Anwahl von IdleCon in der Liste „1 Database“ durch **Component** → **Show References** überprüft werden.

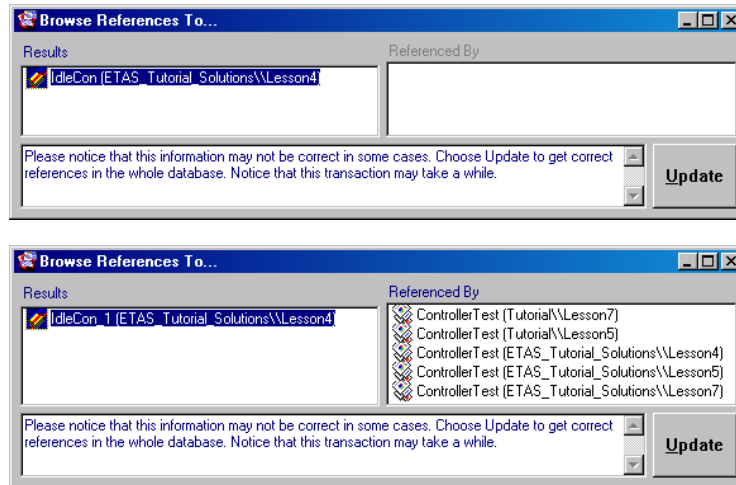


Die Komponente IdleCon wird in der ganzen Datenbank von fünf gleichnamigen Projekten (ControllerTest) referenziert.

Wie oben beschrieben wird die Referenz auf IdleCon mit **Component** → **Replace References** durch eine Referenz auf IdleCon\_1 ersetzt. Nach Ausführung des Befehls ist das Feld „1 Database“ unverändert, aber in der Elementansicht kann man erkennen, dass das Projekt ControllerTest jetzt unter dem alten Namen den IdleCon\_1 referenziert. Beide Komponenten sind aber nach wie vor in der Datenbank vorhanden.



Die Überprüfung beider Komponenten mit **Component → Show References** ergibt folgendes Resultat:



Es existiert keine Referenz auf `IdleCon` mehr, die fünf Projekte referenzieren jetzt `IdleCon_1`.

In ASCET 5.2 können Sie einen Datenbankeintrag und alle Referenzen von anderen Komponenten oder Projekten darauf auch vollständig ersetzen. Diese Möglichkeit ist von besonderer Bedeutung, wenn Sie mit einem Konfigurationsmanagement-Tool arbeiten und auseinandergehende Entwicklungsströme für ein System zusammenführen müssen.

Da jeder Datenbankeintrag nur an seinem Identitätszeichen erkannt wird, übernimmt ein ersetzender Eintrag einfach nur das Identitätszeichen des zu ersetzenden Eintrags. Hat Ihre Datenbank beispielsweise zwei Objekte A1 und A2, und soll A1 durch A2 ersetzt werden, erhält das resultierende Objekt A2 den Identifikator und die Referenzen von A1, aber es behält seine Funktionalität und seinen Namen A2.

### Ersetzen eines Datenbankeintrags:

- Wählen Sie in der Liste „1 Database“ des Komponentenmanagers den Eintrag, der den Originaleintrag ersetzen soll.
- Wählen Sie **Component → Become another Item**.

Das Fenster „Select Item“ öffnet sich.

- Wählen Sie den Eintrag, der ersetzt werden soll.
- Klicken Sie auf **OK**, um die Referenzen auf diesen Eintrag und ihn selbst zu ersetzen.
- Bestätigen Sie den Befehl im Fenster „Confirm“ mit **OK**.

Das Fenster „Glue <item>: with <item>:“ wird geöffnet, in dem alle anfallenden Ersetzungen aufgelistet sind.

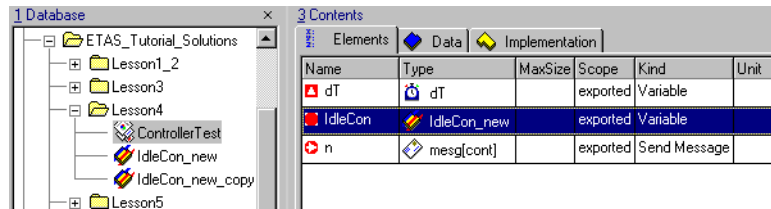
- Bestätigen Sie mit **OK**.

Die ursprüngliche Instanz des Eintrags, der den zweiten Eintrag ersetzt, behält seinen Identifikator und wird umbenannt in <item\_name>\_copy.

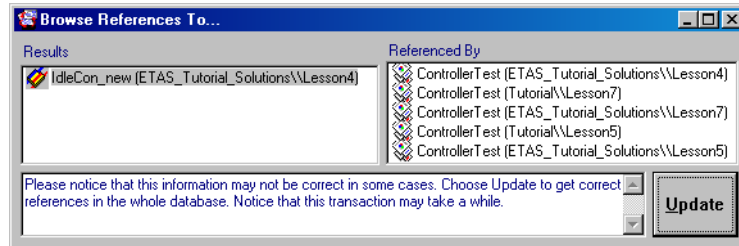
Wieder soll ein Beispiel die Arbeitsweise des Befehls verdeutlichen. Es handelt sich um dasselbe Beispiel wie eben, diesmal wird jedoch die Komponente IdleCon komplett ersetzt, nicht nur die Referenzen darauf.

Vor Ausführung des Befehls **Component → Become another Item** referenziert das Projekt ControllerTest die Komponente IdleCon. Die Komponente IdleCon\_new wird von einem Projekt namens Project referenziert.

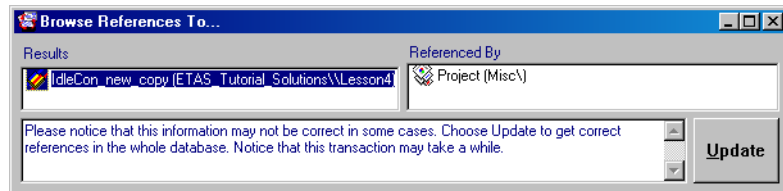
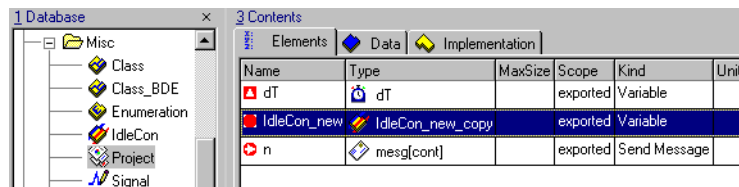
Wie oben beschrieben wird jetzt mit dem Befehl **Component → Become another Item** die Komponente IdleCon durch IdleCon\_new ersetzt. Nach Ausführung des Befehls ist die Komponente IdleCon aus dem Komponentenmanager gelöscht. Im Feld „1 Database“ finden sich jetzt die Komponenten IdleCon\_new und IdleCon\_new\_copy.



In der Elementansicht ist zu erkennen, dass ControllerTest (und die anderen gleichnamigen Projekte) jetzt unter dem alten Namen die Komponente IdleCon\_new referenziert. Diese Komponente hat jetzt den Identifikator der ersetzten Komponente IdleCon.



Die Komponente IdleCon\_new\_copy beinhaltet die ursprüngliche Instanz der ersetzenden Komponente und wird entsprechend vom Projekt Project referenziert.



### Bearbeiten

Das Layout einer Komponente lässt sich bearbeiten, ohne zuvor den Komponenteneditor aufzurufen. Die öffentliche Schnittstelle einer Komponente wird im Layout-Editor deklariert.

### Bearbeiten eines Komponenten-Layouts:

- Wählen Sie eine Komponente aus.

- Wählen Sie **Component** → **Edit Layout**.

Der Layout-Editor für die gewählte Komponente wird geöffnet. Er ist in „Bearbeiten des Layouts einer Komponente“ auf Seite 552 beschrieben.

An jeden Datenbankeintrag können zwei Sorten von Text angehängt werden: ein Kommentar und Notizen. Der Kommentar wird im Feld „2 Comment“ im Komponentenmanager eingegeben und automatisch abgelegt. Die Notizen werden in einem separaten Editorfenster eingegeben. Bei der automatischen Erzeugung einer Dokumentation werden Kommentare im Gegensatz zu Notizen nicht mit in die Dokumentation eingefügt.

#### **Bearbeiten der Notizen eines Datenbankeintrags:**

---

- Wählen Sie den Datenbankeintrag mit den zu ändernden Notizen aus.
- Wählen Sie **Component** → **Notes**.

Der Notizeneditor für den gewählten Eintrag wird geöffnet. Einzelheiten dazu siehe Kapitel 7.4 „Notizen“ auf Seite 747.

In Abhängigkeit vom Attribut „Memory“ (siehe „Elementkonfiguration“ auf Seite 482) werden Variablen und Parameter von der Codegenerierung unterschiedlich behandelt: Nur flüchtige (*volatile*) Elemente werden automatisch initialisiert. Nicht-flüchtige (*non-volatile*) Daten werden bei der Initialisierung nicht überschrieben.

#### **Allen Variablen das Attribut *volatile* zuweisen:**

---

Um sämtlichen Variablen in der Datenbank das Attribut *volatile* zuzuweisen, gehen Sie wie folgt vor:

- Wählen Sie im Komponentenmanager **Tools** → **Database** → **Convert** → **Variables to Volatile**.

Sämtliche Variablen sind nun flüchtig und werden automatisch initialisiert.

### **Allen Parametern das Attribut *non-volatile* zuweisen:**

---

Um sämtlichen Parametern in der Datenbank das Attribut *non-volatile* zuzuweisen, gehen Sie wie folgt vor:

- Wählen Sie im Komponentenmanager **Tools** → **Database** → **Convert** → **Parameters to Nonvolatile**.

Sämtliche Parameter sind nun nicht-flüchtig und werden bei der Initialisierung nicht überschrieben.

### *Suchen und Ersetzen in C-Code- und ESDL-Komponenten*

---

Sie können aus dem Komponentenmanager heraus sämtliche C-Code- und ESDL-Komponenten nach einer Zeichenkette durchsuchen. Auf Wunsch können Sie ein einzelnes Vorkommen der Zeichenkette, alle Vorkommen in einer Komponente oder alle Vorkommen in der gesamten Datenbank ersetzen.

Diese Funktion erleichtert zum Beispiel das Umbenennen von Messages (oder anderen globalen Elementen). Diese sind ja über ihre Namen verknüpft (siehe das Kapitel „Interprozesskommunikation“ im ASCET-Referenzhandbuch); bei einer Namensänderung müssen daher alle Vorkommen geändert werden. Durch die datenbankweite Suchfunktion entfällt die mühsame Suche von Hand.

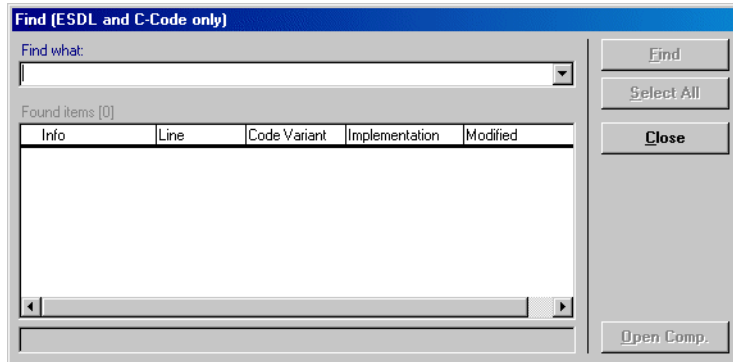
### **Zeichenkette suchen:**

---

- Wählen Sie im Komponentenmanager **Edit** → **Find**

oder

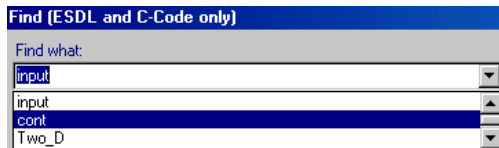
- drücken Sie <CTRL> + <F>.  
Das Fenster „Find (ESDL and C-Code only)“ öffnet sich.



Die Schaltflächen sind, mit Ausnahme von **Close**, noch deaktiviert.

- Geben Sie im Feld „Find what“ die Zeichenkette ein, die Sie suchen wollen.

Wenn Sie schon mehrere Suchen durchgeführt haben, können Sie einen der letzten Suchbegriffe aus dem Kombikästchen des Feldes auswählen.



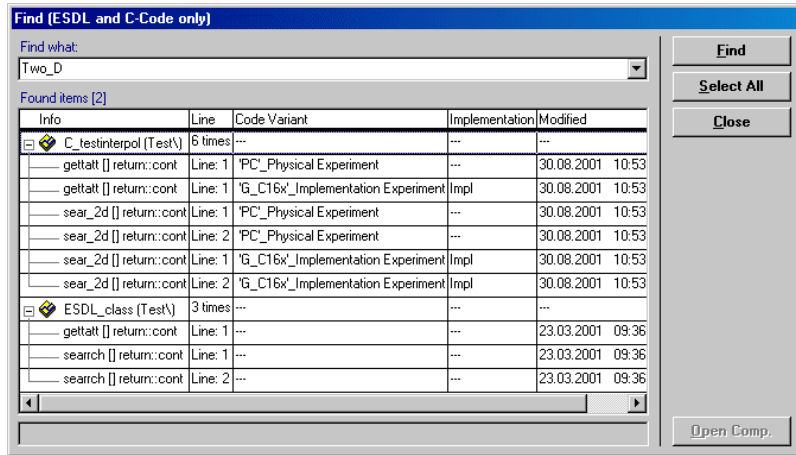
Die Schaltfläche **Find** wird jetzt aktiviert.

- Klicken Sie auf **Find**, um die Suche zu starten.

Wenn Sie die Datenbank das erste Mal durchsuchen, kann die Suche mehrere Minuten dauern.

Die Suche unterscheidet nicht zwischen Groß- und Kleinschreibung. Wird z.B. `cont` eingegeben, werden auch `Cont` und `CONT` gefunden. Die Zeichenkette wird auch dann gefunden, wenn sie Teil eines längeren Wortes ist; beispielsweise liefert die Suche nach `cont` auch alle Vorkommen von `Continous`.

Die Ergebnisse der Suche werden in der Liste „Found items“ angezeigt, die Schaltfläche **Select All** wird aktiviert. Die Anzahl der Komponenten, die die gesuchte Zeichenkette enthalten, wird in eckigen Klammern hinter dem Namen der Liste angegeben.



Die Liste „Found Items“ enthält fünf Spalten.

- In der Spalte „Infos“ werden die gefundenen Komponenten sowie die einzelnen Methoden/Prozesse, in denen die gesuchte Zeichenkette vorkommt, aufgelistet. Standardmäßig ist die Liste der Methoden/Prozesse zugeklappt.
- Die Spalte „Line“ gibt für eine Komponente die Anzahl der Fundstellen an, für die einzelnen Methoden/Prozesse die Nummer der Zeile an, in der die Zeichenkette vorkommt.
- Die Spalte „Code Variant“ gibt bei C-Code-Komponenten an, für welches Target und welches Experiment der Code spezifiziert wurde. Bei ESDL-Komponenten ist die Spalte leer.
- Die Spalte „Implementation“ enthält die verwendete Implementierung.
- Die Spalte „Modified“ enthält Datum und Uhrzeit der letzten Änderung der Komponente.

#### Ansehen der Suchergebnisse:

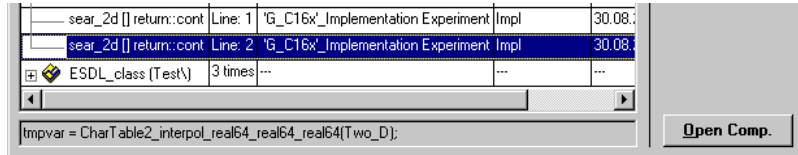
- Klicken Sie in der Liste „Found Items“ auf das Pluszeichen neben einer Komponente  
oder



- doppelklicken Sie auf eine Komponente.  
Die Liste der Methoden/Prozesse wird geöffnet.

- Klicken Sie auf eine Zeile.

In der Fußzeile wird die entsprechende Zeile aus dem Code angezeigt. Außerdem wird die Schaltfläche **Open Comp.** aktiviert.

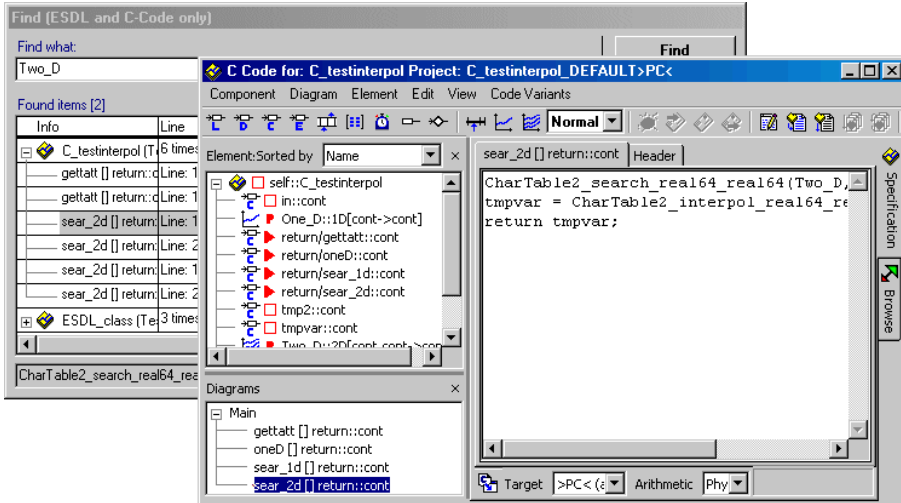


### Öffnen der Komponente aus dem Suchfenster:

Um aus dem Suchfenster heraus die Komponente zu öffnen, gehen Sie wie folgt vor:

- Klicken Sie auf die Schaltfläche **Open Comp.**  
Der Editor für die Komponente wird geöffnet und die Methode/der Prozess wird angezeigt. Die Zeile, die die Zeichenkette enthält, ist markiert.

Wenn es sich um eine C-Code-Komponente handelt, wird der Code für die in den Spalten „Code Variant“ und „Implementation“ angegebene Variante angezeigt, erkennbar an den Inhalten der Kombikästchen (siehe auch Kapitel 4.3).



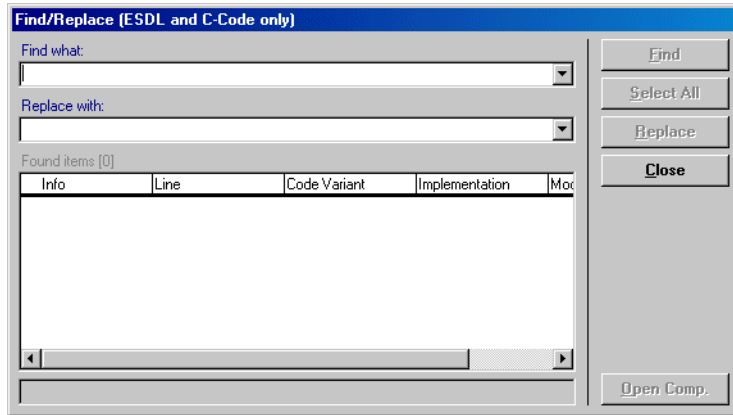
## Ersetzen von ausgewählten Zeichenketten:

### Hinweis

Verwenden Sie die Replace-Funktion vorsichtig, da es **kein Rückgängigmachen** gibt!

- Wählen Sie im Komponentenmanager **Edit** → **Replace**  
oder

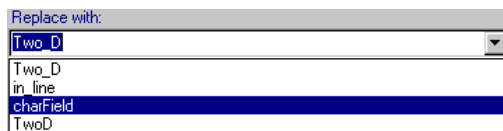
- drücken Sie <CTRL> + <H>. Das Fenster „Find/Replace (ESDL and C-Code only)“ öffnet sich.



Abgesehen von dem Feld „Replace with“ und der Schaltfläche **Replace** entspricht das Fenster in Aufbau und Funktion dem Fenster der einfachen Suche (s. Seite 120).

- Geben Sie im Feld „Find what“ die Zeichenkette ein, die Sie ersetzen wollen.
- Geben Sie im Feld „Replace with“ den neuen Text ein.

Wenn Sie schon mehrmals Text ersetzt haben, können Sie einen der letzten Begriffe aus dem Kombikästchen des Feldes auswählen.



- Klicken Sie auf **Find**, um die Vorkommen der alten Zeichenkette zu suchen.

### **Hinweis**

*Die Reihenfolge der letzten beiden Schritte ist egal.*

Sie können die angezeigten Komponenten wie auf Seite 121 beschrieben aus dem Suchfenster heraus öffnen.

Die Schaltfläche **Replace** ist noch deaktiviert.

- Suchen Sie die Komponente, in der Sie die Zeichenkette ersetzen wollen, und klappen Sie die Methoden-/Prozessliste auf.
- Markieren Sie ein oder mehrere Vorkommen der Zeichenkette.

Jetzt wird die Schaltfläche **Replace** aktiviert.

- Klicken Sie auf **Replace**.

Die ausgewählten Vorkommen der Zeichenkette werden durch den neuen Text ersetzt. Die entsprechenden Zeilen verschwinden aus der Liste „Found Items“.

### **Ersetzen von allen Zeichenketten einer Komponente:**

---

Um alle Vorkommen einer Komponente zu ersetzen, gehen Sie wie folgt vor:

- Suchen Sie die Zeichenkette, die Sie ersetzen wollen, wie auf Seite 122 beschrieben.
- Geben Sie im Fenster „Find/Replace (ESDL and C-Code only)“ im Feld „Replace with“ den neuen Text ein.
- Markieren Sie die Komponente, in der Sie die Zeichenkette ersetzen wollen.

Damit sind alle Vorkommen der Zeichenkette in dieser Komponente ausgewählt.

- Klicken Sie auf **Replace**.

Alle Vorkommen der ausgewählten Komponente (bei C-Code auch in allen Implementierungen und Targets) werden ersetzt.

### **Ersetzen von allen Zeichenketten in der Datenbank:**

---

Um alle Vorkommen in der Datenbank zu ersetzen, gehen Sie wie folgt vor:

- Suchen Sie die Zeichenkette, die Sie ersetzen wollen, wie auf Seite 122 beschrieben.

- Geben Sie im Fenster „Find/Replace (ESDL and C-Code only)“ im Feld „Replace with“ den neuen Text ein.
- Klicken Sie auf **Select All**, um alle Vorkommen in allen Komponenten auszuwählen.
- Klicken Sie auf **Replace**, um alle Vorkommen in der gesamten Datenbank zu ersetzen.

### 2.3.6 Datenbanken verwalten

---

Zu Beginn Ihrer Arbeit an einem neuen Projekt ist es sinnvoll, eine neue Datenbank anzulegen. Dieser Vorgang entspricht dem Wechsel in ein leeres Verzeichnis auf Ihrem Rechner. Sie haben dann die Möglichkeit, sich eine Ihren Wünschen angepasste Ordnerstruktur zu erstellen, bestimmte Objekte neu anzulegen bzw. zu importieren.

#### Hinweis

*Den Namen der aktuellen Datenbank können Sie in der Fußzeile des Komponentenmanagers ablesen.*

#### *Basisaufgaben*

---

#### **Anlegen einer neuen Datenbank:**

---

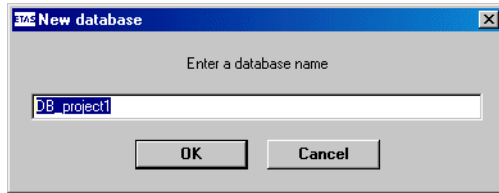
#### Hinweis

*Es empfiehlt sich mit mehreren Datenbanken zu arbeiten, da die Datenmenge klein und übersichtlich gehalten wird und die Performance von ASCET optimal genutzt werden kann.*

- Wählen Sie den Befehl **File** → **New Database**  
oder
- klicken Sie auf die Schaltfläche **New**  
oder



- drücken Sie <CTRL> + <N>.  
Das Fenster „New database“ erscheint:



- Tragen Sie einen Namen ein.
- Klicken Sie auf **OK**.  
Die neue, bis auf den Datenbanknamen und den Ordner Default noch leere Datenbank wird geöffnet. In dem unter „Database Path“ eingestellten Verzeichnis (s. Seite 42) wird ein Unterverzeichnis mit dem Namen der Datenbank angelegt.
- Gehen Sie jetzt vor, wie unter „Datenbankeinträge verwalten“ auf Seite 78 beschrieben.

### Laden einer Datenbank:

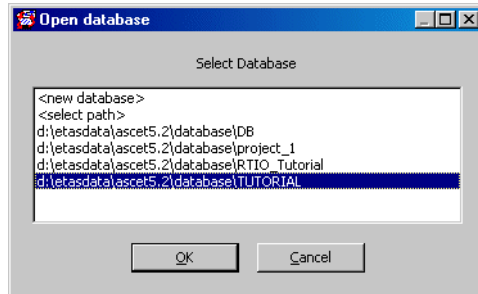
#### Hinweis

*Datenbanken aus älteren Programmversionen können ebenfalls geladen werden (siehe „Umwandeln einer ASCET-SD 4.x-Datenbank:“ auf Seite 134).*

- Wählen Sie **File** → **Open Database**  
oder
- klicken Sie auf die Schaltfläche **Open**  
oder



- drücken Sie <CTRL> + <O>.  
Das Fenster „Open database“ erscheint:



In der Liste werden alle im Standardverzeichnis befindlichen Datenbanken enthalten.

- Wählen Sie den gewünschten Eintrag und klicken Sie auf **OK**.

Möchten Sie eine Datenbank laden, die sich nicht im Standardverzeichnis befindet, so können Sie über den Eintrag <select path> einen entsprechenden Pfad auswählen.

- Bestätigen Sie mit **OK**.

Die Datenbank wird geladen.

### Hinweis

Sie können auch über dieses Fenster eine neue Datenbank anlegen. Wählen Sie hierzu den Eintrag <new database>.

### Speichern einer Datenbank:

- Wählen Sie **File** → **Save Database**

oder



- klicken Sie auf die Schaltfläche **Save**

oder

- drücken Sie <CTRL> + <S>.

Alle Änderungen werden in der Datenbank gespeichert. Mit diesem Befehl sollten Sie Ihre Datenbank regelmäßig sichern.

### Hinweis

Über die Benutzeroptionen (Menü **Tools** → **Options**, Knoten „Options“, s. Seite 42) können Sie die automatische Speicherfunktion aktivieren und ein gewünschtes Zeitintervall eingeben. Die Datenbank wird auch automatisch beim Verlassen des Komponentenmanagers gespeichert.

### Speichern einer Datenbank unter anderem Namen:

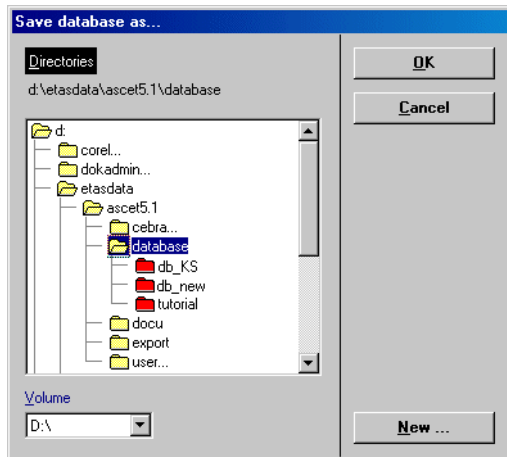
Sie können die Datenbank auch unter einem beliebigen Namen speichern. Auf diese Weise können Sie z.B. eine Sicherheitskopie der gesamten Datenbank und ihren ergänzenden Dateien erstellen.

- Wählen Sie im Komponentenmanager **File** → **Save Database As**.

Das Pfadauswahlfenster erscheint.

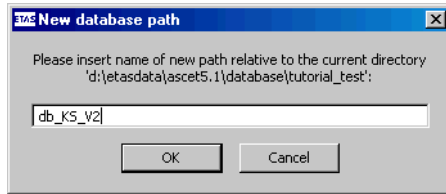
- Wählen Sie das Verzeichnis, in dem die Datenbank abgelegt werden soll.

Sie dürfen *keine* Datenbank auswählen.



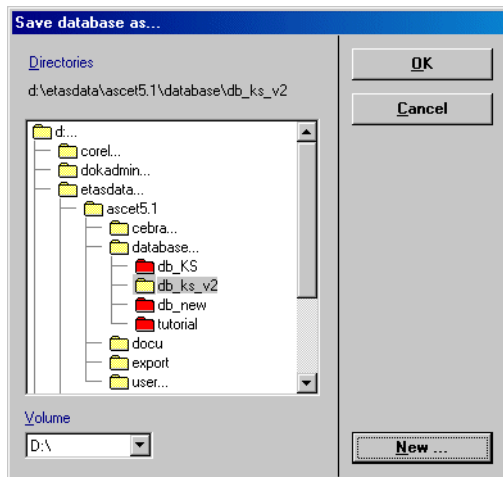


- Klicken Sie auf die Schaltfläche **New**, um ein neues leeres Verzeichnis zu erzeugen.



- Geben Sie einen Namen ein und klicken Sie **OK**.

Das neue Verzeichnis wird angelegt und im Pfadauswahlfenster als Zielverzeichnis für die Sicherungskopie ausgewählt. Die Farbe des Verzeichnisses ändert sich nach dem Kopiervorgang automatisch.



- Klicken Sie im Pfadauswahlfenster auf **OK**.

### **Hinweis**

Wenn Sie ein bestehendes Verzeichnis ausgewählt haben, wird eine Warnung angezeigt, dass der gesamte Inhalt gelöscht wird, wenn Sie die Sicherung fortsetzen. Sie können die Erzeugung der Kopie hier abbrechen.

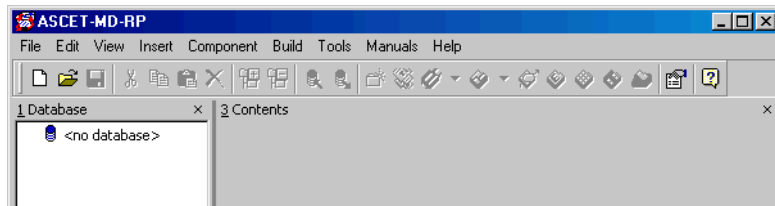
Die Inhalte der aktuellen Datenbank werden kopiert und die neue Datenbank wird angelegt und geladen (siehe Fußzeile im Komponentenmanager).

### **Schließen einer Datenbank:**

Sie können die Datenbank, mit der Sie arbeiten, schließen, ohne ASCET zu beenden.

- Wählen Sie **File** → **Close Database**.

Die Datenbank wird geschlossen. Der Komponentenmanager bleibt geöffnet, aber die meisten Schaltflächen sind deaktiviert.



### **Löschen einer Datenbank:**

Sie können eine Datenbank *nicht* aus ASCET heraus löschen.

- Falls die Datenbank, die Sie löschen wollen, in ASCET geöffnet ist, schließen Sie sie.

Wenn Sie versuchen, eine Datenbank zu löschen, die in ASCET geöffnet ist, wird eine Fehlermeldung angezeigt.

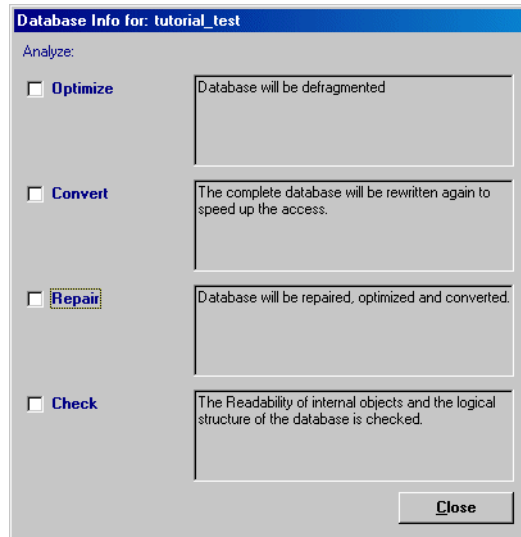
- Wählen Sie im Windows-Explorer das Verzeichnis (z.B. . . \ETASData\Ascet5.2), in dem die Datenbank abgelegt ist, die Sie löschen wollen.
- Löschen Sie im Windows-Explorer das Unterverzeichnis, das die Datenbank enthält.

## Optimieren einer Datenbank:

---

- Wählen Sie im Komponentenmanager **Tools** → **Database** → **Performance Utilities**.

Das Dialogfenster „Database Info“ öffnet sich.



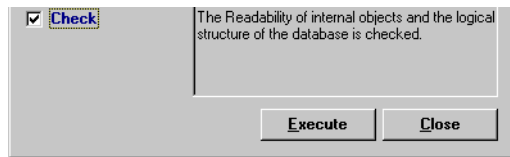
- Aktivieren Sie die Option **Optimize**.  
Dieser Befehl fasst die durch vielfaches Editieren entstandenen Datenbankbruchstücke wieder zusammen (Defragmentierung).
- Aktivieren Sie die Option **Convert**.  
Dieser Befehl bereinigt die Referenzierungen der Datensätze untereinander („X benutzt Y“) nach Änderungen der Datenbankstruktur und beschleunigt so den Zugriff.

In den meisten Fällen wird eine reguläre Optimierung für eine akzeptable Leistung Ihrer Datenbank ausreichen. Nur wenn Sie größere Leistungsmängel feststellen, sollten Sie überlegen, die Datenbank umzuwandeln.

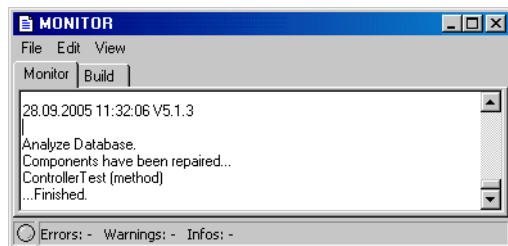
### **Hinweis**

*Das Umwandeln einer großen Datenbank kann sehr zeitaufwendig sein. Es wird empfohlen, den Umwandlungsvorgang über Nacht laufen zu lassen.*

- Aktivieren Sie die Option **Repair**.  
Dieser Befehl baut die Datenbank komplett neu auf. Beschädigungen Ihrer Datenbank, z.B. zerstörte Referenzen, werden repariert. Die Reparatur beinhaltet immer auch Funktionen **Optimize** und **Convert**; die entsprechenden Optionsfelder werden gesperrt.
- Aktivieren Sie die Option **Check**.  
Dieser Befehl kontrolliert die Strukturen und Referenzierungen in der Datenbank und protokolliert die Ergebnisse im Monitorfenster.  
Wenn Sie eine Option aktiviert haben, erscheint die Schaltfläche **Execute**.



- Klicken Sie auf die Schaltfläche **Execute**.  
Das gewählte Datenbankverwaltungsprogramm wird gestartet. Die Ergebnisse werden im ASCET-Monitor gezeigt.



## Vergleichen zweier Datenbanken:

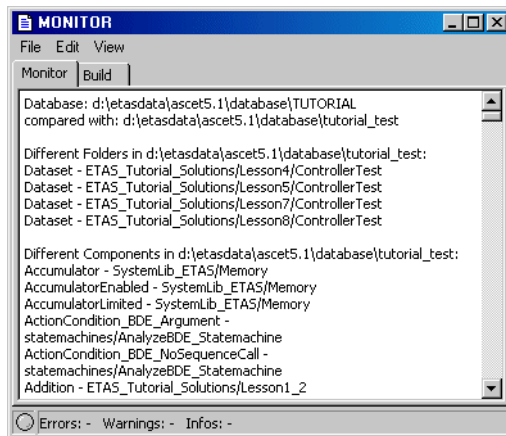
Sie können Datenbanken miteinander vergleichen. Dies ist hilfreich, wenn Sie mit verschiedenen Kopien oder Entwicklungsrichtungen einer Datenbank oder in Entwicklungsteams arbeiten.

- Wählen Sie im Komponentenmanager **Tools** → **Database** → **Compare Database**. Das Fenster „Compare database“ erscheint.
- Wählen Sie die Datenbank aus, die Sie mit der aktuellen vergleichen wollen.
- Starten Sie den Vergleich mit **OK**.

### Hinweis

*Der Vergleich von großen Datenbanken kann sehr zeitaufwendig sein.*

Die Datenbanken werden verglichen. Eine Liste von Einträgen und Projekten, in denen sich die Datenbanken voneinander unterscheiden, erscheint im Monitorfenster.



### Datenbanken aus früheren ASCET-Versionen verwenden

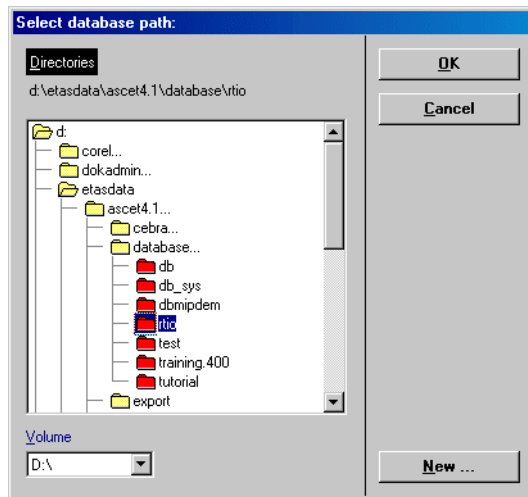
Sie können existierende Datenbanken aus früheren ASCET-SD Versionen mit ASCET 5.2 verwenden. Allerdings müssen diese Datenbanken konvertiert werden, da das bestehende ASCET-Datenbankformat mit Formaten aus früheren ASCET-Versionen nicht kompatibel ist.

Die Umwandlung einer bestehenden Datenbank wird automatisch vorgenommen, wenn Sie die Datenbank, die mit ASCET-SD 4.x erstellt wurde, zum ersten Mal öffnen.

### Umwandeln einer ASCET-SD 4.x-Datenbank:

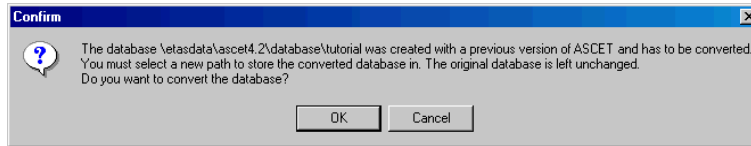
---

- Öffnen Sie die Datenbank wie unter „Laden einer Datenbank:“ auf Seite 126 beschrieben.
- Wählen Sie im Fenster „Open Database“ den Eintrag <select path> und klicken Sie **OK**. Das Fenster „Select database path“ öffnet sich.



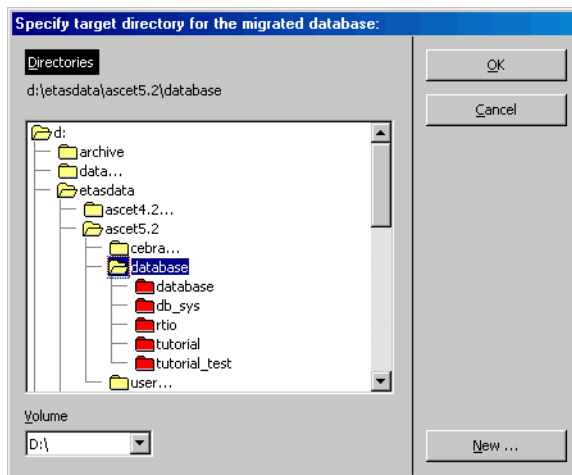
- Wählen Sie das Verzeichnis aus, in dem die alte Datenbank abgelegt ist.

- Bestätigen Sie mit **OK**.  
Das System bittet um eine Bestätigung zur Umwandlung Ihrer Datenbank.



Sie können den Vorgang mit **Cancel** abbrechen.

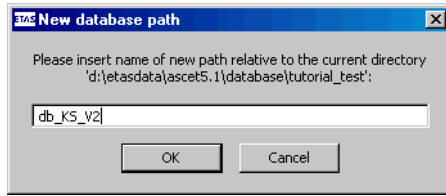
- Bestätigen Sie die Umwandlung mit **OK**.  
Das Fenster „Specify target directory for the migrated database“ öffnet sich. Es entspricht dem Fenster „Select database path“.



### Hinweis

*Wenn Sie hier ein nicht-leeres Verzeichnis auswählen erscheint eine Warnung. Bestätigen Sie diese und erzeugen Sie im Pfadauswahlfenster ein neues, leeres Verzeichnis.*

- Klicken Sie auf die Schaltfläche **New**, um ein neues leeres Verzeichnis zu erzeugen.



- Geben Sie einen Namen ein und klicken Sie **OK**.  
Das neue Verzeichnis wird angelegt und als Zielverzeichnis für die Umwandlung ausgewählt.
- Starten Sie den Konvertierungsvorgang, indem Sie im Pfadauswahlfenster auf **OK** klicken.  
Die Datenbank wird in das ASCET 5.2-Format umgewandelt und in das neu angelegte Verzeichnis geschrieben. Die Dauer des Konvertierungsvorgangs hängt von der Menge der Daten ab.  
Die originale alte Datenbank bleibt unverändert erhalten.
- Bestätigen Sie zum Abschluß die Meldung **Finished converting the database** mit **OK**.

Datenbanken, die mit ASCET-SD vor der Version 4.0 erstellt wurden, können nicht in ASCET 5.2 importiert werden, da die aktuelle Version keine Schnittstellen zu den damaligen Datenbankformaten enthält. Wenn Sie versuchen, eine solche Datenbank wie auf Seite 134 beschrieben zu öffnen, erhalten Sie folgende Fehlermeldung:

```
This file is compatible with ASCET-SD V3.0 or earlier.
It cannot be imported directly. Convert it to a data-
base using your old version of ASCET-SD, then open
this database.
```



## Umwandeln einer Datenbank der Version ASCET-SD 4.0 oder früher:

---

Wenn Sie ASCET-SD 4.x noch haben, können Sie die ganz alte Datenbank dennoch mit ASCET 5.2 verwenden. Gehen Sie wie folgt vor:

- Starten Sie ASCET-SD 4.x.  
Diese Versionen enthalten noch Schnittstellen zu den ganz alten Datenbankformaten.
- Öffnen Sie die alte Datenbank.  
Die Datenbank wird dabei in das Format der entsprechenden Version von ASCET-SD konvertiert.
- Schließen Sie ASCET-SD 4.x.
- Starten Sie ASCET 5.2.
- Öffnen Sie die eben nach ASCET-SD 4.x konvertierte Datenbank wie auf Seite 134 beschrieben.

### *ANSI C-Umwandlung*

---

In ASCET wird für Versionen höher als 2.x ANSI-C-Verträglichkeit für alle Namen gefordert. Beim Hinzufügen neuer Einträge oder Umbenennen bestehender müssen gültige ANSI-C-Bezeichnungen verwendet werden.

Die Umwandlung von Ordner- und Eintragsnamen in bestehenden Datenbanken muss explizit ausgelöst werden, nachdem die Datenbank umgewandelt worden ist. Alle Ordner- und Eintragsnamen werden automatisch umgewandelt; Sonderzeichen werden gemäß Tab. 2-4 dargestellt.

#### **Sonderzeichen**

Ä, ä, Ö, ö, Ü, ü, ß

<Leer>, <Bindestrich>, <Punkt>

#### **In ANSI C abgebildet**

Ae, ae, Oe, oe, Ue, ue, ss

in jedem Falle <Unterstrich>

**Tab. 2-4** ANSI-C-Zeichenumwandlung

## Umwandeln einer Datenbank nach ANSI-C:

---

- Öffnen Sie die Datenbank, die Sie konvertieren wollen.

- Wählen Sie **Tools** → **Database** → **Convert** → **All Names to ANSI C**.

Die Namen aller Ordner, Datenbankeinträge, Methoden, Messages, Variablen usw. werden nach ANSI-C konvertiert, wobei die obige Umwandlungstabelle für alle Sonder-, Interpunktions- und Leerzeichen verwendet wird.

### **Hinweis**

*Bei großen Datenbanken sollten Sie ein Komprimieren Ihrer Datenbank nach der Umwandlung in Betracht ziehen, um den Zugriff zu beschleunigen (siehe Abschnitt „Datenbank-Serviceroutinen“ auf Seite 138).*

### Auflösen von Namenskonflikten

In einigen Fällen, wo sich Namen in einer Datenbank aus Version 2.x nur durch Interpunktions- oder Leerzeichen unterscheiden, führt die Standardumwandlung nach ANSI-C zu Namenskonflikten.

Beispielsweise wird aus „Integrator I21“ und „Integrator-I21“ in beiden Fällen „Integrator\_I21“. Da Namen von Einträgen jeweils nur einmal zu vergeben sind, weist der Konvertierungsalgorithmus dem zweiten Eintrag den Namen „Integrator\_211“ zu.

Sie können diese Lösung für Namenskonflikte bei der Umwandlung nach ANSI-C akzeptieren oder die entsprechende Umwandlungsdatei ändern. Die Datei `mapping.ini` befindet sich im ASCET-Verzeichnis und kann mit jedem Standard-Texteditor bearbeitet werden.

### **Hinweis**

*Die Umwandlung von Leerzeichen kann nicht verändert werden; sie werden immer zu Unterstrichen.*

### Datenbank-Serviceroutinen

ASCET bietet Werkzeuge zur Datenbankreparatur, -komprimierung und zum Erstellen einer Sicherungskopie. Zusätzlich können Sie generierten Code für die gesamte Datenbank verwerfen oder ein neues aufbauen und Vergleichen von Datenbanken erzwingen. Dieser Abschnitt erklärt die Datenbank-Dienstprogramme, die vom Komponentenmanager aus benutzt werden.

Informationen zum Umwandeln von Datenbanken siehe Abschnitt „Umwandeln einer ASCET-SD 4.x-Datenbank:“ auf Seite 134.

## Generierter Code

---

Sie können alle Codes, die beim Experimentieren erzeugt wurden, aus der Datenbank löschen, um die Datenbankgröße zu verringern und den Code neu zu generieren.

### Verwerfen des in der Datenbank gespeicherten Codes:

---

- Wählen Sie im Komponentenmanager **Build** → **Clean All**.  
Sie werden gebeten, das Löschen des generierten Codes aus der Datenbank zu bestätigen.
- Bestätigen Sie mit **OK**.  
Sämtlicher generierter Code wird aus der Datenbank entfernt.

Bei der Codegenerierung wird eine Make-Anweisung ausgeführt, also nur Code für neue oder veränderte Teile erzeugt und kompiliert. Möglicherweise wollen Sie sicherstellen, dass eine Build-Anweisung ausgeführt, d.h. Code insgesamt erzeugt und kompiliert, wird. Dabei wird vorher erzeugter Code nicht gelöscht.

### Erzwingen einer Gesamtcompilierung:

---

- Wählen Sie **Build** → **Touch All**.  
Alle Komponenten in der Datenbank werden als geändert markiert (auch wenn nichts wirklich geändert wurde). Damit ist sichergestellt, dass beim nächsten Mal neuer Code erzeugt und kompiliert wird.

#### Hinweis

*Eine Gesamtcompilierung kann sehr lange dauern, wenn Sie an einer großen Datenbank arbeiten.*

## Durchsuchen einer Datenbank

---

Sie können die gesamte Datenbank nach Informationen zu Beziehungen zwischen Einträgen oder Operatorimplementierungen durchsuchen. Zum Beispiel ist im Komponentenmanager vollständig sichtbar, welche Einträge von anderen benutzt oder referenziert werden. Sie können die Datenbank mittels komplexer Suchkriterien nach solchen Beziehungen durchblättern; dabei werden nur die Einträge angezeigt, die die Kriterien erfüllen.

Beim Durchblättern einer Datenbank durchsucht ASCET immer die gesamte Datenbank, unabhängig von ausgewählten Ordnern oder Einträgen. Platzhalter sind bei der Angabe der Kriterien erlaubt, z.B. werden durch Eingeben von `*acc*` alle Einträge und Ordner gefunden, die die Zeichenkette `acc` in ihrem Namen enthalten. Groß-/Kleinschreibung spielt keine Rolle.

Die Suche nach Operatorimplementierungen (**Tools** → **Database** → **List Operator Implementations**) ist am Ende des Abschnitts „Operatorimplementierungen“ beschrieben (Seite 544).

### Durchsuchen einer Datenbank (Fenster „Query“):

---

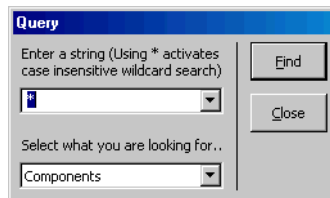
- Öffnen Sie den Komponentenmanager für die zu durchsuchende Datenbank.

- Wählen Sie **Edit** → **Query**

oder

- drücken Sie `<CTRL> + <Q>`.

Das Fenster „Query“ öffnet sich.

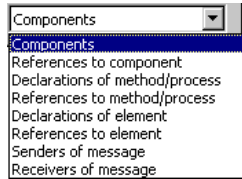


- Geben Sie im Feld „Enter a string“ den Suchbegriff ein.

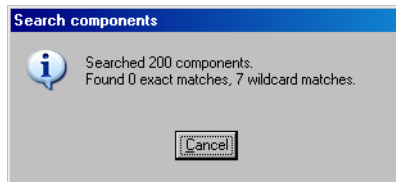
Der Suchbegriff sollte der Name eines Datenbankeintrags, einer Methode, eines Prozesses oder Elements sein.

Das zum Feld gehörige Kombikästchen enthält frühere Suchbegriffe, die Sie auswählen können.

- Wählen Sie im Kombikästchen „Select what you are looking for“ ein Suchkriterium (siehe Seite 143) aus.



- Klicken Sie auf **Find**, um die Suche zu starten. Während die Datenbank durchsucht wird, wird das Fenster „Search <objects>“ angezeigt. Sie können darin die Suche mit **Cancel** abbrechen.



Die gefundenen Informationen werden in einem Dialogfenster angezeigt.

### Durchsuchen einer Datenbank (Schaltflächenleiste):

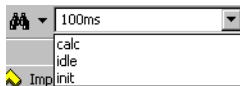
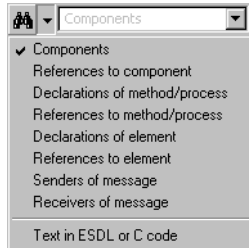
Sie können die Suche auch aus der Schaltflächenleiste des Komponentenmanagers heraus durchführen.



- Klicken Sie im Komponentenmanager auf die Pfeil-Schaltfläche neben **Search - <information type>**.

Eine Auswahlliste wird aufgeklappt. Das aktuell für die Suche gewählte Suchkriterium (s. Seite 143) ist mit einem Häkchen gekennzeichnet.

- Wählen Sie ein Suchkriterium.  
Das gewählte Kriterium wird in blasser Schrift im Kombikästchen angezeigt.



- Geben Sie im Kombikästchen einen neuen Suchstring ein

oder

- wählen Sie einen früher verwendeten Suchstring.
- Klicken Sie auf die Schaltfläche **Search - <information type>**, um die Suche zu starten.



Während die Datenbank durchsucht wird, wird das Fenster „Search <objects>“ angezeigt. Sie können darin die Suche mit **Cancel** abbrechen.

Die gefundenen Informationen werden in einem Dialogfenster angezeigt.

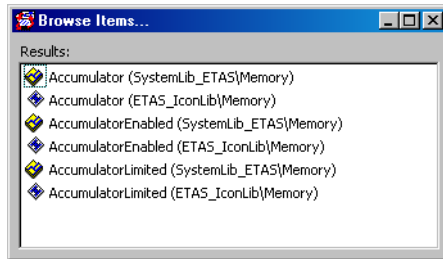
### Hinweis

Wenn Sie den Informationstyp `Text in ESDL or C code` gewählt haben, öffnet **Search - Text in ESDL or C code** *nicht* das Fenster mit den Suchergebnissen, sondern das Fenster „Find (ESDL and C-Code only)“. Machen Sie weiter wie in „Zeichenkette suchen:“ auf Seite 118 beschrieben.

Nach den folgenden Informationstypen kann gesucht werden:

### Components:

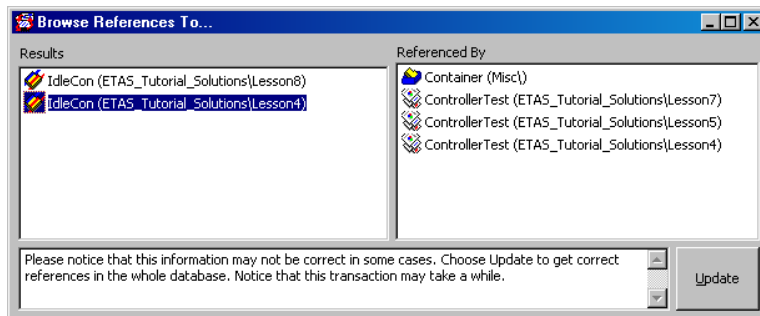
Mit diesem Suchkriterium finden Sie alle Einträge in der Datenbank, deren Namen zum Suchstring passen. Die Informationen werden in folgendem Fenster angezeigt:



Das Dialogfenster „Browse Items“ zeigt die vollständigen Namen der passenden Komponenten sowie ihre Datenbankpfade an. Durch Anklicken eines Namens wird die Komponente angezeigt und im Komponentenmanager hervorgehoben.

### References to component:

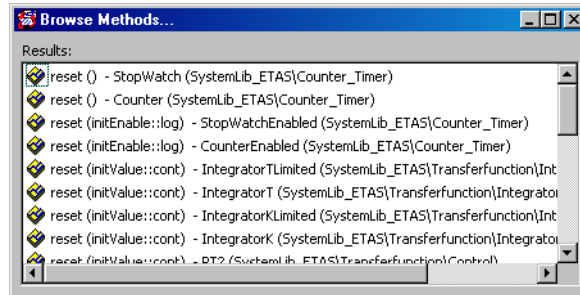
Mit diesem Suchkriterium finden Sie alle Referenzen auf Komponenten in der Datenbank, deren Namen zum Suchstring passen. Die Informationen werden in folgendem Fenster angezeigt.



Die Einträge, die zum Suchstring passen, erscheinen in der Fläche „Results“ des Dialogfensters. Durch Anklicken eines Eintrags in dieser Fläche werden die referenzierenden Einträge in der Fläche „Referenced By“ gezeigt. Durch Klicken auf eine Komponente in einer der beiden Listen wird sie im Komponentenmanager angezeigt und hervorgehoben.

## Declarations of method/process:

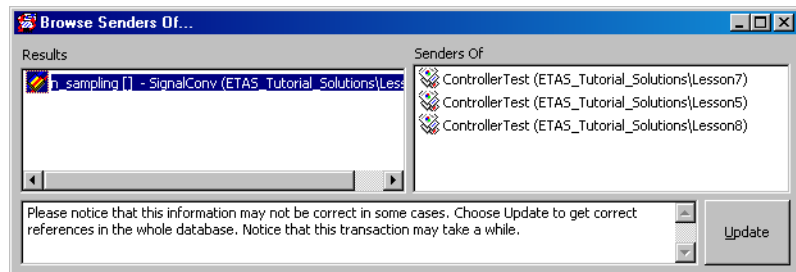
Mit diesem Suchkriterium finden Sie alle Methoden oder Prozesse, die zum Suchstring passen. Diese werden im folgendem Fenster angezeigt:



Das Dialogfenster „Browse Methods“ zeigt den voll ausgebildeten Methoden-selektor und den Datenbankpfad der definierenden Komponente. Durch Klicken auf eine Methode wird die definierende Komponente im Komponentenmanager angezeigt und hervorgehoben.

## References to method/process:

Mit diesem Suchkriterium finden Sie alle Referenzen auf Methoden oder Prozesse in der Datenbank, deren Namen zum Suchstring passen. Sie werden in folgendem Fenster angezeigt:

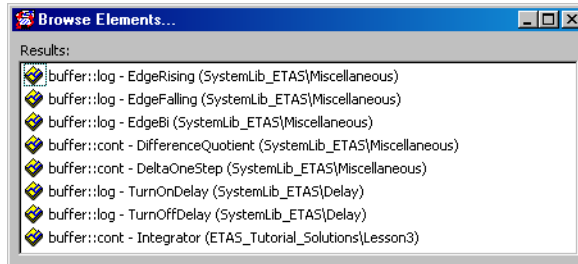


Das Dialogfenster „Browse Senders Of“ zeigt die zum Suchstring passenden Einträge in der Fläche „Results“. Methoden-/Prozess-Selektoren werden gemeinsam mit ihren definierenden Komponenten aufgelistet. Durch Anklicken einer Methode/eines Prozesses werden die Komponenten, die die ausgewählte Methode/Prozess aufrufen, in der Fläche „Senders Of“ des Dialogfensters aufgelistet. Durch Anklicken eines Eintrags in einer der Listen wird die betreffende Komponente im Komponentenmanager angezeigt und hervorgehoben.



### Declarations of element:

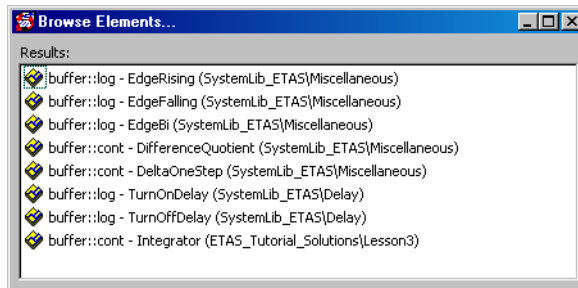
Mit diesem Suchkriterium finden Sie alle Komponenten, in denen die zum Suchstring passenden Elemente (z.B. Variablen oder Argumente) deklariert werden. Sie werden in folgendem Fenster angezeigt:



Das Dialogfenster „Browse Elements“ zeigt die Elemente zusammen mit ihrer definierenden Komponente und dem Datenbankpfad dieser Komponente. Durch Anklicken eines Elements wird die betreffende Komponente im Komponentenmanager angezeigt und hervorgehoben.

### References to element:

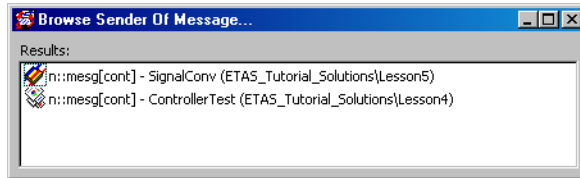
Mit diesem Suchkriterium finden Sie alle Komponenten, die Referenzen auf die zum Suchstring passenden Elemente enthalten. Sie werden in folgendem Fenster angezeigt:



Das Dialogfenster „Browse Elements“ zeigt die Elemente zusammen mit den Komponenten, in denen sie verwendet werden, und den Datenbankpfaden dieser Komponenten. Durch Anklicken eines Elements wird die betreffende Komponente im Komponentenmanager angezeigt und hervorgehoben.

### Senders of message:

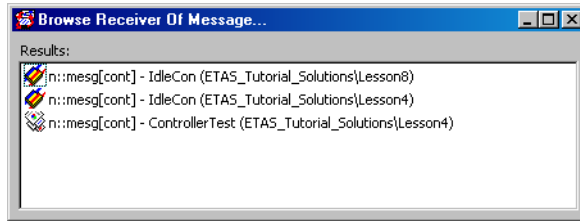
Mit diesem Suchkriterium finden Sie alle Module oder Projekte, die die zum Suchstring passenden Messages versenden. Sie werden in folgendem Fenster angezeigt:



Die Messages werden zusammen mit dem definierenden Modul/Projekt angezeigt. Durch Anklicken einer Message wird das betreffende Modul/Projekt im Komponentenmanager angezeigt und hervorgehoben.

### Receivers of message:

Dieses Suchkriterium arbeitet analog zum vorigen, jedoch für empfangene Messages.



## 2.3.7 Datenbankzugriff

Es ist möglich, Zugriffsberechtigungen sowohl für einzelne Komponenten als auch ganze Ordner zu setzen. Zugriffsberechtigungen lassen sich durch ein Passwort schützen, so dass nur Benutzer, die das Passwort kennen, die Berechtigung ändern können. Auf diese Weise wird der Austausch von ASCET-Daten vertraulich.

Ist beispielsweise eine gewisse Komponente „nur ausführbar“, kann sie zwar zum Aufbau anderer verwendet, aber nicht angesehen werden; die benutzten Algorithmen sind unsichtbar.

Folgende Zugriffsberechtigungen sind möglich:

**Read:** Der Datenbankeintrag kann mit einem entsprechenden Eintragseditor angesehen werden.

**Write:** Einträge können zur Datenbank hinzugefügt und gelöscht werden. Einträge können im entsprechenden Editor bearbeitet werden.

**Calibration:** Einträge sind in Experimenten kalibrierbar.

**Execute:** Mit Einträgen kann man experimentieren. Dies gilt auch für referenzierte Einträge; wenn ein Benutzer einen Eintrag nicht ausführen darf, kann nicht mit einem anderen, umschließenden Eintrag experimentiert werden.

**Code Generation:** Diese Option erlaubt die Codegenerierung auch dann, wenn der Schreibzugriff nicht gesetzt ist. Die Komponente kann, wenn der Lesezugriff gesetzt ist, angeschaut, aber nicht verändert werden; Experimentieren ist möglich.

Es gelten stets die letzten Änderungen. Wird zum Beispiel die Berechtigung für eine Komponente geändert und später für den gesamten Ordner, gilt damit die frühere Berechtigung für die Komponente nicht mehr.

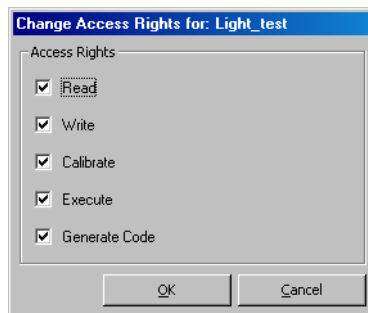
Ihre Zugriffsberechtigung für den gewählten Eintrag sind in der Fußzeile des Komponentenmanagers zu sehen.

### **Ändern von Zugriffsberechtigungen für einen Ordner oder Eintrag:**

---

- Wählen sie in der Liste „1 Database“ des Komponentenmanagers den gewünschten Ordner oder Eintrag aus.
- Wählen Sie in der Menüzeile **Component** → **Access Rights**.

Das Dialogfenster „Overwrite Access Rights for“ öffnet sich. Es zeigt die aktuelle Zugriffsberechtigung für den gewählten Eintrag.



- Aktivieren Sie Optionen, um Zugriffsrechte zu gewähren.

- Deaktivieren Sie Optionen, um Zugriffsrechte zu entfernen.
- Klicken Sie auf **OK**, um die Änderungen zu bestätigen.

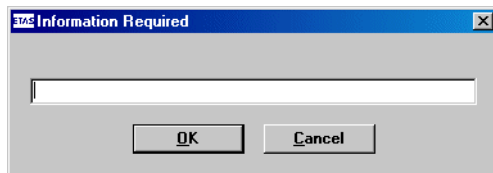
Zugriffsberechtigungen können für jeden Datenbankeintrag gesetzt werden. Änderungen sind nur beschränkt möglich, wenn der Passwortschutz aktiviert ist. Passwortschutz kann separat für jeden Hauptordner aktiviert werden. Wenn der Passwortschutz aktiv ist, kann ein Benutzer nur dann Zugriffsberechtigungen ändern, wenn er das Passwort kennt.

### Hinweis

*Wenn der Passwortschutz aktiv ist, müssen Sie für jede Änderung der Zugriffsrechte ein Passwort eingeben. Die ist ziemlich lästig, wenn Sie mehrere ändern wollen. Daher ist es effizienter, die Zugriffsrechte zuerst zu setzen und dann den Passwortschutz zu aktivieren.*

### Aktivieren des Passwortschutzes:

- Wählen Sie in der Liste „1 Database“ des Komponentenmanagers einen Hauptordner.
- Wählen Sie **Component** → **Password**.  
Sie werden um eine Bestätigung für die Passwortschutz-Aktivierung gebeten.
- Bestätigen Sie im Fenster „Confirm“ mit **Yes**.  
Das Fenster „Information Required“ öffnet sich.



- Geben Sie ein Passwort ein, das mindestens sechs Zeichen lang ist, und klicken Sie auf **OK**.  
Ein Fenster zur Verifikation des Passworts erscheint.

- Geben Sie das Passwort noch einmal ein und klicken Sie auf **OK**, um den Passwortschutz zu aktivieren.

Von da ab muss ein Benutzer das Passwort eingeben, sobald er die Zugriffsberechtigung in diesem Ordner ändern will.

Der Passwortschutz kann deaktiviert werden, indem das Passwort wieder eingegeben wird.

#### **Deaktivieren des Passwortschutzes:**

---

- Wählen Sie in der Liste „1 Database“ des Komponentenmanagers den Hauptordner, dessen Passwortschutz Sie deaktivieren wollen.
- Wählen Sie **Component** → **Password**.  
Das Fenster „Information Required“ öffnet sich.
- Geben Sie das Passwort ein und klicken Sie auf **OK**.  
Sie werden aufgefordert, die Deaktivierung zu bestätigen.
- Bestätigen Sie mit **Yes**.
- Der Passwortschutz wird deaktiviert.

## 2.4 Das ASCET-Monitorfenster

---

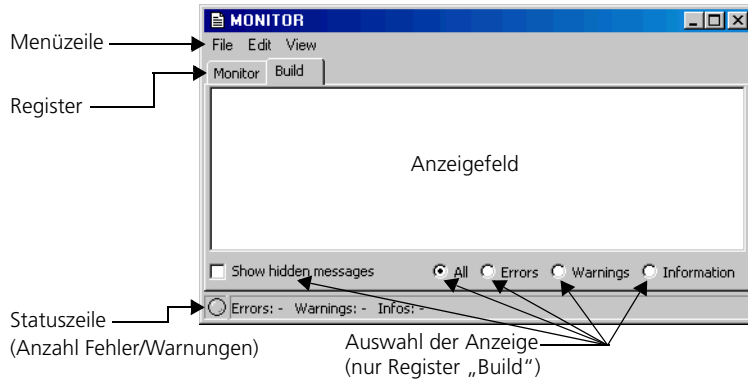
Im ASCET-Monitorfenster werden in zwei Registern die Ergebnisse verschiedener Operationen angezeigt.

Im Register „Monitor“ sind das unter anderem folgende Operationen:

- Codegenerierung, Experimentieren (s. Kapitel 4.1.10, Kapitel 4.2.5, Kapitel 4.5.4 und Kapitel 4.8.10)
- Analyse von Diagrammen (s. Seite 275 und Kapitel 4.4.3)
- Datenbankoptimierungen (s. Seite 131)
- Export- und Importvorgänge (s. Kapitel 2.3.3 und Kapitel 2.3.4)
- Suche nach Operatorimplementierungen (s. Seite 544)

Im Register „Build“ sind das folgende Operationen:

- Codegenerierung, Experimentieren
- Analyse von Diagrammen



### Menüfunktionen:

- **File**

- *Open* (<CTRL> + <O>)

Öffnet eine bestehende Log-Datei. Die Datei wird immer im Register „Monitor“ geöffnet, auch wenn gerade das Register „Build“ angezeigt wird. Der Inhalt im Register „Monitor“ wird beim Laden überschrieben.

- *Save* (<CTRL> + <S>)

Speichert den Inhalt des angezeigten Registers in der Datei `ASCET_Monitor.log` (Register „Monitor“) bzw. `CodeGeneration.log` (Register „Build“).

- *Save As*

Speichert den Inhalt des angezeigten Registers unter einem beliebigen Namen.

- *Show in Editor*

Öffnet die Log-Datei des aktuellen Registers in einem Texteditor. Der Texteditor kann im ASCET-Optionsfenster, Knoten „External Tools“ (s. Seite 61), gewählt werden.

- *Reposition*

Versetzt das Monitorfenster in die Ausgangsgröße und -position auf dem Bildschirm.

- *Exit* (<ALT> + <F4>)

Schließt das ASCET-Monitorfenster.

- **Edit**  
(steht im Register „Monitor“ auch als Kontextmenü zur Verfügung)

#### Hinweis

*Dieses Menü ist nur im Register „Monitor“ verfügbar.*

- *Cut* (<CTRL> + <X>)  
Schneidet den ausgewählten Text aus dem Anzeigefeld aus.  
Dieser Befehl kann mit <CTRL> + <Z> rückgängig gemacht werden.
- *Copy* (<CTRL> + <C>)  
Kopiert Text aus dem Anzeigefeld in die Zwischenablage.
- *Paste* (<CTRL> + <V>)  
Fügt Text aus der Zwischenablage in das Anzeigefeld.  
Dieser Befehl kann mit <CTRL> + <Z> rückgängig gemacht werden.
- *Select All* (<CTRL> + <A>)  
Wählt den gesamten Inhalt des Anzeigefelds aus.
- *Find/Replace* (<CTRL> + <F>)  
Sucht/ersetzt Text im Anzeigefeld.
- *Clear* (<CTRL> + <R>)  
Löscht den Inhalt des Anzeigefelds.

- **View**

#### Hinweis

*Dieses Menü ist nur im Register „Build“ verfügbar.*

- *Collapse All*  
Die Anzeige wird weitestmöglich zugeklappt.
- *Expand All*  
Die Anzeige wird vollständig aufgeklappt.

#### **Kontextmenüs:**

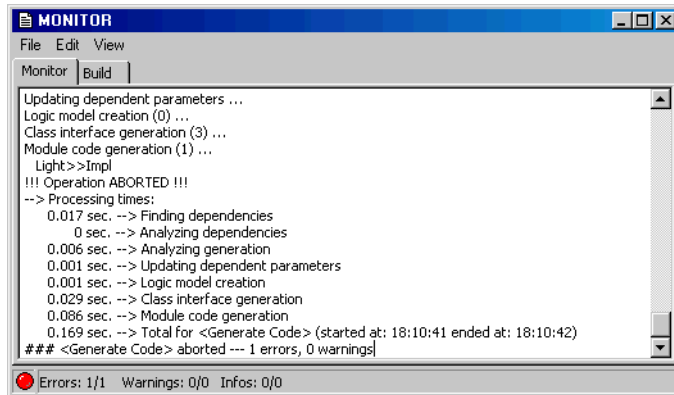
- Register „Monitor“  
Das Kontextmenü im Register „Monitor“ enthält dieselben Funktionen wie das Menü **Edit**.
- Register „Build“

- *Open*  
Öffnet den Editor für die Komponente, die die Meldung verursacht hat.
- *File Out*  
Speichert den Inhalt des Registers unter einem beliebigen Namen.
- *Hide*  
Blendet alle Meldungen eines gewählten Typs aus.
- *Promote to warning*  
Stuft eine Information hoch zur Warnung.
- *Promote to error*  
Stuft eine Information oder Warnung hoch zur Fehlermeldung.
- *Revoke Promotion*  
Widerruft eine Hochstufung und stellt den ursprünglichen Typ wieder her.
- *Settings*  
Öffnet das Fenster „CodeGen Message Configuration“ (siehe Seite 163).



## 2.4.1 Register „Monitor“

Im Register „Monitor“ werden die Ergebnisse verschiedener Operationen angezeigt. In der nachstehenden Abbildung werden beispielsweise die Ergebnisse der Codegenerierung und -compilierung eines Zustandsautomaten angezeigt. Ältere Log-Texte werden nicht überschrieben, die neuen Log-Texte werden am Ende eingefügt.

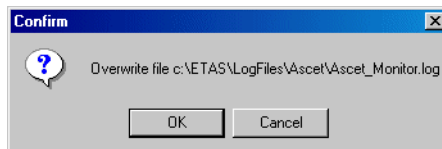


### Inhalt des Registers „Monitor“ speichern:

Um den Inhalt des Registers „Monitor“ zu speichern, gehen Sie wie folgt vor.

- Wählen Sie **File** → **Save**
- oder
- drücken Sie **<CTRL> + <S>**, um den Text im Verzeichnis `ETAS\LogFiles\Ascet` in der Datei `Ascet_Monitor.log` zu speichern.

Wenn diese Datei existiert, werden Sie aufgefordert, das Überschreiben zu bestätigen.



- Bestätigen Sie mit **OK**.
- Der Inhalt des Registers „Monitor“ wird gespeichert.

## Inhalt des Registers „Monitor“ unter beliebigem Namen speichern:

---

- Wählen Sie **File** → **Save As**, um die Log-Datei unter einem beliebigen Pfad und Namen zu speichern.  
Ein Dateiauswahlfenster öffnet sich.
- Wählen Sie einen Pfad und einen Dateinamen und klicken Sie auf **Speichern**.  
Der Inhalt des Registers „Monitor“ wird in die entsprechende Datei geschrieben.

Wenn Sie nicht den ganzen Text im Register „Monitor“ speichern wollen, können Sie im Monitorfenster wie in einem beliebigen Texteditor Abschnitte löschen, kopieren und einfügen. Dabei stehen Ihnen im Menü **Edit** die Funktionen **Cut**, **Copy** und **Paste** zur Verfügung. Ebenfalls können Sie mit der Menüfunktion **Edit** → **Find/Replace** Textteile suchen und ersetzen.

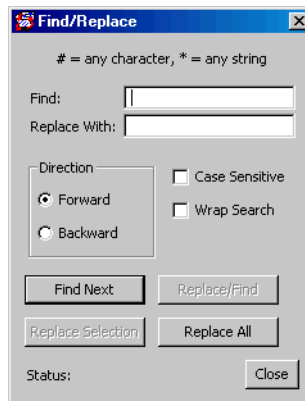
### Hinweis

*Denken Sie beim Bearbeiten daran, dass es kein Rückgängigmachen gibt.*

## Suchen/Ersetzen im Register „Monitor“:

---

- Wählen Sie **Edit** → **Find/Replace**  
oder
- drücken Sie <CTRL> + <F>.  
Das Fenster „Find/Replace“ öffnet sich.



- Geben Sie im Feld „Find“ den Suchtext ein.

- Geben Sie im Feld „Replace“ den Text ein, der den gesuchten Text ersetzen soll.
- Wählen Sie unter „Direction“ die Suchrichtung.
- Aktivieren Sie die Option **Case Sensitive**, wenn die Suche Groß-/Kleinschreibung berücksichtigen soll.
- Aktivieren Sie die Option **Wrap Search**, wenn die Suche nach Erreichen des Endes der Suchrichtung am Anfang fortgesetzt werden soll.
- Klicken Sie auf **Find Next**, um das nächste Vorkommen des Suchtextes zu finden.
- Klicken Sie auf **Replace Selection**, um das ausgewählte Vorkommen zu ersetzen.
- Klicken Sie auf **Replace/Find**, um das ausgewählte Vorkommen zu ersetzen und das nächste Vorkommen zu suchen.
- Klicken Sie auf **Replace All**, um alle Vorkommen des Suchtextes zu ersetzen.
- Schließen Sie das Fenster „Find/Replace“ mit **Close**.

Wenn Sie den ganzen Inhalt des Registers löschen wollen, z.B. vor einer Operation, deren Ergebnis Sie speichern wollen, gehen Sie wie folgt vor:

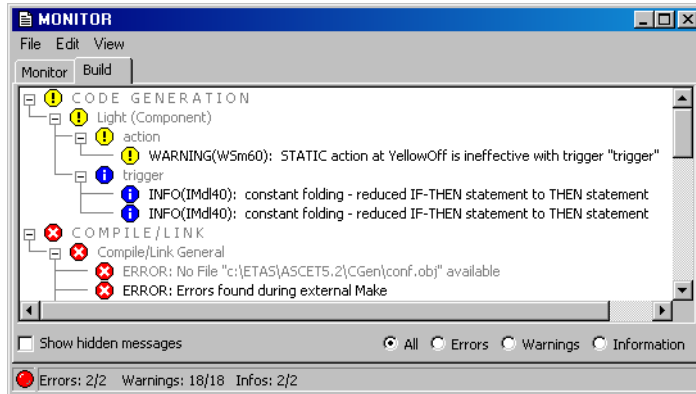
#### **Löschen des Registerinhalts:**

---

- Wählen Sie **Edit** → **Clear**  
*oder*
- drücken Sie auf <CTRL> + <R>.  
Der gesamte Text im Register „Monitor“ wird gelöscht.

## 2.4.2 Register „Build“

Im Register „Build“ werden die Ergebnisse der Codegenerierung und -compilierung sowie der Analyse von Diagrammen angezeigt. Anders als im Register „Monitor“ werden hier jedoch nur die Ergebnisse der letzten Operation angezeigt.



Angezeigt werden drei Arten von Meldungen, die bei den genannten Operationen ausgegeben werden: Fehlermeldungen (roter Punkt), Warnungen (gelber Punkt) und Informationen (blauer Punkt). Die verschiedenen Typen der Meldungen sind durch Kürzel, z.B. WSm60 oder IMd140, gekennzeichnet. Die Anzeige für jede Operation ist in einer Baumstruktur nach Komponenten und Methoden/Prozessen geordnet.

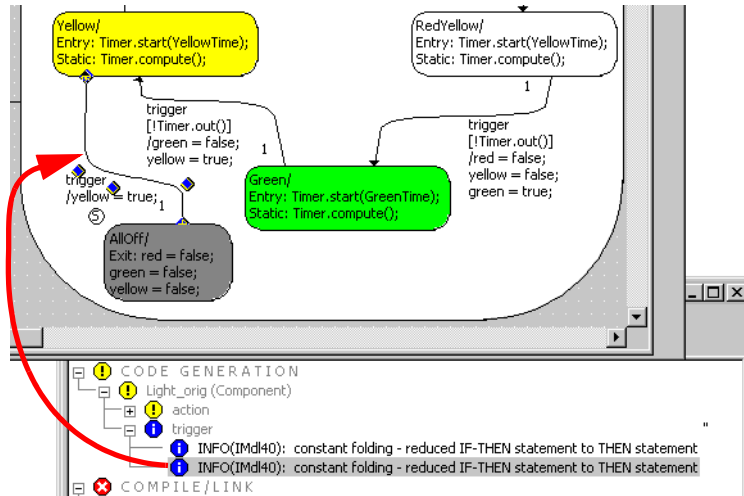
Für eine einfache Lokalisierung der Problemstellen sind die angezeigten Meldungen gleichzeitig Links zu den betroffenen Stellen.

### Ursache der Meldung anzeigen:

- Doppelklicken Sie auf eine Meldung im Register „Build“  
*oder*

- klicken Sie mit der rechten Maustaste auf eine Meldung und wählen Sie **Open** aus dem Kontextmenü.

Der Editor für die betroffene Komponente wird geöffnet und die betroffene Stelle markiert.



Im abgebildeten Beispiel hat der Übergang vom Zustand `AllOff` in den Zustand `yellow` die Meldung ausgelöst.

Für Compiler- oder Linkerfehler wird ein Textfenster oder ein Texteditor mit der betroffenen generierten Datei geöffnet. Die Verwendung eines Texteditors wird im Knoten „External Tools“ des ASCET-Optionsfensters gesteuert (s. Seite 61).

### Inhalt des Registers „Build“ speichern:

Wenn Sie den Inhalt des Registers „Build“ speichern wollen, gehen Sie wie folgt vor.

- Wählen Sie im Register „Build“ **File** → **Save** oder
- drücken Sie `<CTRL> + <S>`, um den Log-Text im Verzeichnis `ETAS\LogFiles\Ascet` in der Datei `CodeGeneration.log` zu speichern.

Anders als beim Register „Monitor“ wird hier eine bereits vorhandene Datei dieses Namens *ohne Warnung* überschrieben.

Der im Register „Build“ angezeigte Inhalt wird in die entsprechende Datei geschrieben.

#### **Hinweis**

Ausgeblendete Meldungen (s. Seite 159) werden **nicht** gespeichert.

### **Inhalt des Registers „Build“ unter beliebigem Namen speichern:**

---

- Wählen Sie im Register „Build“ aus dem Kontextmenü **File Out**

oder

- wählen Sie die Menüfunktion **File → Save As**

oder

- klicken Sie mit der rechten Maustaste in das Register und wählen Sie **File Out** aus dem Kontextmenü, um den Inhalt des Registers „Build“ unter einem beliebigen Pfad und Namen zu speichern.

Ein Dateiauswahlfenster öffnet sich.

- Wählen Sie im Kombikästchen "Dateiname" das Format **XML log file (\*.xml)** oder **ASCII Log file (\*.log;\*.txt)**.
- Wählen Sie einen Pfad und einen Dateinamen und klicken Sie auf **Speichern**.

#### **Hinweis**

Wenn Sie das ASCII-Format gewählt haben, müssen Sie beim Dateinamen auch die Datei-Erweiterung angeben. Diese wird nicht automatisch gesetzt.

Der im Register „Build“ angezeigte Inhalt wird in die entsprechende Datei geschrieben.

## Meldungen im Register „Build“ konfigurieren

Sie haben verschiedene Möglichkeiten, die Anzeige von Meldungen aus dem Register „Build“ heraus zu konfigurieren. Weitere Konfigurationsmöglichkeiten sind im Abschnitt „Meldungen im Fenster „CodeGen Message Configuration“ konfigurieren“ auf Seite 162 beschrieben. Ihre Konfigurationen werden in den Optionen des aktuellen Projekts (für die Arbeit mit Komponenten: des Standardprojekts) gespeichert und sind von dort aus zugänglich (s. Seite 445).

**Ein-/Ausblenden von Meldungen:** Sie können Meldungen gleichen Typs ausblenden, um das Fenster übersichtlicher zu gestalten oder gezielt Meldungen desselben Typs zu analysieren.

### Hinweis

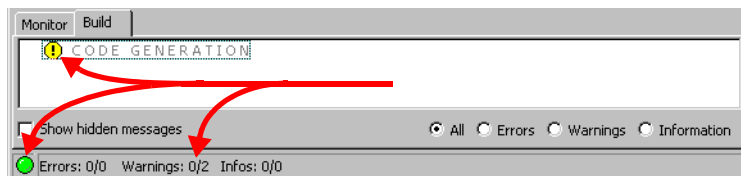
*Fehlermeldungen (auch wenn es sich um hochgestufte Informationen oder Warnungen handelt) können nicht ausgeblendet werden.*

### Meldungen ausblenden:

- Klicken Sie im Register „Build“ mit der rechten Maustaste auf eine Meldung des Typs, den Sie ausblenden wollen.
- Wählen Sie aus dem Kontextmenü **Hide**.

Alle Meldungen des gewählten Typs werden ausgeblendet. Die Nummer der angezeigten Meldungen in der Fußzeile des Monitorfensters wird angepasst.

Wenn Sie alle Fehler- und Warnmeldungen ausblenden, erscheint links in der Fußzeile ein grüner Punkt. Dass ausgeblendete Meldungen vorhanden sind, zeigen neben der Anzahl auch die farbigen Symbole im Anzeigefeld.



## Alle ausgeblendeten Meldungen einblenden:

---

Sie können alle ausgeblendeten Meldungen wieder anzeigen.

- Aktivieren Sie im Register „Build“ die Option **Show hidden messages**.

Sämtliche ausgeblendeten Meldungen werden angezeigt.

### Hinweis

*Eine Option, um ausgeblendete Meldungen eines einzelnen Typs wieder einzublenden, gibt es **nicht**. Das geht nur im Fenster „CodeGen Message Configuration“ (s. Seite 162).*

**Hochstufen von Meldungen:** Informationen und Warnungen unterbrechen den Codegenerierungsprozess nicht. Der jeweilige Kontext eines Projekts kann für manche Informationen oder Warnungen eine strengere Interpretation erforderlich machen, daher gibt es die Möglichkeit, Informationen oder Warnungen eines Typs hochzustufen. Die nachstehende Tabelle enthält die möglichen Hochstufungen.

↓ hochstufen zu →	Information	Warnung	Fehlermeldung
<b>Information</b>		möglich	möglich
<b>Warnung</b>	---		möglich
<b>Fehlermeldung</b>	---	---	

Der Typ einer hochgestuften Meldung bleibt erhalten, daher hat das Hochstufen keinen Einfluss auf das Ein-/Ausblenden von Meldungen.

Wird eine ausgeblendete Meldung hochgestuft, wird sie wieder eingeblendet.

## Informationen und Warnungen hochstufen:

---

- Klicken Sie im Register „Build“ mit der rechten Maustaste auf eine Meldung des Typs, den Sie hochstufen wollen.
- Wählen Sie aus dem Kontextmenü **Promote to warning**

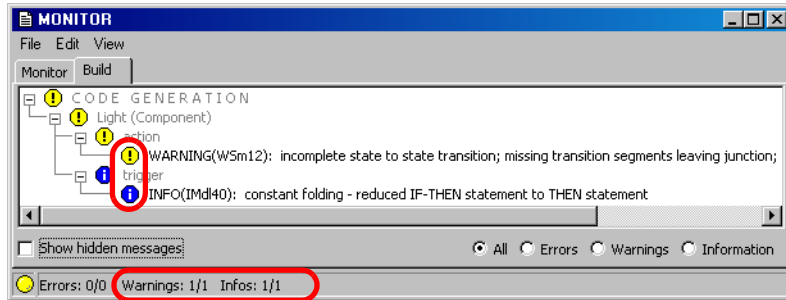
oder




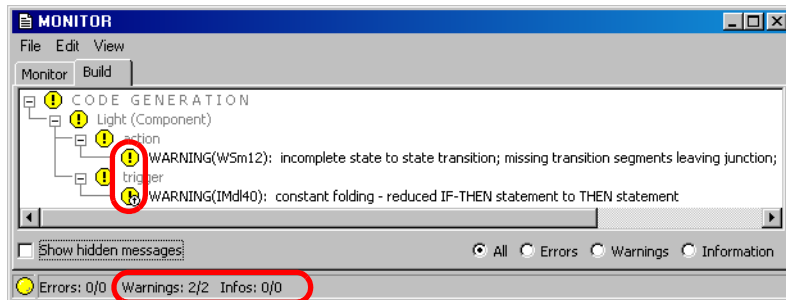
- wählen Sie aus dem Kontextmenü **Promote to error**.


Die Meldungen der gewählten Typen werden ab der nächsten Codegenerierung als Warnungen bzw. Fehlermeldungen gezählt und im Register „Build“ entsprechend dargestellt.

Die folgenden Abbildungen sollen das Hochstufen verdeutlichen. Die erste Abbildung zeigt eine Warnung und eine Information, wie sie von der Codegenerierung ausgegeben werden.

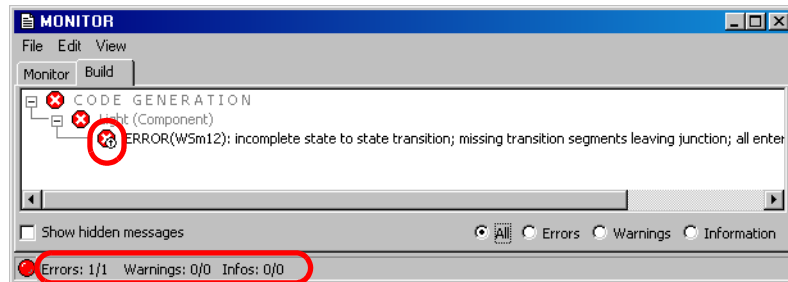


Für die zweite Abbildung wurde die Information zur Warnung hochgestuft. Der Typ bleibt derselbe – IMd140 –, aber die Meldung bekommt das Symbol für Warnungen mit einer Kennzeichnung für hochgestufte Informationen: . Im Register „Build“ werden nun zwei Warnungen angezeigt und gezählt.



Für die dritte Abbildung wurde die ursprüngliche Warnung zur Fehlermeldung hochgestuft. Der Typ bleibt auch hier derselbe – WSm12 –, aber die Meldung bekommt das Symbol für Fehlermeldungen mit einer Kennzeichnung für hochgestufte Meldungen: .

Durch die Hochstufung zur Fehlermeldung bricht die Codegenerierung an dieser Stelle ab, sie erreicht die Stelle der zweiten Meldung nicht. Daher wird in diesem Fall im Register „Build“ nur eine Fehlermeldung angezeigt.



### Hochstufung entfernen:

Wenn eine hochgestufte Meldung im Register „Build“ angezeigt wird, können Sie die Hochstufung entfernen.

- Klicken Sie im Register „Build“ mit der rechten Maustaste auf eine hochgestufte Meldung.
- Wählen Sie aus dem Kontextmenü **Revoke Promotion**.

Bei der nächsten Codegenerierung werden Meldungen dieses Typs wieder in ihrer ursprünglichen Art angezeigt.

### Meldungen im Fenster „CodeGen Message Configuration“ konfigurieren

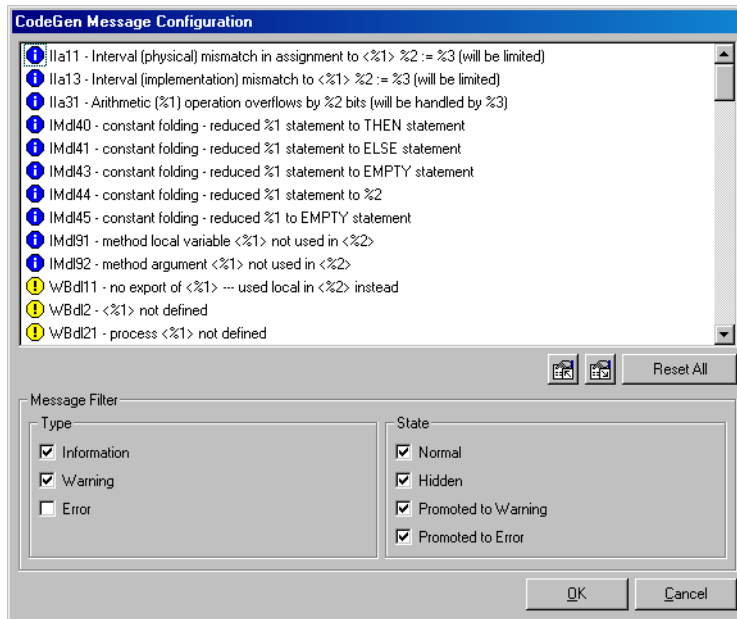
Weitere Konfigurationsmöglichkeiten haben Sie im Fenster „CodeGen Message Configuration“. Dort können Sie nicht nur die aktuell angezeigten, sondern alle vorhandenen Meldungen hochstufen oder verstecken, Sie können Hochstufungen entfernen und versteckte Meldungen eines Typs wieder anzeigen.

### Fenster „CodeGen Message Configuration“ öffnen:

Aus dem Monitorfenster können Sie das Fenster nur öffnen, wenn im Register „Build“ mindestens eine Meldung angezeigt wird. Wie Sie über die Projektoptionen zum Fenster „CodeGen Message Configuration“ gelangen, ist im Abschnitt „Knoten „Build““ auf Seite 443 beschrieben.

- Klicken Sie mit der rechten Maustaste in das Register „Build“.

- Wählen Sie aus dem Kontextmenü **Settings**.  
Das Fenster „CodeGen Message Configuration“ öffnet sich.



Das Fenster enthält folgende Elemente:

- Listenfeld  
Dieses Feld kann alle Informationen, Warnungen und Fehlermeldungen anzeigen. Sie steuern die Anzeige über die Optionen im Feld „Message Filter“. Die Liste ist alphabetisch nach den Typen sortiert.
- Schaltflächen **Import Message Configuration from XML File** und **Export Message Configuration to XML File**  
Starten Import und Export von Meldungskonfigurationen.
- Schaltfläche **Reset All**  
Versetzt alle Meldungen in den vom System ursprünglich vorgegebenen Zustand.
- Feld „Message Filter“  
Sie verwenden die Optionen in diesem Feld, um die Anzeige im Listenfeld einzurichten.

- Feld „Type“ mit Optionen **Information**, **Warning** und **Error**  
Diese Optionen schalten die Anzeige der jeweiligen Meldungsart global ein/aus. Die folgenden Optionen legen die Details fest.
- Option **Normal**  
Schaltet die Anzeige unbearbeiteter Meldungen (nicht versteckt, nicht hochgestuft) der gewählten Arten ein/aus.
- Option **Hidden**  
Schaltet die Anzeige versteckter Meldungen der gewählten Arten ein/aus.
- Optionen **Promoted to Warning** und **Promoted to Error**  
Schaltet die Anzeige der zu Warnungen bzw. Fehlern hochgestuften Meldungen der gewählten Arten ein/aus.

### **Anzeige einrichten:**

---

- Aktivieren Sie im Fenster „CodeGen Message Configuration“ die Optionen der Meldungsarten, die Sie im Listenfeld anzeigen wollen.

Zur Verfügung stehen **Information**, **Warning** und **Error**.

- Verfeinern Sie die Anzeige mit den Optionen **Normal**, **Hidden** und **Promoted to Warning/Error**.

Sie können so gezielt nach Meldungsgruppen (nur unbearbeitete Informationen, keine verborgenen Meldungen, hochgestufte Warnungen usw.), suchen. Die nachfolgende Einstellung sorgt z.B. dafür, daß nur ausgeblendete Meldungen angezeigt werden.


Type	State
<input checked="" type="checkbox"/> Information	<input type="checkbox"/> Normal
<input checked="" type="checkbox"/> Warning	<input checked="" type="checkbox"/> Hidden
<input type="checkbox"/> Error	<input type="checkbox"/> Promoted to Warning
	<input type="checkbox"/> Promoted to Error

## Meldungen im Monitorfenster ausblenden:

---

### Hinweis

*Fehlermeldungen (auch wenn es sich um hochgestufte Informationen oder Warnungen handelt) können nicht ausgeblendet werden.*

- Wählen Sie im Fenster „CodeGen Message Configuration“ eine oder mehrere Meldungen, die Sie ausblenden wollen.
- Wählen Sie aus dem Kontextmenü **Hide**.  
Im Listenfeld werden die hochgestuften Meldungen durch blässere Symbole markiert, z.B. diese: 
- Schließen Sie das Fenster „CodeGen Message Configuration“ mit **OK**.

Im Register „Build“ des Monitorfensters werden alle Meldungen der gewählten Typen ausgeblendet.

Die Nummer der angezeigten Meldungen in der Fußzeile des Monitorfensters wird angepasst.

Wenn Sie alle Fehler- und Warnmeldungen ausblenden, erscheint links in der Fußzeile ein grüner Punkt. Dass ausgeblendete Meldungen vorhanden sind, zeigen neben der Anzahl auch die farbigen Symbole im Anzeigefeld.

## Ausgeblendete Meldungen einblenden:

---

- Wählen Sie im Fenster „CodeGen Message Configuration“ eine oder mehrere ausgeblendete Meldungen, die Sie einblenden wollen.  
Zum leichteren Auffinden ausgeblendeter Meldungen kann es hilfreich sein, die Optionen **Normal** und **Promoted to \*** zu deaktivieren.
- Wählen Sie aus dem Kontextmenü **Show**.

- Schließen Sie das Fenster „CodeGen Message Configuration“ mit **OK**.

Im Register „Build“ des Monitorfensters werden alle Meldungen der gewählten Typen einblendet.

### Informationen und Warnungen hochstufen:

---

- Wählen Sie im Fenster „CodeGen Message Configuration“ eine oder mehrere Meldungen, die Sie hochstufen wollen.
- Wählen Sie aus dem Kontextmenü **Promote to warning**

oder

- wählen Sie aus dem Kontextmenü **Promote to error**.
- Schließen Sie das Fenster „CodeGen Message Configuration“ mit **OK**.

Die Meldungen der gewählten Typen werden ab der nächsten Codegenerierung als Warnungen bzw. Fehlermeldungen gezählt und im Register „Build“ entsprechend dargestellt. Im Listenfeld werden die hochgestuften Meldungen durch spezielle Symbole markiert:



Information hochgestuft zur Warnung



Information hochgestuft zur Fehlermeldung



Warnung hochgestuft zur Fehlermeldung

### Hochstufung entfernen:

---

- Wählen Sie im Fenster „CodeGen Message Configuration“ eine oder mehrere Meldungen, deren Hochstufung Sie entfernen wollen.
- Wählen Sie aus dem Kontextmenü **Revoke Promotion**.

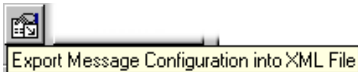
- Schließen Sie das Fenster „CodeGen Message Configuration“ mit **OK**.

Meldungen dieser Typen werden bei der nächsten Codegenerierung im Register „Build“ wieder in ihrer ursprünglichen Art angezeigt.

Sie können Ihre Konfiguration aus versteckten und hochgestuften Meldungen in eine XML-Datei exportieren, und sie können eine solche Konfiguration aus einer XML-Datei importieren.

### Einstellungen exportieren:

---



- Öffnen Sie das Fenster „CodeGen Message Configuration“.
- Klicken Sie auf die Schaltfläche **Export Options of selected Node into XML File**.

Ein Bestätigungsfenster öffnet sich.

- Wenn das Bestätigungsfenster in Zukunft nicht mehr angezeigt werden soll, deaktivieren Sie die Option **Show next time**. (Siehe auch Seite 47.)
- Bestätigen Sie das Speichern Ihrer Änderungen mit **OK**.

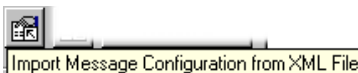
Der Windows-Dateiauswahldialog öffnet sich. Das Format \*.xml ist fest eingestellt.

- Setzen Sie Pfad und Namen der Exportdatei.
- Klicken Sie auf **Speichern**.

Die versteckten und hochgestuften Meldungen werden in die angegebene XML-Datei geschrieben.

### Einstellungen importieren:

---



- Öffnen Sie das Fenster „CodeGen Message Configuration“.
- Klicken Sie auf die Schaltfläche **Import Message Configuration from XML File**.

Der Windows-Dateiauswahldialog öffnet sich. Das Format \*.xml ist fest eingestellt.

- Wählen Sie die XML-Datei, die die gewünschte Konfiguration enthält.
- Klicken Sie auf **Öffnen**.

Die in der angegebenen XML-Datei enthaltenen versteckten und hochgestuften Meldungen werden importiert und entsprechend im Fenster „CodeGen Message Configuration“ und im Monitorfenster angezeigt.



### 3 Einfügen benutzerdefinierter Funktionen

Für ein einfacheres Arbeiten mit Zusätzen zu ASCET (z. B. Versionsmanagement) und externen Programmen gibt es die Möglichkeit, eigene Funktionalitäten zu definieren. Das sind

- eigene Menüs in verschiedenen Fenstern,
- eine Autostartaktion (ausgeführt beim Start von ASCET),
- eine Abschlussaktion (ausgeführt beim Beenden von ASCET).

Diese werden in `*.ini`-Dateien spezifiziert, die in einem Unterverzeichnis der ASCET-Installation, `ETAS\Ascet5.2\Executer`, oder des Datenverzeichnisses, `ETASData\Ascet5.2\Executer`, abgelegt werden. Das Unterverzeichnis `Executer` kann auch weitere Unterverzeichnisse enthalten.

#### Hinweis

*Um den Inhalt des Unterverzeichnisses `Executer` bei einem Update oder einer Neuinstallation von ASCET zu erhalten, platzieren Sie es unterhalb des Datenverzeichnisses (`ETASData\Ascet5.2`).*

Die `*.ini`-Dateien werden nur einmal beim Start von ASCET gelesen; spätere Änderungen werden erst nach einem Neustart wirksam. Die Syntax dieser Dateien entspricht dem Windows `*.ini`-Format.

Jede `*.ini`-Datei besteht aus einem oder mehreren Abschnitten, deren Namen in eckige Klammern gesetzt sind. Folgende Abschnitte stehen zur Verfügung:

- **[ <NAME> ]** — definiert eine neue Menüfunktion. Der Name kann beliebig gewählt werden, allerdings darf in einer `*.ini`-Datei jeder Name nur einmal vorkommen. Siehe auch Kapitel 3.1 „Menüfunktionen definieren“.
- **[ AUTOSTART ]** — definiert die Autostartaktion, siehe Kapitel 3.2 „Autostartaktion definieren“.
- **[ SHUTDOWN ]** — definiert die Abschlussaktion, siehe Kapitel 3.3 „Abschlussaktion definieren“.

## 3.1 Menüfunktionen definieren

---

Der Abschnitt der \*.ini-Datei, der eine Menüfunktion definiert, sieht wie folgt aus:

```
[ <Name> ]  
WINDOW=<window name>  
MENU=~<menu name>  
ITEM=~<menu item name>  
DESCRIPTION="<text>"  
SEPARATOR=<separator>  
FILE=<filename>.txt
```

### Hinweis

Die Begriffe links von den Gleichheitszeichen **müssen** in Grossbuchstaben geschrieben werden.

Die Variablen haben folgende Bedeutung:

- <Name> ist der Name des Abschnitts. Der Name muss innerhalb der \*.ini-Datei eindeutig sein. Dabei wird *nicht* zwischen Groß- und Kleinschreibung unterschieden, [ Input ] und [ INPUT ] z. B. sind gleich.  
Wert: beliebig, aber eindeutig
- <window name> ist der Name des Fensters, in dem das Menü erzeugt wird.  
Wert: siehe Tab. 3-1 auf Seite 171
- <menu name> ist der Name des Menüs, das im angegebenen Fenster erzeugt wird.  
Wert: beliebig
- <menu item name> ist der Name der Menüfunktion, die im Menü <menu name> erzeugt wird.  
Wert: beliebig
- <text> ist eine Beschreibung der Menüfunktion, die nur für Dokumentationszwecke benötigt wird.  
Wert: beliebig

- `<separator>` gibt an, ob eine Trennlinie vor oder nach der Menüfunktion erzeugt wird. Soll keine Trennlinie gezeichnet werden (Nothing), kann die Zeile ganz weggelassen werden.  
Wert: Before / After / Nothing
- `<filename>` – die Skriptdatei `<filename>.txt` wird ausgeführt, wenn die Menüfunktion ausgewählt wird. Die Struktur dieser Dateien ist in Kapitel 3.4 „Struktur der Skriptdateien“ erklärt.  
Wert: beliebig

Name des Fensters	ASCET Fenster
mainwindow oder databasebrowser	Komponentenmanager
projecteditor	Projekteditor
blockdiagrameditor	Blockdiagrammeditor
smeditor	Zustandsautomaten-Editor
esdleditor	ESDL-Editor
ccodeeditor	C-Code-Editor
conttimeblockdiagrameditor	Blockdiagrammeditor (CT-Blöcke)
conttimeesdleditor	ESDL-Editor (CT-Blöcke)
conttimeccodeeditor	C-Code-Editor (CT-Blöcke)
booleantableeditor	Boolescher Tabelleneditor
conditionaltableeditor	Editor für Bedingunstabellen
datadialog	Fenster „Data for ...“
impldialog	Implementierungseditor
onlinesimulation	Online-Experimentierumgebung
offlinesimulation	Offline-Experimentierumgebung
intecriobackanimation	Experimentierumgebung für Back-Animation mit INTECRIO
incabackanimation	Experimentierumgebung für Back-Animation mit INCA

**Tab. 3-1** Namen der Fenster, in die Menüfunktionen eingefügt werden können

Werden mehrere \*.ini-Dateien angelegt, werden diese in alphabetischer Reihenfolge ausgewertet. Die Menüfunktionen, die in einer Datei `asd_menu.ini` definiert werden, erscheinen im entsprechenden Menü weiter oben als die Funktionen desselben Menüs, die in der Datei `my_menu.ini`

definiert werden.

Innerhalb derselben \*.ini-Datei werden die Menüfunktionen in der Reihenfolge angelegt, in der sie definiert werden.

### Menüfunktion definieren:

---

- Erzeugen Sie eine neue \*.ini-Datei  
*oder*
- öffnen Sie eine vorhandene \*.ini-Datei.
- Geben Sie einen Namen für den Abschnitt ein, z.B.

[ Input ].

Achten Sie darauf, dass der Name in der Datei nur einmal verwendet wird.

- Geben Sie das Fenster an, in dem das Menü erzeugt werden soll, z.B.

WINDOW=databasebrowser.

- Geben Sie den Namen ein, den Ihr Menü haben soll, z. B.

MENU=~My Menu.

- Geben Sie den Namen ein, den die Menüfunktion haben soll, z. B.

ITEM=~Show OID.

- Geben Sie eine Beschreibung ein, z. B.

DESCRIPTION=show object IDs.

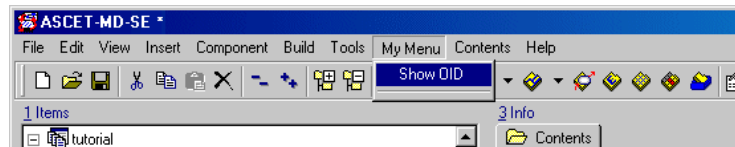
- Geben Sie ein, wo Sie eine Trennlinie haben wollen, z.B.

SEPARATOR=After.

- Geben Sie die Skriptdatei an, z. B.

FILE=show.txt.

Beim nächsten Start von ASCET wird im Komponentenmanager das Menü **My Menu** erzeugt.



Wenn Sie mehrere Menüfunktionen im selben oder in verschiedenen ASCET-Fenstern definieren wollen, müssen Sie für jede Menüfunktion einen solchen Abschnitt anlegen. Konstruktionen wie die folgende funktionieren *nicht*, die zweite Definition wird ignoriert.

```
[ EXPORT ]
WINDOW=databasebrowser
MENU=~My Menu
ITEM=~Show OID
DESCRIPTION=show object IDs
FILE=show.txt
ITEM=~open editor
DESCRIPTION=opens external text editor
FILE=openeditor.txt
```

Die Abschnitte können wahlweise in derselben oder in separaten \*.ini-Dateien stehen. Separate Dateien bieten den Vorteil, dass Sie über deren Namen die Reihenfolge der Menüfunktionen eines Menüs leichter kontrollieren können.

Wenn Sie eine \*.ini-Datei für mehrere Funktionen verwenden, achten Sie darauf, dass jeder Name nur einmal verwendet wird. Taucht ein Name ein zweites Mal auf, wird die erste Definition ein zweites Mal ausgeführt, unabhängig davon, wie die zweite Definition lautet.

**Ein Beispiel:** Mit den folgenden Abschnitten soll im Projekteditor die Menüfunktion **My Menu** → **Show OID** und im Komponentenmanager die Menüfunktion **My Menu** → **Open Editor** erzeugt werden.

```
[ Input ]
WINDOW=projecteditor
MENU=~My Menu
ITEM=~Show OID
DESCRIPTION=show object IDs
SEPARATOR=After
FILE=show.txt
```

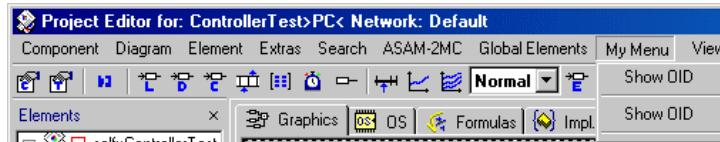
```
[ INPUT ]
WINDOW=databasebrowser
MENU=~My Menu
```

```

ITEM=-Open Editor
DESCRIPTION=open external text editor
SEPARATOR=After
FILE=openeditor.txt

```

Da aber nicht zwischen Groß- und Kleinschreibung unterschieden wird, sind die Namen der Abschnitte identisch, und im Projekteditor wird unter **My Menu** zweimal die Menüfunktion **Show OID** angelegt.



## 3.2 Autostartaktion definieren

Eine Autostartaktion wird in der \*.ini-Datei im Abschnitt [AUTOSTART] definiert. Der Abschnitt kann an beliebiger Stelle in der \*.ini-Datei stehen, darf aber *nur ein einziges Mal* auftauchen, auch wenn mehrere \*.ini-Dateien verwendet werden. Er enthält nur eine Zeile `FILE=<filename>`.

Beispiel:

```

[AUTOSTART]
FILE=start.txt

```

Beim Start von ASCET wird die Skriptdatei `start.txt` ausgeführt.

Die Autostartfunktion wird verwendet, um Konfigurationen für Dateioperationen (z.B. Templates für die Codegenerierung anpassen, Verzeichnisse löschen) zeitgleich mit dem Start von ASCET durchzuführen. Damit ist eine eindeutige reproduzierbare Situation gegeben. Des weiteren kann ein Editor oder anderes Werkzeug aufgerufen werden, um für die Sitzung notwendige Vorarbeiten und Einstellungen durchzuführen. Eine Nachricht (s. Seite 177) kann über die vorgenommenen Einstellungen informieren.

Ein Beispiel für eine solche Skriptdatei:

```

EXECUTE c:\Programme\TextPad 4\TextPad.exe
        C:\ETAS\Ascet5.2\target\c16x\codegen.ini
OBJECT message: "codegen.ini is updated"

```

Auf die COM-API können Sie mit der Autostartaktion nicht zugreifen, da der Automation Server zu diesem Zeitpunkt noch nicht gestartet ist.

### 3.3 Abschlussaktion definieren

---

Eine Abschlussaktion wird in der \*.ini-Datei im Abschnitt [SHUTDOWN] definiert. Der Abschnitt kann an beliebiger Stelle in der \*.ini-Datei stehen; er enthält nur eine Zeile `FILE=<filename>`.

Beispiel:

```
[SHUTDOWN]
FILE=shutdown.txt
```

Wenn ASCET beendet wird, wird die Skriptdatei `shutdown.txt` ausgeführt. Auf die COM-API können Sie mit der Abschlussaktion nicht zugreifen, da der Automation Server zu diesem Zeitpunkt schon beendet ist.

### 3.4 Struktur der Skriptdateien

---

Die Skriptdateien (Zeile `File = ...` in der \*.ini-Datei) werden als Textdateien geschrieben. Neun Schlüsselwörter dienen der Spezifikation verschiedenster Aktionen:

`EXECUTE`, `NOWAIT`, `OBJECT`, `SELECTOR`, `MENUITEM`, `FILE`, `FORK`, `SEND`, `POST`.

Die Verwendung dieser Schlüsselwörter wird in den folgenden Abschnitten erklärt.

Anders als bei den \*.ini-Dateien werden Änderungen in den Skriptdateien wirksam, sobald die entsprechende Menüfunktion das nächste Mal ausgewählt wird.

#### 3.4.1 EXECUTE

---

Das Schlüsselwort `EXECUTE` wird verwendet, um ein externes Programm oder eine Stapelverarbeitungsdatei zu starten. Solange das externe Programm läuft, wird die Ausführung der Skriptdatei unterbrochen.

##### Beispiele:

- `EXECUTE C:\ETAS\Ascet5.2\import.bat` ruft die Stapelverarbeitungsdatei `import.bat` im Verzeichnis `C:\ETAS\Ascet5.2` auf.
- `EXECUTE explorer C:\ETASData\ASCET5.2` startet den Explorer, der den Inhalt des ASCET Datenverzeichnisses anzeigt.

Die Unterbrechung der Ausführung kann unerwünschte Folgen haben. Wird beispielsweise eine Menüfunktion **My Menu** → **open PSP** hinzugefügt, um das Erstellen von Screenshots zu erleichtern, passiert folgendes:

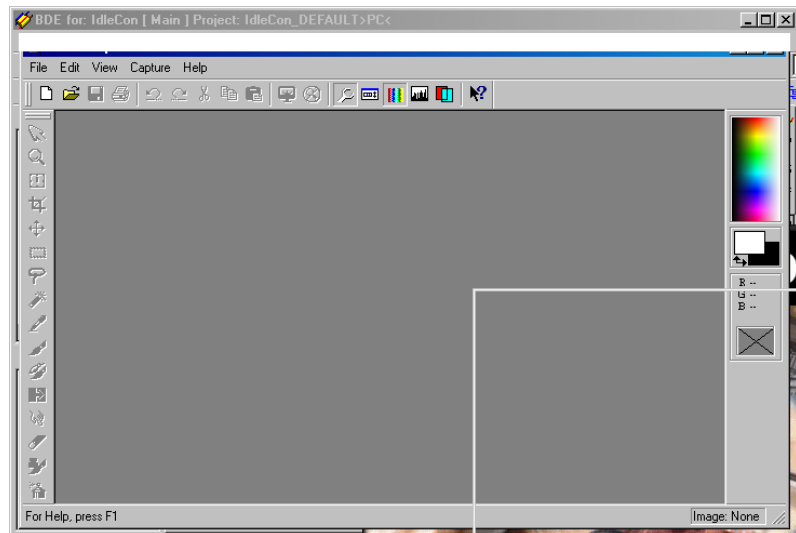
- Die Menüfunktion ruft die Skriptdatei auf.

```
OBJECT popWindow: BDE
```

```
EXECUTE c:\Programme\Paint Shop Pro 5\psp.exe
```

- Die erste Zeile holt den Blockdiagrammeditor nach vorn, die zweite öffnet das Bildverarbeitungsprogramm.

Da die Ausführung der Skriptdatei aber unterbrochen wird, wird der Blockdiagrammeditor nicht aktualisiert, und man erhält in etwa das folgende Bild.



Ein Vertauschen der Zeilen löst das Problem nicht, da in dem Fall der Blockdiagrammeditor erst nach vorn geholt wird, wenn das Grafikprogramm schon wieder geschlossen ist.

### 3.4.2 NOWAIT

Das Schlüsselwort `NOWAIT` wird genau wie `EXECUTE` verwendet. Allerdings wird hier die Ausführung der Skriptdatei nicht unterbrochen.

#### **Beispiel:**

Das Beispiel ist dasselbe wie am Ende des vorigen Abschnitts, nur wird in der Skriptdatei jetzt `NOWAIT` anstelle von `EXECUTE` verwendet.

```
OBJECT popWindow: BDE  
NOWAIT c:\Programme\Paint Shop Pro 5\psp.exe
```



Wieder holt die erste Zeile den Blockdiagrammeditor nach vorn, die zweite öffnet das Bildverarbeitungsprogramm. Da hier die Ausführung der Skriptdatei nicht unterbrochen wird, wird der Blockdiagrammeditor sofort aktualisiert und der Screenshot kann gemacht werden.

### 3.4.3 OBJECT

Das Schlüsselwort `OBJECT` kann zusammen mit einer Reihe von Identifikatoren für verschiedene Kommandos verwendet werden.

- **help: <editor>**

Zeigt die Hilfedatei `exe_hlp.txt` im angegebenen Editor an.

Beispiel: `OBJECT help: notepad` zeigt die Hilfe im Notepad-Editor an.

#### Hinweis

*Das Leerzeichen nach dem Doppelpunkt ist für alle Identifikatoren obligatorisch. Wird es vergessen, kann das Kommando nicht ausgeführt werden.*

- **log: <editor>**

Zeigt die Log-Datei `exe_log.txt` im angegebenen Editor an.

Beispiel: `OBJECT log: notepad`

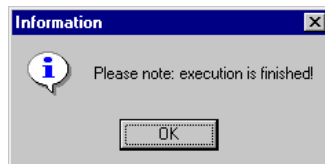
#### Hinweis

*Sollte es bei der Ausführung einer Skriptdatei zu unerwartetem Verhalten kommen, so enthält die Log-Datei Hinweise zu möglichen Fehlerquellen.*

- **message: <text>**

Öffnet ein Fenster mit einer Nachricht für den Benutzer.

Beispiel: `OBJECT message: "Please note: execution is finished!"` erzeugt das abgebildete Fenster.



- **popWindow: <title>**  
Holt das ASCET-Fenster mit dem angegebenen Namen ganz nach vorn auf den Bildschirm. <title> muss nicht der vollständige Name sein, ein Teil des in der Titelzeile angegebenen Namens reicht aus.  
Beispiel: OBJECT popWindow: BDE holt den Blockdiagrammeditor nach vorn.
- **generatePopWindowCommandFile: <filename>**  
Erzeugt unter dem angegebenen Namen eine Datei, in der das korrekte popWindow-Kommando für das derzeit aktive ASCET-Fenster generiert wird. Die gewünschte Dateiendung muss in <filename> mit angegeben werden, Pfadangaben sind optional.  
Beispiel: OBJECT generatePopWindowCommandFile: pop.txt aus dem Komponentenmanager aus aufgerufen erzeugt die Datei pop.txt mit folgendem Inhalt (nur ASCET-MD ist installiert):  

```
OBJECT popWindow: "ASCET-MD"
```
- **wait: <n>**  
Unterbricht die Ausführung der Skriptdatei für n Sekunden.  
Beispiel: Die Zeilen  

```
OBJECT wait: 5  
EXECUTE C:\PVCS\nt\PVCSvmt.exe
```

  
bewirken eine Wartezeit von 5 Sekunden, ehe PVCS gestartet wird.
- **windows: <editor>**  
Zeigt eine Liste (exe\_win.txt) mit den Namen aller geöffneten ASCET-Fenster im angegebenen Editor an.  
Beispiel: OBJECT windows: notepad zeigt die Liste im Notepad-Editor an.

#### 3.4.4 SELECTOR

---

Das Schlüsselwort **SELECTOR** wird verwendet, um im Komponentenmanager oder den Spezifikationseditoren eine Reihe vordefinierter Befehle aufzurufen.

**Beispiel:** Ausgabe der Objekt-IDs von ausgewählten Komponenten

`SELECTOR executerFileOutSelectedElementsTo: "<filename>"` legt eine Datei an, die die Objekt-IDs enthält. Die gewünschte Dateiendung *muss* in `<filename>` angegeben werden, Pfadangaben sind optional.

### Hinweis

*Wenn Sie in <filename> Pfadangaben machen, achten Sie darauf, dass sämtliche Unterverzeichnisse existieren. Andernfalls kann die Datei nicht erzeugt werden.*

Im Komponentenmanager wurde eine Menüfunktion hinzugefügt, die die folgende Skriptdatei aufruft.

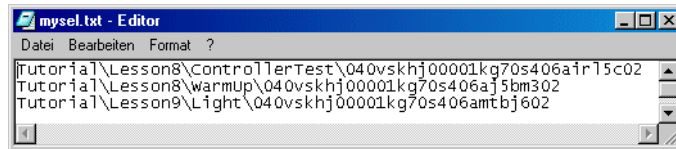
```
SELECTOR executerFileOutSelectedElementsTo:
    ".\Executer\mysel.txt"

EXECUTE Notepad .\Executer\mysel.txt

MENUITEM View>Update
```

Die Komponenten `ControllerTest`, `WarmUp` und `Light` werden ausgewählt, dann wird die Menüfunktion aufgerufen. Folgendes geschieht:

- Die erste Zeile (`SELECTOR . . .`) legt die Datei `mysel.txt` im Unterverzeichnis `Executer` des Installationsverzeichnisses an.
- Die zweite Zeile (`EXECUTE . . .`) öffnet die neu erzeugte Datei im Notepad-Editor.



- Die dritte Zeile (`MENUITEM . . .`) aktualisiert den Komponentenmanager, nachdem Notepad geschlossen wurde.

### Verfügbare Befehle:

- `executerCollapseAll`

Funktion: Klappt die Elementliste im Feld „1 Database“ zu.

Verwendung: im Komponentenmanager

Syntax:

```
SELECTOR executerCollapseAll
```

- `executerExpandCollapseSelectedElement`

Funktion: Klappt das ausgewählte Verzeichnis im Feld „1 Database“ auf und zu. (Unterverzeichnisse werden zwar zugeklappt, aber nicht wieder aufgeklappt.)

Verwendung: im Komponentenmanager

Syntax:

```
SELECTOR executerExpandCollapseSelectedElement
```
- `executerDeselectElements`

Funktion: Hebt die Komponenten- oder Elementauswahl auf.

Verwendung: im Komponentenmanager sowie den Spezifikationseditoren

Syntax:

```
SELECTOR executerDeselectElements
```
- `executerFileOutSelectedDiagrams`

Funktion: Schreibt alle Diagramme und Hierarchien der ausgewählten Komponente sowie der darin aufgenommenen Komponenten als Bild-dateien in das ASCET-Installationsverzeichnis. Dieser Befehl hat auf ESDL- oder C-Code-Komponenten keine Wirkung.

Verwendung: im Komponentenmanager

Syntax:

```
SELECTOR executerFileOutSelectedDiagrams
```
- `executerFileOutSelectedElementsTo:`

Funktion: Schreibt die Objekt-IDs der ausgewählten Elemente in die angegebene Datei.

Verwendung: im Komponentenmanager sowie den Spezifikationseditoren

Syntax<sup>1</sup>:

```
SELECTOR executerFileOutSelectedElementsTo:
    "<filename>"
```

---

<sup>1</sup>. `<filename>` muss die Dateiendung enthalten, Pfadangaben sind optional.

- `executerSelectElementFromString`:

Funktion: Wählt in der Liste „1 Database“ (Komponentenmanager) oder „Elements“ (Spezifikationseditoren) das erste sichtbare Element, dessen Name mit dem Suchstring <name> beginnt. Zum Suchstring passende Elemente in zugeklappten Ordnern oder Komponenten sowie nachfolgende sichtbare passende Elemente werden ignoriert.

Verwendung: im Komponentenmanager sowie den Spezifikationseditoren

Syntax<sup>1</sup>:

```
SELECTOR executerSelectElementFromString:
    "<name>"
```
- `executerFileOutDiagrams`

Funktion: Schreibt alle Diagramme und Hierarchien der editierten Komponente als Bilddateien in das ASCET-Installationsverzeichnis. Dieser Befehl hat auf ESDL- oder C-Code-Komponenten keine Wirkung.

Verwendung: in den Spezifikationseditoren

Syntax:

```
SELECTOR executerFileOutDiagrams
```
- `executerFileOutSelectedDiagramTo`:

Funktion: Schreibt den Pfadnamen des ausgewählten Diagramms der editierten Komponente in die angegebene Datei. Zusätzlich werden für Blockdiagramme im ASCET-Installationsverzeichnis Bilder des gewählten Diagramms und der darin enthaltenen Hierarchien erstellt.

Verwendung: in den Spezifikationseditoren

Syntax<sup>1</sup>:

```
SELECTOR executerFileOutSelectedDiagramTo:
    "<filename>"
```

---

<sup>1</sup>. Groß- und Kleinschreibung wird bei der Suche nicht berücksichtigt.

- executerSelectDiagramFromString:**  
 Funktion: Wählt das Diagramm, das den Suchstring <name> enthält, ohne es jedoch zu laden.  
 Verwendung: in den Spezifikationseditoren  
 Syntax<sup>1</sup>:  

```
SELECTOR executerSelectDiagramFromString:
    "<name>"
```
- executerSelectPage:**  
 Funktion: Öffnet das angegebene Register des Projekteditors.  
 Verwendung: im Projekteditor  
 Syntax:  

```
SELECTOR executerSelectPage: "<tab name>"
```

 (Der Registername muss exakt angegeben werden.)

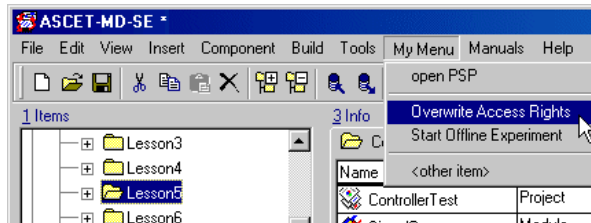
### 3.4.5 MENUITEM

Mit dem Schlüsselwort `MENUITEM` kann eine schon vorhandene Menüfunktion aus dem aktiven Fenster aufgerufen werden.

Beispiel:

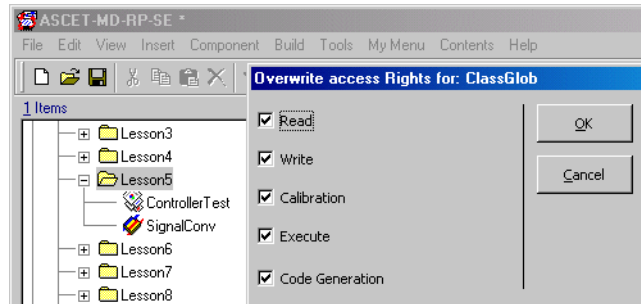
Im Komponentenmanager wurde das Menü **My Menu** hinzugefügt, das über die Menüfunktion **Overwrite Access Rights** die Datei `set_access.txt` mit dem folgendem Inhalt aufruft.

```
MENUITEM Component>Change Access Rights>Overwrite
```



<sup>1</sup>. Groß- und Kleinschreibung wird bei der Suche nicht berücksichtigt.

Wird diese Menüfunktion aufgerufen, so wird für den ausgewählten Datenbankbeitrag das Fenster „Overwrite Access Rights for ...“ geöffnet.



Details über Zugriffsrechte auf Datenbankinträge finden Sie in Kapitel 2.3.7 „Datenbankzugriff“ auf Seite 146.

### 3.4.6 FILE

---

Das Schlüsselwort `FILE` wird verwendet, um aus einer Skriptdatei eine weitere Skriptdatei aufzurufen. Die Ausführung der ersten Skriptdatei wird unterbrochen, solange die zweite ausgeführt wird..

Beispiel:

- `File set_access.txt` ruft aus der aktuellen Skriptdatei heraus die Skriptdatei `set_access.txt` auf.

### 3.4.7 FORK

---

Das Schlüsselwort `FORK` wird genau wie `FILE` verwendet, um eine weitere Skriptdatei aufzurufen. Allerdings wird hier die Ausführung der ersten Skriptdatei nicht unterbrochen.

### 3.4.8 SEND

---

Das Schlüsselwort `SEND` wird verwendet, um einer anderen gestarteten Anwendung eine Nachricht (Windows Message) zu schicken. ASCET wartet, bis die Anwendung die Nachricht verarbeitet hat. Es gibt folgende Möglichkeiten, eine Nachricht zu schicken:

```
SEND <class> <message>
```

```
SEND <class> <window> <message>
```

```
SEND <class> <window> <message> <lparam>
```

```
SEND <class> <window> <message> <lparam> <wparam>
```

<class> ist der Klassenname der Anwendung

**<window>** ist der Fenstertitel der Anwendung

**<message>** ist der Name oder Identifikator der zu sendenden Nachricht

**<lparam>** Erster optionaler Parameter (0 ... 4294967295)

**<wparam>** Zweiter optionaler Parameter (0 ... 4294967295)

Genauere Informationen entnehmen Sie der Microsoft Windows Win32 Schnittstellendokumentation unter dem Stichwort `SendMessage`.

### 3.4.9 POST

---

Das Schlüsselwort `POST` wird genau wie `SEND` verwendet, um einer anderen gestarteten Anwendung eine Nachricht (Windows Message) zu schicken. Allerdings wird hier nicht auf die Verarbeitung der Nachricht gewartet.

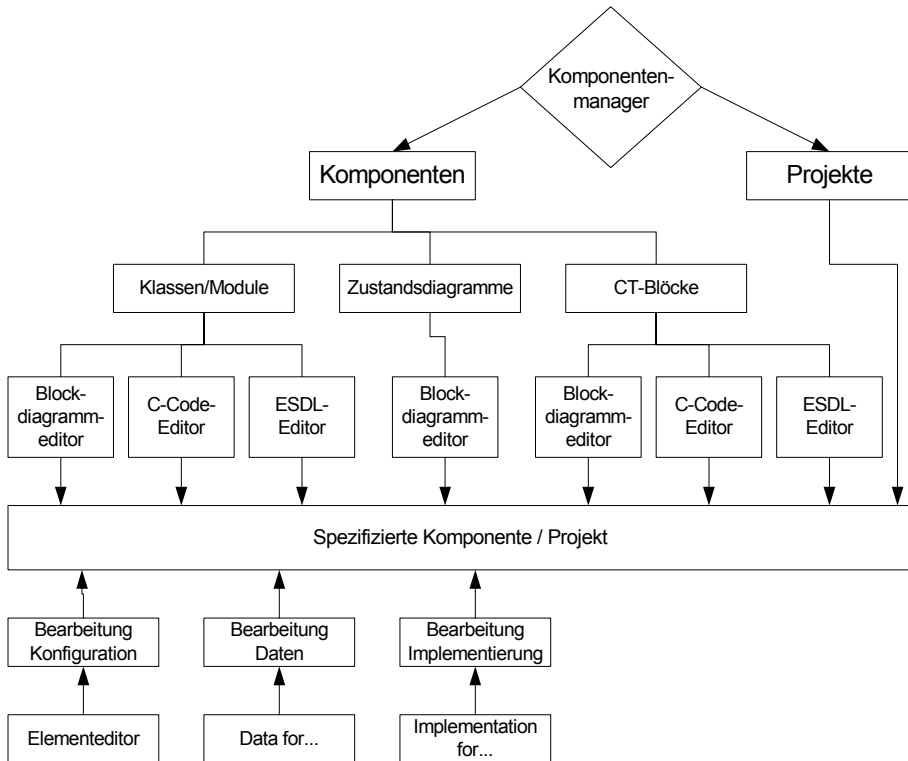
Genauere Informationen entnehmen Sie der Microsoft Windows Win32 Schnittstellendokumentation unter dem Stichwort `PostMessage`.



## 4 Spezifikation von Komponenten und Projekten

In einem Projekt wird das Embedded Control System spezifiziert. Ein Projekt ist wiederum aus verschiedenen Komponenten zusammengesetzt.

In einer Komponente werden der Inhalt (Vereinbarung von Variablen, Parametern), definierte Schnittstellen (Prozesse, Methoden) und ein Algorithmus, der die Verarbeitung vorgibt, spezifiziert.



Die Spezifikation von Komponenten kann grafisch, im ESDL- oder C-Code vorgenommen werden.

Jedes für die Spezifikation verwendete Element hat Standardwerte für die Konfiguration, die Implementierung und die Daten. Diese Werte können über Editoren bearbeitet werden.

Dieses Kapitel beschreibt das Arbeiten mit den verschiedenen Spezifikationseditoren, und wie Sie die Standardwerte der verwendeten Elemente bearbeiten können.

## 4.1 Der Blockdiagrammeditor

---

Im vorliegenden Abschnitt wird die Verwendung des Blockdiagrammeditors ausführlich beschrieben. Dabei steht die Verwendung des Programms im Mittelpunkt und nicht die Entwicklung einer speziellen Funktionalität. Die grafische Sprache von ASCET wird im Abschnitt „Die Modellersprache“ im ASCET-Referenzhandbuch detailliert beschrieben.

Der ASCET Blockdiagrammeditor wird verwendet, um Komponenten grafisch zu spezifizieren. Der Benutzer zeichnet ein Blockdiagramm, das festlegt, was die Komponente macht.

Eine Komponente kann Basiselemente und komplexe Elemente, Operatoren, Kontrollflussanweisungen und Verweise auf andere Komponenten enthalten. Wie Sie eine Komponente anlegen, ist unter „Erstellen eines Datenbankeintrags:“ auf Seite 79 beschrieben.

### Öffnen des Blockdiagrammeditors:

---

- Markieren Sie im Komponentenmanager das gewünschte Objekt.

- Doppelklicken Sie auf das Objekt

*oder*

- wählen Sie **Component** → **Edit Item**

*oder*

- wählen Sie **Edit Item** aus dem Kontextmenü

*oder*

- drücken Sie <ENTER>.

Das gewählte Objekt wird im Blockdiagrammeditor geöffnet.

Wenn Sie eine Komponente öffnen, die über die aktuell gewählte Größe des Zeichenbereichs (siehe „Größe des Zeichenbereichs einstellen:“ auf Seite 189) hinausragt, öffnet sich ein Meldungsfenster, das die Größe des Zeichenbereichs und die Größe des Blockdiagramms nennt und Ihnen die Möglichkeit gibt, die Größe des Zeichenbereichs anzupassen.

## Beenden des Blockdiagrammeditors:

---

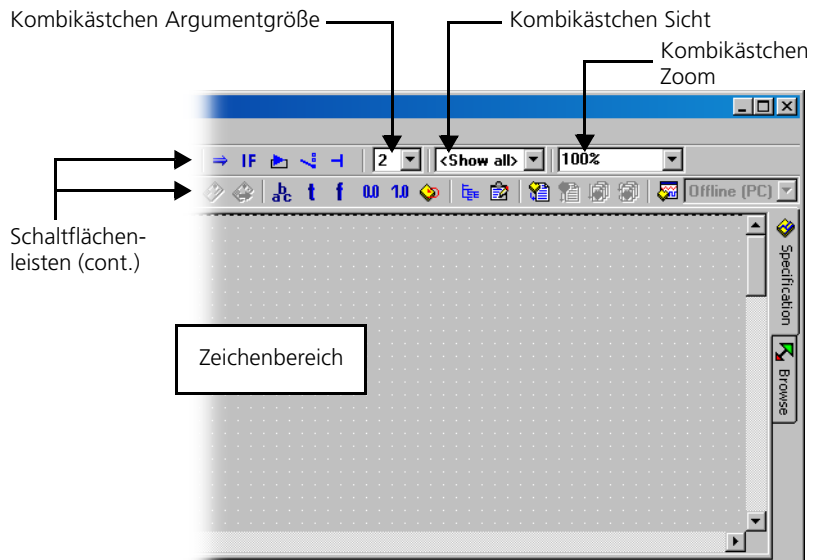
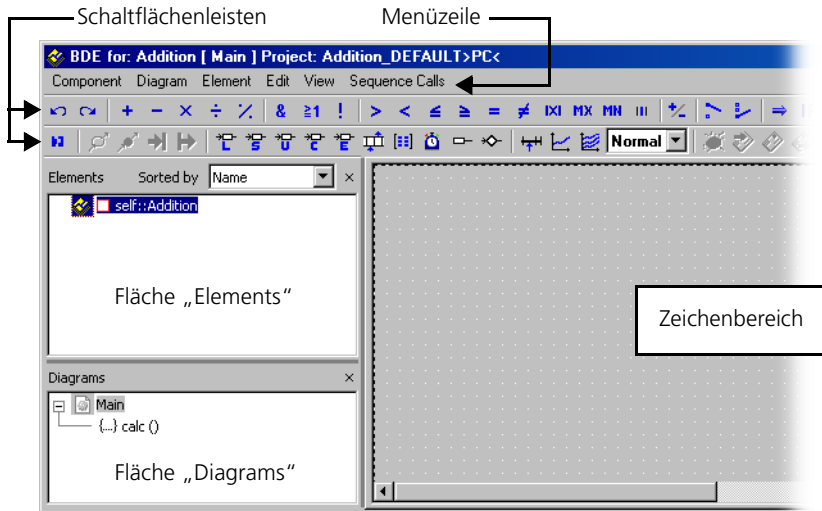
- Wählen Sie im Blockdiagrammeditor **Component** → **Exit**

oder

- Klicken Sie auf das **X** am rechten Rand der Titelzeile.  
Falls ein Diagramm ungesicherte Änderungen enthält, werden Sie gefragt, ob Sie die Änderungen sichern wollen.
- Aktivieren Sie die Option **Remember my Decision**, wenn Sie *jede* Abfrage dieser Art auf dieselbe Weise beantworten wollen.  
Das Fenster „Save Changes in Graphic“ wird dann nicht mehr angezeigt. Sie können die Einstellung im ASCET-Optionsfenster, Knoten „Confirmation Dialogs“ , wieder rückgängig machen (s. „Optionen für Bestätigungsfenster“ auf Seite 47).
- Bestätigen Sie die Sicherung mit **Yes**.  
Die Änderungen werden im Cache-Speicher gesichert, der Blockdiagrammeditor wird geschlossen.
- Verwerfen Sie die Änderungen mit **No**.  
Der Blockdiagrammeditor wird geschlossen, ohne dass die Änderungen gesichert werden.
- Klicken Sie **Cancel**, um das Schließen des Blockdiagrammeditors abubrechen.

#### 4.1.1 Die Benutzeroberfläche des Blockdiagrammeditors

Der Blockdiagrammeditor besitzt zwei Register mit verschiedenen Anzeigemodi. Diese beiden Modi besitzen unterschiedliche Funktionalitäten. Durch Klicken auf ein Register können Sie den Modus wechseln.



Die Abbildungen zeigen den Blockdiagrammeditor im Spezifizierungsmodus mit Bezeichnung der wichtigsten Teile.

Die Titelleiste des Blockdiagrammeditors enthält die folgenden Informationen über das aktuelle Blockdiagramm: Name der Komponente, Name des aktuellen Diagramms, Name des aktuellen Projekts und aktuelles Target.

Der *Zeichenbereich* ist der Bereich, in dem das eigentliche Blockdiagramm erstellt wird. Die Elemente der Komponente (d.h. die Eingänge, Ausgänge und Variablen) werden in einer Baumstruktur in der *Fläche „Elements“* dargestellt, deren Stamm der Name der Komponente selbst ist. Die Diagramme und Methoden oder Prozesse werden in der *Fläche „Diagrams“* dargestellt. Die Flächen „Elements“ und „Diagrams“ können bei Bedarf aus- und wieder eingeblendet werden.

Die Größe des Zeichenbereichs kann im ASCET-Optionsfenster eingestellt werden. Achten Sie dabei auf folgendes: Wenn Sie eine Größe über der in früheren ASCET-Versionen (V5.2.0 und älter) fest vorgegebenen Größe (2000 x 2000 px) wählen und den großen Bereich beim Spezifizieren der Komponente ausnutzen, kann die Komponente in älteren ASCET-Versionen nicht vollständig angezeigt werden. Das bedeutet auch, dass Sie Elemente außerhalb des Zeichenbereichs nicht entfernen können, weil diese Elemente nicht aus dem Diagramm entfernt werden können.

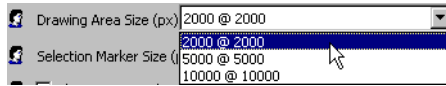
#### **Größe des Zeichenbereichs einstellen:**

---

Die eingestellte Größe gilt für alle grafischen Editoren – Blockdiagrammeditor (auch für CT-Blöcke), Zustandsautomaten-Editor, Projekteditor (Register „Graphic“). Bereits geöffnete Fenster sind von einer Änderung allerdings *nicht* betroffen.

- Wählen Sie im **Komponentenmanager Tools** → **Options**, um das ASCET-Optionsfenster zu öffnen.
- Öffnen Sie den Knoten „Block Diagram“ (s. Seite 56).

- Wählen Sie im Kombikästchen „Drawing Area Size (px)“ die gewünschte Größe.



Wenn Sie einen höheren Wert als 2000 @ 2000 gewählt haben, erscheint folgende Meldung:

```
If a size > 2000 @ 2000 px is
selected, components may be
displayed incompletely in
ASCET versions < V5.2.1!
```

- Bestätigen Sie die Meldung mit **OK**, um die Größe auszuwählen.
- Schließen Sie das Optionsfenster mit **OK**.

Wenn Sie nun den Blockdiagrammeditor (oder einen anderen grafischen Editor) öffnen, hat der Zeichenbereich die ausgewählte Größe.

### Öffnen einer zu großen Komponente:

---

Wenn Sie eine Komponente öffnen, die über die aktuell gewählte Größe des Zeichenbereichs hinausragt, öffnet sich ein Meldungsfenster, das die Größe des Zeichenbereichs und die Größe des Blockdiagramms nennt und Ihnen die Möglichkeit gibt, den Zeichenbereich zu vergrößern.

- Aktivieren Sie die Option **Remember my Decision**, wenn Sie *jede* Abfrage dieser Art auf dieselbe Weise beantworten wollen.

Das Fenster „Resize Drawing Area“ wird dann nicht mehr angezeigt. Sie können die Einstellung im ASCET-Optionsfenster, Knoten „Confirmation Dialogs“, wieder rückgängig machen (s. „Optionen für Bestätigungsfenster“ auf Seite 47).

- Klicken Sie im Meldungsfenster auf **Yes**, um eine passende Größe für den Zeichenbereich einzustellen.

Die Größe wird so gewählt, dass die Komponente vollständig angezeigt wird.

- Klicken Sie auf **No**, wenn Sie die Größe des Zeichenbereichs nicht ändern wollen.  
Sie können die Komponente nicht vollständig sehen. Elemente, die außerhalb des Zeichenbereichs liegen, können nicht entfernt werden.

### Ein- und Ausblenden der Listen:

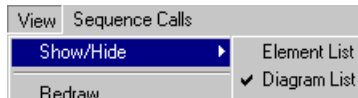


- Klicken Sie auf das **X** neben dem Namen der Liste, die Sie ausblenden wollen,

*oder*

- wählen Sie **View** → **Show/Hide** → **Element List** bzw. **Diagram List**.

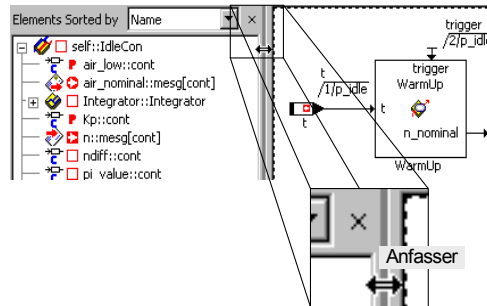
Die gewählte Liste verschwindet. Im Menü **View** → **Show/Hide** wird das entsprechende Häkchen entfernt.



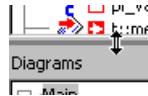
- Um die Liste wieder anzuzeigen, wählen Sie erneut **View** → **Show/Hide** → **Element List** bzw. **Diagram List**.

## Ändern von Breite und Höhe der Listen:

- Bewegen Sie den Mauszeiger auf den Trennbalken, der sich zwischen der Fläche „Elements“ und dem Zeichenbereich befindet. Der Mauszeiger verwandelt sich in einen Doppelpfeil zu Größenänderung.



- Ziehen Sie den Trennbalken nach rechts oder nach links, um die Breite der Listen zu ändern.
- Ziehen Sie den Trennbalken zwischen den Flächen „Elements“ und „Diagrams“ nach oben oder unten, um die Höhe anzupassen.



Um mehr Informationen über die Elemente der spezifizierten Komponente zu bekommen, ist es – in allen Komponenteneditoren – möglich, in einen Browser-Modus zu wechseln.

### Wechsel des Anzeigenmodus:

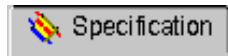


- Klicken Sie auf die Lasche **Browse**. Sie erhalten den Browser-Modus:

Name	Type	MaxSize	Scope	Kind	Existence	Dependency	Memory	Calibration	Unit	Comment
distrib	distrib[cont]	5	local	Parameter	real	independent	nonvolatile	calibration		
mat_result	msg[cont]	---	exported	Send Message	real	independent	volatile	calibration	v	voltage
matrix	mat[cont]	4 @ 5	local	Variable	real	independent	volatile	calibration		
One_D	1D[cont->cont]	5	local	Parameter	real	independent	nonvolatile	calibration		
tab_result	msg[cont]	---	exported	Send Message	real	independent	volatile	calibration		
Two_D	2D[cont,cont->]	5 @ 6	local	Parameter	real	independent	nonvolatile	calibration		



Diese Ansicht besitzt die gleiche Funktionalität und den gleichen Aufbau wie das im Komponentenmanager vorhandene Ansichtenkonzept (siehe Kapitel 2.1.3 und Kapitel 2.3.2). Die Menüfunktionen sind hier allerdings nur über die Kontextmenüs der Register verfügbar.



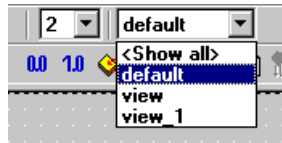
- Klicken Sie auf die Lasche **Specification**, um zur Spezifikationsansicht zurückzukehren.

### Auswählen einer Sicht des Diagramms:

---

Details zu Sichten finden Sie in Kapitel 7.3 „Sichten“ auf Seite 736.

- Wählen Sie aus dem Kombikästchen „Sicht“ in der rechten oberen Ecke der Werkzeugleiste eine Sicht.



Die Darstellung der Komponente ändert sich gemäß den Einstellungen für die gewählte Sicht. Alle in dieser Sicht als unsichtbar markierte Diagrammelemente werden verborgen.

### Beschreibung der Menüfunktionen

---

- **Component**
  - *Clean code generation directory*  
Löscht alle Dateien im Codegenerierungsverzeichnis.
  - *Touch*  
Erzwingen einer Neugenerierung bei der nächsten Codegenerierung  
(*Flat* → die editierte Komponente,  
*Recursive* → auch die referenzierten Komponenten).
  - *Generate Code* (<CTRL> + <F7>)  
Der Code für eine Komponente wird generiert.
  - *Compile*  
Kompiliert den generierten Code.
  - *Open Experiment*  
Ein Experiment wird gestartet.

- *Default Project*  
Erlaubt die Bearbeitung des Standardprojektes, das beim Experimentieren mit Komponenten verwendet wird  
(*Edit* → öffnet einen Editor für das Standardprojekt,  
*Resolve Globals* → erzeugt für jedes importierte Element der Komponente, für das kein exportiertes Element vorhanden ist, automatisch ein globales Element,  
*Delete Unused Globals* → löscht nicht genutzte globale Elemente).
- *Edit Layout*  
Der Layout-Editor wird geöffnet.
- *Edit Data*  
Der Dateneditor für eine Komponente wird geöffnet. Das Durchsuchen der Daten einer Komponente ist möglich.
- *Edit Implementation*  
Der Implementierungseditor wird geöffnet. Das Durchsuchen der Implementierungen einer Komponente ist möglich.
- *Edit Notes*  
Öffnet den Notizeneditor - Hier können Notizen zur Komponente gemacht werden.
- *Check Dependency*  
Es wird überprüft, ob die Zuweisung der formalen Parameter zu den Modellparametern korrekt ist.
- *Export Data*  
Exportieren des Datensatzes einer Komponente.
- *Delete Unused Elements*  
Löscht die Elemente aus dem Feld „Elements“, die kein Vorkommen im Diagramm haben (für Zustandsautomaten deaktiviert).
- *Show Path*  
Zeigt den Pfad einer aufgenommenen Komponente.
- *Copy Path to Clipboard*  
Kopiert den Pfad einer aufgenommenen Komponente in die Zwischenablage  
(*Model Path* → Pfad im Modell,  
*Model asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente im Modell als Hyperlink ermöglicht,  
*Database Path* → Pfad in der Datenbank,

*Dabase asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente in der Datenbank als Hyperlink ermöglicht).

– *File out Generated Code*

Speichern des generierten Codes im Dateisystem  
(*Flat* → die editierte Komponente,  
*Recursive* → auch die referenzierten Komponenten).

– *View Generated Code*

Generiert den Code für die Komponente und zeigt diesen in einem Texteditor an.

Der Texteditor kann im ASCET-Optionsfenster, Knoten „ASCII Editor“ (s. Seite 61), gewählt werden.

– *File Out Generic Code For External Make*

Speichern des generierten Codes in ein beliebiges Verzeichnis zur späteren Weiterverarbeitung durch separate Werkzeuge, z.B. in einem externen Make/Build - Prozess.

– *Exit*

Beenden des Blockdiagrammeditors

• **Diagram**

(steht in der Fläche „Diagrams“ auch als Kontextmenü zur Verfügung)

– *Store to Cache*

Die Spezifikation einer Komponente wird im Cache-Speicher gespeichert. (Es erfolgt keine permanente Speicherung in der Datenbank.)

– *Analyze Diagram*

Analysieren des aktuellen Diagramms.

– *Load Diagram*

Laden eines Diagramms.

– *Add Diagram*

Erzeugen eines neuen Diagramms.  
(*Public* → enthält nur öffentliche Methoden,  
*Private* → enthält nur private Methoden)

– *Rename Diagram*

Umbenennen eines Diagramms.

– *Delete Diagram*

Löschen eines Diagramms aus der Komponente.

- *Add Method*  
Erzeugen einer Methode. Verfügbar für Klassen und Module.
- *Add Process*  
Erzeugen eines Prozesses.
- *Add Trigger / Action / Condition*  
Nur im Zustandsautomaten-Editor verfügbar, siehe „Spezielle Menüfunktionen“ auf Seite 286.
- *Edit*  
Bearbeiten der Schnittstelle einer Methode / eines Prozesses.
- *Copy*  
Kopieren einer ausgewählten Methode / eines Prozess.
- *Rename*  
Umbenennen einer Methode / eines Prozesses.
- *Delete*  
Löschen einer Methode / eines Prozesses.
- *Move Up*  
Bewegen eines Diagramms oder einer Methode / eines Prozesses (nach oben).
- *Move Down*  
Bewegen eines Diagramms oder einer Methode / eines Prozesses (nach unten).
- *Move*  
Bewegen von Methoden / Prozessen zwischen Diagrammen.
- *Edit Implementation*  
Spezifizieren der Implementierung einer Methode / eines Prozesses.
- **Element**  
(steht in der Fläche „Elements“ auch als Kontextmenü zur Verfügung)

#### Hinweis

Die Funktionen **Add Item**, **Rename**, **Delete** und **Edit** stehen nur zur Verfügung, wenn ASCET-MD installiert ist.

- *View* → *Collapse all* klappt die Baumstruktur in der Fläche „Elements“ zu, sodass nur noch die Komponente selbst zu sehen ist.

- *View* → *Expand all* klappt die Baumstruktur vollständig auf, sodass der gesamte Inhalt der Komponente angezeigt wird.
- *Add Item*  
Aufnehmen einer Komponente als komplexes Element.
- *Rename* (<F2>)  
Umbenennen eines ausgewählten Diagrammelements.
- *Delete* (<DEL>)  
Löschen eines ausgewählten Diagrammelements.
- *Show Occurences*  
Anzeigen aller Vorkommen eines Elements (nur im Blockdiagramm-editor).
- *Edit*  
Bearbeiten der Konfiguration eines ausgewählten Elements.
- *Edit Data*  
Bearbeiten der Daten eines ausgewählten Elements.
- *Edit Implementation*  
Bearbeiten der Implementierung eines ausgewählten Elements.
- *Set Cache Locking*  
Diese Menüfunktion ist nur für die Arbeit mit ASCET-RP von Bedeutung. Sie ist im ASCET-RP-Handbuch beschrieben.
- *Edit Component*  
Öffnen des Spezifikationseditors für eine ausgewählte aufgenommene Komponente.
- *Replace Component*  
Ersetzen einer Komponente durch eine andere Komponente. Der Name der alten Komponente bleibt erhalten.
- *Show Path*  
Zeigt den Pfad einer ausgewählten aufgenommenen Komponente.
- *Copy Path to Clipboard*  
Kopiert den Pfad einer aufgenommenen Komponente in die Zwischenablage  
(*Model Path* → Pfad im Modell,  
*Model asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente im Modell als Hyperlink ermöglicht,

*Database Path* → Pfad in der Datenbank,  
*Database asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente in der Datenbank als Hyperlink ermöglicht).

- *Notes*  
Öffnet den Notizeneditor für eine ausgewählte aufgenommene Komponente.
- *Edit Distribution*  
Öffnet einen Editor für die ausgewählte Distribution (nur bei Gruppentabellen).
- *Edit Max Size*  
Öffnet einen Editor, in dem die maximale Größe eines Arrays, einer beliebigen Tabelle oder Matrix vorgegeben werden kann.
- *Copy Elements* (<CTRL> + <C>)  
Kopieren eines oder mehrerer ausgewählter Elemente aus der Liste „Elements“.
- *Paste Elements* (<CTRL> + <V>)  
Einfügen eines oder mehrerer kopierter Elemente in die Liste „Elements“.
- *File Out Data*  
Schreiben der Daten eines Arrays oder einer Tabelle in eine Datei.
- *File In Data*  
Lesen der Daten eines Arrays oder einer Tabelle aus einer Datei.
- *Export Data*  
Exportieren des Datensatzes einer aufgenommenen Komponente.

- **Edit**

- *Cut* (<CTRL> + <X>)  
Ausschneiden eines Diagrammelements.
- *Copy* (<CTRL> + <C>)  
Kopieren eines Diagrammelements.
- *Paste* (<CTRL> + <V>)  
Einfügen eines Diagrammelements.
- *Delete* (<DEL>)  
Löschen einer Verbindung oder eines Elements.

- *File In Buffer*  
Lesen eines Diagramms aus einer Datei.
- *File Out Buffer*  
Teile eines Diagramms werden in einer Datei gespeichert.
- *Views*  
Öffnet das Dialogfenster „Views“. Die Sichten, in denen das Diagrammelement sichtbar ist, sind aufgelistet und können bearbeitet werden.
- **View**
  - *Show/Hide*  
Zeigt/verbirgt verschiedene Teile des Editors oder der Komponente.  
(*Element List* → Liste „Elements“,  
*Diagram List* → Liste „Diagrams“,  
*Connected Elements* → Browser-Feld für die Elemente, die mit einem ausgewählten Diagrammelement verbunden sind (siehe Seite 233),  
*Impl. Casts in Element List* → Implementation-Casts in der Liste „Elements“; im Zeichenbereich bleiben die Implementation-Casts sichtbar, wenn für die Sicht nichts anderes ausgewählt ist.)
  - *Redraw*  
Zeichnet das Diagramm neu.
  - *Rebuild Connections*  
Glättet fragmentierte Verbindungen, die bei der automatischen Konvertierung von Operatorimplementierungen (s. Seite 549) entstanden sind.
  - *Undo* (<CTRL> + <Z>)  
Rückgängigmachen der letzten Aktion.
  - *Redo* (<CTRL> + <Y>)  
Erneute Ausführung einer rückgängig gemachten Aktion.
  - *Show Hierarchy Path*  
Der komplette Hierarchiepfad der Komponente wird angezeigt.
  - *Parent Component*  
Öffnet einen Editor für die aufnehmende Komponente. (Der Punkt ist nur verfügbar, wenn eine aufgenommene Komponente editiert wird.)

- *Parent Hierarchy Level*  
Der Hierarchiepfad wird angezeigt.
- *Page frame Portrait*  
Darstellung des Diagramms in Hochformat.
- *Page frame Landscape*  
Darstellung des Diagramms in Querformat.
- *Grid*  
Einstellung des Rasters im Zeichenbereich.
- *Print Diagram*  
Die Komponente wird gedruckt.
- *Save as Postscript / Bitmap/ RTF / GIF*  
Speichert das Diagramm im angegebenen Format.
- **Sequence Calls**
  - *Sequencing - Ignore Info*  
Automatisches Zuweisen von Sequenzaufrufen.
  - *Sequencing Starting With*  
Automatisches Zuweisen von Sequenzaufrufen ab einer bestimmten Nummer.
  - *Sequencing Appending*  
Anhängen von Sequenzaufrufen an eine existierende Sequenz.
  - *Scale To Step Size*  
Skalieren von Sequenzaufrufen  
(*For Diagram* → für das gesamte Diagramm,  
*For Method/Process* → für eine einzelne Methode / Prozess)
  - *Reset*  
Zurücksetzen von Sequenzaufrufen  
(*For Diagram* → für das gesamte Diagramm,  
*For Method/Process* → für eine einzelne Methode / Prozess,  
*For Selected Blocks* → für ausgewählte Blöcke)
  - *Show*  
Anzeigen von Gruppen von Sequenzaufrufen  
(*For Diagram* → für das gesamte Diagramm,  
*For Method/Process* → für eine einzelne Methode / Prozess,  
*For Selected Blocks* → für ausgewählte Blöcke,  
*Unused* → unbenutzte Sequenzaufufe)



- *Hide*  
Verbergen von Gruppen von Sequenzaufrufen  
(enthält dieselben Menüfunktionen wie **Sequence Calls** → **Show**)
- *Next Seq. Call* (<CTRL> + <PFEIL RECHTS>)  
Bewegen zwischen Sequenzaufrufen (nächstfolgender Aufruf).
- *Previous Seq. Call* (<CTRL> + <PFEIL LINKS>)  
Bewegen zwischen Sequenzaufrufen (vorhergehenderer Aufruf).

#### 4.1.2 Definieren der Schnittstelle einer Komponente

---

Der erste Schritt beim Spezifizieren einer Komponente ist das Definieren ihrer Schnittstelle. Die Schnittstelle bestimmt, wie die Komponente mit anderen Komponenten in Wechselwirkung steht, d. h. welche Daten sie empfängt und weiterleitet und wie sie adressiert werden kann. Die Schnittstelle ist der Hauptunterschied beim Spezifizieren verschiedener Typen von Komponenten im Blockdiagrammeditor; deshalb wird dies für Klassen und Module getrennt erläutert.

##### *Klassen*

---

Die Schnittstelle einer Klasse besteht aus ihren öffentlichen Methoden und deren Argumenten und Rückgabewerten. Die Argumente bilden die Eingabeparameter der Klasse, die Rückgabewerte bilden die Ausgabeparameter. Die Methoden können von außen angeregt werden, die Berechnungen innerhalb einer Komponente auszulösen.

##### **Erzeugen einer Methode:**

---

- Wählen Sie **Diagram** → **Add Method**.  
Eine neue Methode wird in der Fläche „Diagrams“ erzeugt.
- Geben Sie einen Namen für die Methode ein und drücken Sie <ENTER>.

Jede Klasse enthält eine oder mehrere Methoden, die Standardmethode `calc` wird automatisch erzeugt. Eine Methode kann eine Anzahl von Argumenten und einen Rückgabewert haben. Die Argumente und der Rückgabewert können unter Verwendung des Schnittstelleneditors geändert werden.

## Bearbeiten der Schnittstelle einer Methode:

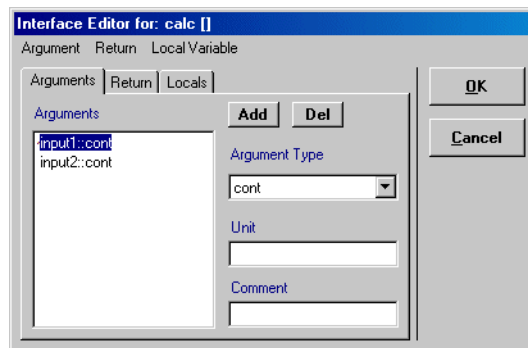
---

- Wählen Sie in der Fläche „Diagrams“ den Namen der Methode aus, die Sie ändern möchten.
- Wählen Sie **Diagram** → **Edit**

*oder*

- doppelklicken Sie auf die Methode.

Der Schnittstelleneditor für die Methode wird geöffnet. In diesem Fenster können Sie der Methode Argumente und den Rückgabewert zuweisen und lokale Variablen definieren.



- Definieren Sie die Argumente, den Rückgabewert und die lokalen Variablen für die Methode, wie nachstehend beschrieben.
- Klicken Sie auf **OK**, um den Schnittstelleneditor zu schließen.

## Hinzufügen eines Arguments zu der Methode:

---

- Aktivieren Sie im Schnittstelleneditor das Register „Arguments“.
- Klicken Sie auf die Schaltfläche **Add**

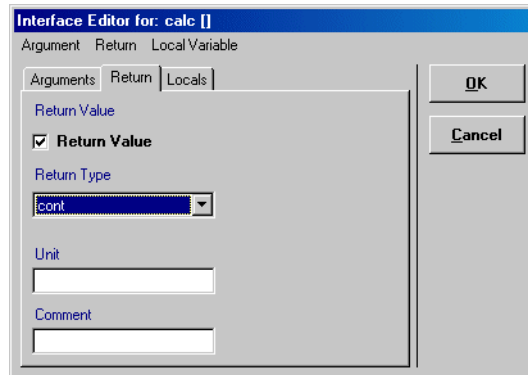
*oder*

- wählen Sie **Argument** → **Add**.  
Ein neues Argument wird zur Liste „Arguments“ hinzugefügt. Sein Name wird hervorgehoben, um ein Bearbeiten an Ort und Stelle zu ermöglichen.
- Benennen Sie das Argument und drücken Sie <ENTER>.
- Legen Sie den Typ des Arguments fest, indem Sie den gewünschten Wert aus dem Kombikästchen „Argument Type“ auswählen.  
Der Standard-Argumenttyp ist `cont`.
- Wenn Sie als Typ einen Array oder eine Matrix gewählt haben, wählen Sie **Argument** → **Edit Max Size**.
- Geben Sie im Fenster „Max Size for: <argument>“ die Array- oder Matrixgröße ein.
- Geben Sie in das Feld „Unit“ eine Maßeinheit für das Argument ein.  
Die Maßeinheit hat nur eine beschreibende Funktion und keinen Einfluss auf die Funktionalität der Klasse.
- Geben Sie einen Kommentar, der das Argument betrifft, in das Feld „Comment“ ein.  
Mit jedem Argument oder Rückgabewert kann ein Kommentar verknüpft sein. Dieser Kommentar wird angezeigt, wenn die Dokumentation für die Komponente generiert wird.

## Hinzufügen eines Rückgabewerts zu der Methode:

---

- Aktivieren Sie im Schnittstelleneditor das Register „Return“.



- Aktivieren Sie das Feld **Return Value**, wenn die Methode einen Rückgabewert haben soll.
- Legen Sie den Datentyp des Rückgabewerts fest.
- Wenn Sie als Typ einen Array oder eine Matrix gewählt haben, wählen Sie **Return** → **Edit Max Size**.
- Geben Sie im Fenster „Max Size for: return“ die Array- oder Matrixgröße ein.
- Geben Sie einen Kommentar und eine Maßeinheit ein.

## Hinzufügen lokaler Variablen zu der Methode:

---

- Aktivieren Sie im Schnittstelleneditor das Register „Locals“.  
Das Register „Locals“ enthält dieselben Elemente wie das Register „Arguments“.
- Klicken Sie auf die Schaltfläche **Add**  
*oder*  
• wählen Sie **Local Variable** → **Add**.  
Eine neue lokale Variable wird zur Liste „Local Variables“ hinzugefügt.

- Geben Sie einen Namen für die lokale Variable ein.
- Legen Sie den Typ der lokalen Variablen fest.
- Wenn Sie als Typ einen Array oder eine Matrix gewählt haben, wählen Sie **Local Variable** → **Edit Max Size**.
- Geben Sie im Fenster „Max Size for: <local>“ die Array- oder Matrixgröße ein.
- Geben Sie einen Kommentar und eine Maßeinheit für die Variable ein.

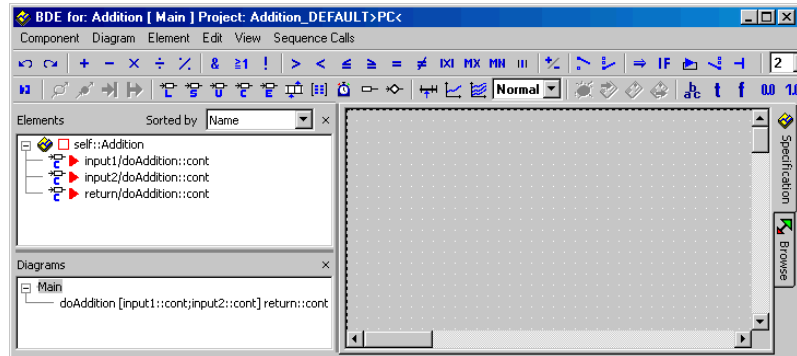
#### **Bearbeiten von Argumenten und lokalen Variablen:**

---

- Wählen Sie **Argument** → **Rename**, um ein Argument umzubenennen.
- Wählen Sie **Argument** → **Move Up** oder **Move Down**, um ein Argument in der Liste zu verschieben.
- Wählen Sie **Argument** → **Delete**  
*oder*
- klicken Sie auf **Del**, um ein Argument zu löschen.

Für die Bearbeitung von lokalen Variablen ist im Register „Locals“ das Menü **Local Variable** mit den gleichen Menüfunktionen verfügbar.

Die Argumente, lokalen Variablen und der Rückgabewert werden zur Liste „Elements“ des Blockdiagrammeditors hinzugefügt. Jedes Argument und jeder Rückgabewert wird als ein Anschluss am grafischen Block der Komponente dargestellt.



### Umbenennen oder Löschen einer Methode:

---

- Wählen Sie in der Fläche „Diagrams“ die Methode aus.
- Wählen Sie **Diagram → Rename**  
*oder*
- drücken Sie <F2>, um die Methode umzubenennen.
- Wählen Sie **Diagram → Delete**  
*oder*
- drücken Sie <DEL>, um die Methode zu löschen.

In der Liste „Diagrams“ sind Diagramme und Methoden standardmäßig in chronologischer Reihenfolge angeordnet. Jedes neue Diagramm bzw. jede neue Methode wird am Ende der Liste hinzugefügt. Die Reihenfolge kann durch Bewegen von Diagrammen und Methoden innerhalb der Liste geändert werden.

### Bewegen eines Diagramms oder einer Methode:

---

- Wählen Sie in der Liste „Diagrams“ den zu bewegendem Eintrag aus.
- Wählen Sie **Diagram → Move Up**, um den Eintrag eine Position nach oben zu bewegen.

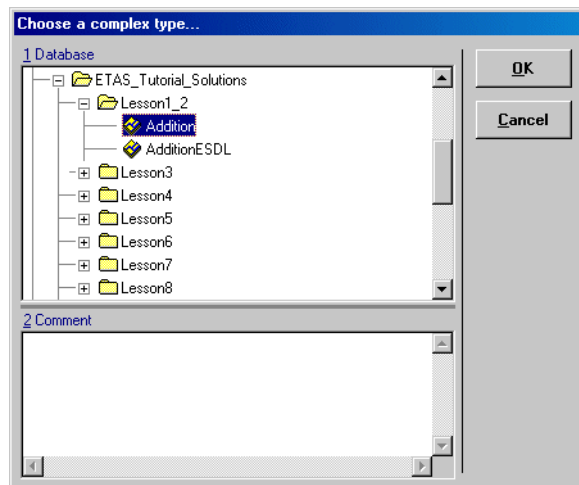
- Wählen Sie **Diagram** → **Move Down**, um den Eintrag eine Position nach unten zu bewegen.

Sie können andere Komponenten als Schnittstellenelemente von Komponenten verwenden. Die Vorgehensweise wird hier für ein Argument vom Typ *benutzerdefiniert* beschrieben. Die gleiche Vorgehensweise kann für einen komplexen Rückgabewert angewandt werden.

### Zuweisen einer Komponente als Argument:

- Öffnen Sie den Schnittstelleneditor für die Methode, zu der Sie die Komponente hinzufügen möchten.
- Wählen Sie aus der Liste „Arguments“ den Eintrag aus, für den Sie den Datentyp spezifizieren möchten (oder fügen Sie einen neuen Eintrag hinzu).
- Wählen Sie in der Liste „Argument Type“ die Eintragung `<user defined>` aus.

Es erscheint ein Auswahl-Dialogfenster für die aktuelle Datenbank.



- Wählen Sie aus der Liste „1 Database“ die gewünschte Komponente aus und klicken Sie auf **OK**, um das Dialogfenster zu schließen.  
Die ausgewählte Komponente erscheint im Feld „Argument Type“ des Schnittstelleneditors.
- Klicken Sie auf **OK**, um Ihre Änderungen der Schnittstelle zu speichern.

### *Module*

---

Die Schnittstelle eines Moduls besteht aus Prozessen und – wenn vorhanden – öffentlichen Methoden. Für jedes neue Modul wird automatisch ein Standardprozess erzeugt, weitere Prozesse und ggf. Methoden müssen Sie anlegen (siehe Seite 201). Prozesse bestimmen die Aktivierung der Funktionalität des Moduls, doch sie definieren nicht Eingänge oder Ausgänge. Module kommunizieren und interagieren über Messages (Nachrichten und globale Elemente (importierte, exportierte Elemente).

#### **Erzeugen eines Prozesses:**

---

- Wählen Sie **Diagram** → **Add Process**.  
Es wird ein neuer Prozess erzeugt, und in der Fläche „Diagrams“ wird sein Name hervorgehoben, um ein Bearbeiten an Ort und Stelle zu ermöglichen.
- Geben Sie einen Namen für den Prozess ein und drücken Sie <Enter>.  
Sie können Prozesse auf die gleiche Weise wie Methoden bewegen (s. Seite 206), umbenennen oder löschen (s. Seite 206).

Da ein Prozess keinerlei Argumente hat, sind im Schnittstelleneditor nur das Register „Locals“ und das Menü **Local Variable** vorhanden (s. Seite 204).

#### 4.1.3 Komplexe Typen als Schnittstellenelemente

---

In diesem Abschnitt wird anhand von Beispielen die Verwendung von zusammengesetzten und benutzerdefinierten Elementen als Schnittstellenelemente erklärt.

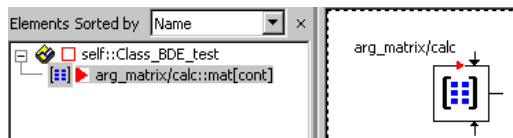


## Arrays und Matrizen

Wenn Sie ein zusammengesetztes oder benutzerdefiniertes Schnittstellenelement (Array, Matrix, Klasse) verwenden, können Sie zum einen über die zum Element gehörigen Anschlüsse darauf zugreifen. Zum anderen können Sie über die Get-/Set-Ports auf die gesamte Datenstruktur zugreifen. Am Beispiel einer Matrix wird das Vorgehen beschrieben.

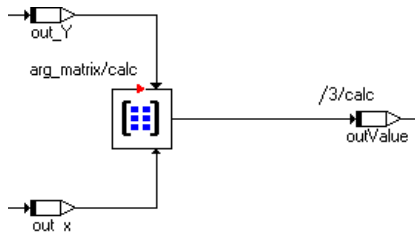
### Matrixargument erzeugen:

- Fügen Sie zu einer Methode ein Argument vom Argumenttyp `mat (cont)` hinzu (s. Seite 202).
- Ordnen Sie das Argument im Zeichenbereich an.



Das Element verfügt nur über die Ausleseanschlüsse (s.a. „Arrays und Matrizen“ im ASCET-Referenzhandbuch), ein Schreiben der Matrix ist nicht möglich.

- Verwenden Sie die üblichen Ausleseanschlüsse der Matrix wie in der Abbildung.

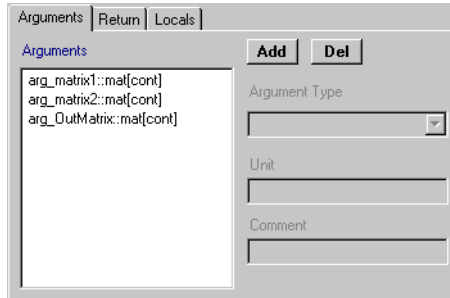


Das Verfahren, um Matrizen oder Arrays zurückzugeben, ist aufwendiger. Es wird im folgenden am Beispiel einer Addition von zwei Matrizen beschrieben.

### Erstellen der Berechnungsklasse:

- Erzeugen Sie eine Klasse mit einer Methode, die die Berechnung mit den Matrizen enthalten soll.

- Erzeugen Sie in der Methode zwei Argumente vom Typ `mat (cont)` für die zu addierenden Matrizen.
- Erzeugen Sie ein zusätzliches Argument (`arg_OutMatrix`) vom selben Typ für die resultierende Matrix.



Da das Argument nicht beschrieben werden kann, brauchen Sie eine Hilfsmatrix.



- Erzeugen Sie mit der Schaltfläche **Matrix** eine Matrix vom Typ `cont`.
- Übernehmen Sie im Fenster „Max Size for“ die voreingestellten Werte.

Dieser Hilfsmatrix wird über den Get-Port das dritte Argument zugewiesen. Damit ist die Größe festgelegt, sie braucht nicht extra eingegeben werden.

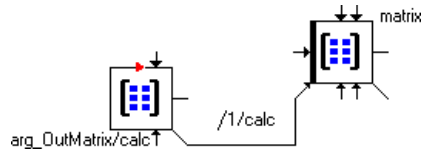
#### Hinweis

Da den Matrizenargumenten keine Größe zugewiesen wurde, **muss** die voreingestellte Größe übernommen werden. Andernfalls liefert die Codegenerierung eine Fehlermeldung.

#### Spezifizieren der Matrizenaddition:

- Ordnen Sie die Hilfsmatrix und das Argument `arg_OutMatrix` im Zeichenbereich an.

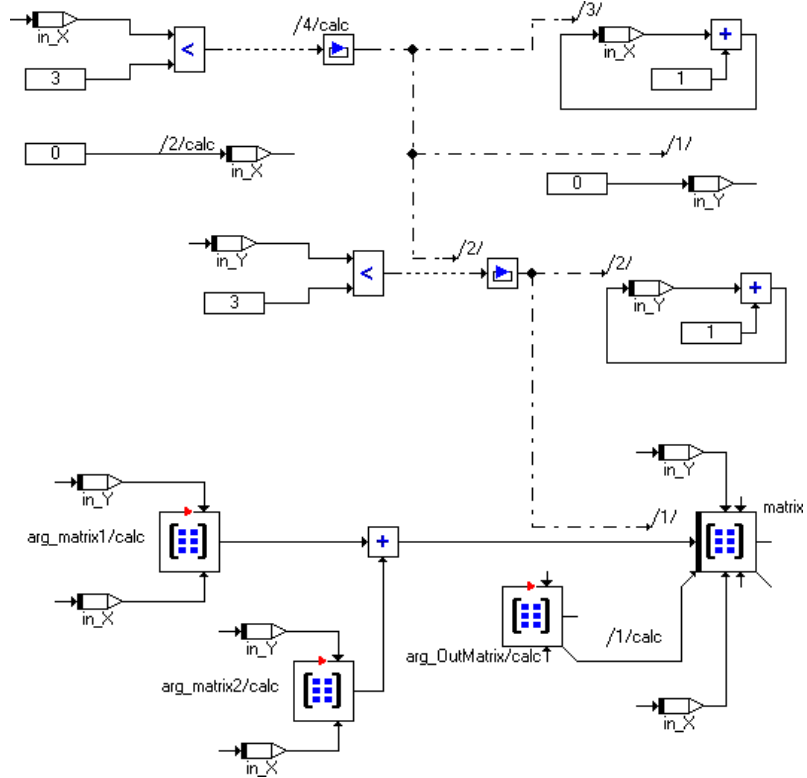
- Klicken Sie mit der rechten Maustaste nacheinander auf die beiden Elemente und wählen Sie **Get/Set Ports** aus dem Kontextmenü.  
Der Get-Port des Arguments sowie die Get- und Set-Ports der Hilfsmatrix werden im Diagramm angezeigt.
- Weisen Sie der Hilfsmatrix über den Set-Port das Argument `arg_OutMatrix` zu.



- Geben Sie dem Sequenzaufruf die Nummer 1, damit die Zuweisung als erster Rechenschritt der Methode ausgeführt wird. (Sequenzaufrufe sind im Abschnitt „Sequenzaufrufe“ auf Seite 247 ausführlich beschrieben.)

Die Hilfsmatrix greift auf den Speicherbereich von `arg_OutMatrix` zu. Auf diese Weise werden die Berechnungsergebnisse außerhalb der berechnenden Methode verfügbar.

- Erzeugen Sie die notwendigen Indizes und spezifizieren Sie die Matrixaddition.



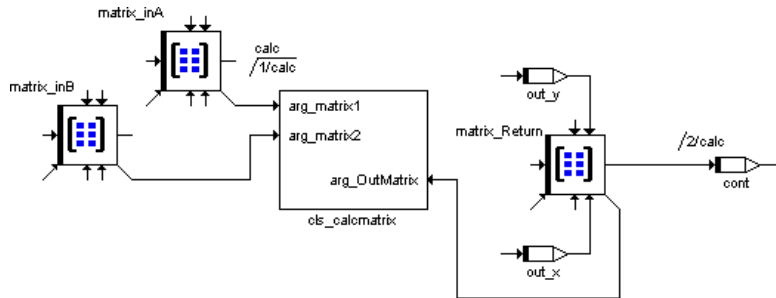
Das Beispiel geht von 3x3-Matrizen aus. Die Schleifen (aber *nicht* die Größen der Matrixargumente und der Hilfsmatrix!) müssen an die tatsächlichen Gegebenheiten angepasst werden.

Um die Berechnung tatsächlich auszuführen, müssen Sie eine weitere Klasse oder ein Modul erzeugen, in dem der Berechnungsklasse tatsächliche Werte für die Argumente übergeben werden.

#### Ausführen der Berechnung:

- Erzeugen Sie eine Klasse oder ein Modul.

- Nehmen Sie mit **Element** → **Add Item** die Berechnungsklasse als komplexes Element auf (s.a. Seite 229).
- Erzeugen Sie zwei Matrizen vom Typ `cont` und füllen Sie sie mit den Eingangsdaten.
- Erzeugen Sie eine dritte Matrix vom selben Typ und der gleichen Größe für das Ergebnis.
- Ordnen Sie die Elemente im Zeichenbereich an und verbinden Sie sie wie nachstehend dargestellt.



- Verwenden Sie die Ausgangsanschlüsse der Ergebnismatrix, um diese zu lesen.

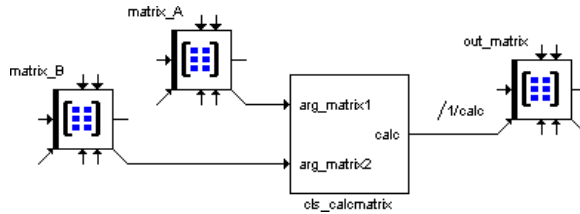
### Hinweis

Wenn Sie für die drei Matrizen eine andere als die im Fenster „Max size“ voreingestellte Größe gewählt haben, wird bei der Codegenerierung folgende **Warnung** angezeigt:

```
type mismatch in array max size: <1@1> and <x@y> -
accepted since larger matches smaller
```

Ein häufig gewählter Ansatz, eine Matrix oder ein Array zurückzugeben, ist der, einer Methode einen Rückgabewert des entsprechenden Typs zu geben und diesen über die Get- und Set-Ports zuzuweisen. Die nachstehende Abbil-

ung zeigt eine solche Anordnung. Die Klasse `cls_calcmatrix` enthält die Methode `calc` mit zwei Argumenten und einem Rückgabewert vom Typ `Matrix`.



*Das funktioniert jedoch nicht!* Über die Ports werden Zeiger übergeben, in diesem Fall auf eine Struktur innerhalb einer Methode. Die ist jedoch nur verfügbar, solange die Methode rechnet - das heißt für das Beispiel, dass die Daten in dem Moment, wo die Zuweisung zu `out_matrix` geschieht, nicht mehr vorhanden sind.

### Kennlinien und Kennfelder

Wenn Sie eine Kennlinie oder ein Kennfeld als Methodenargument übergeben wollen, können Sie das nicht direkt tun. Sie müssen das Element in eine Klasse einbetten; diese wird dann als Argument übergeben. Das Verfahren ist am Beispiel einer Kennlinie beschrieben.

#### Treffen der Vorbereitungen:

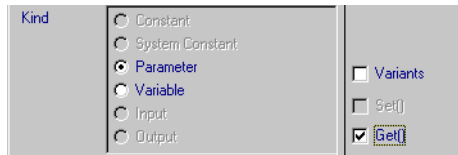


- Erzeugen Sie eine Klasse, z.B. mit Namen `Class_charline`.
- Legen Sie in der Klasse `Class_charline` mit der Schaltfläche **One D Table Parameter** eine Kennlinie an und richten Sie sie ein (siehe auch Seite 222).

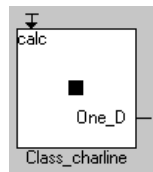
Sie müssen dafür sorgen, dass die Kennlinie auch von außerhalb der Klasse gelesen werden kann. Gehen Sie dazu wie folgt vor.

- Öffnen Sie den Elementeditor für die Kennlinie (s.a. „Elementkonfiguration“ auf Seite 482).

- Fügen Sie einen Ausgang für die Kennlinie hinzu, indem Sie die Option **Get()** aktivieren.

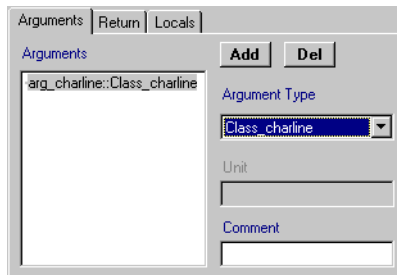


- Schließen Sie den Elementeditor.  
Im Layout der Klasse ist ein Ausgang für die Kennlinie hinzugefügt.



### Kennlinie als Methodenargument übergeben:

- Erzeugen Sie die Klasse, die die Kennlinie als Methodenargument aufnehmen soll.
- Fügen Sie zu einer Methode die Klasse `Class_charline` als Argument vom Typ `<user-defined>` hinzu, wie auf Seite 207 beschrieben.



- Klicken Sie **OK**, um den Schnittstelleneditor zu schließen.

## Kennlinie als Methodenargument verwenden:

Um das Methodenargument nun tatsächlich zu verwenden, gehen Sie wie folgt vor.

- Legen Sie das komplexe Argument im Zeichenbereich ab.

Anders als bei Matrix- und Arrayargumenten haben Sie hier keinen Zugriff auf die Anschlüsse der Kennlinie. Sie müssen diese also an eine lokale Kennlinie übergeben.



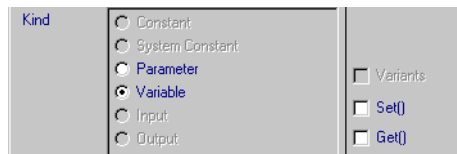
- Erzeugen Sie eine Kennlinie, richten Sie sie ein und ordnen Sie sie im Zeichenbereich an (siehe auch Seite 222).

- Klicken Sie mit der rechten Maustaste auf die Kennlinie und wählen Sie **Get/Set Ports** aus dem Kontextmenü.

Kennlinien und -felder werden als Parameter angelegt, das heißt, sie können im Programm nicht beschrieben werden. Deswegen wird nur der Get-Port verfügbar gemacht.

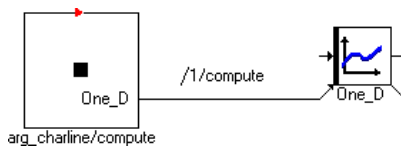


- Weisen Sie der Kennlinie im Elementeditor die Art **Variable** zu (siehe auch „Elementkonfiguration“ auf Seite 482).



Jetzt ist auch der Set-Port verfügbar und Sie können der Kennlinie das komplexe Argument zuweisen.

- Verbinden Sie den Ausgang des Arguments mit dem Set-Port der Kennlinie.





- Geben Sie dem Sequenzaufruf die Nummer 1, damit die Zuweisung als erster Rechenschritt der Methode ausgeführt wird.

### **Hinweis**

*Wird die Zuweisung nicht als erstes ausgeführt, kommt es zu Inkonsistenzen!*

Sie können jetzt die Kennlinie wie gewohnt auswerten. Dabei wird der Speicherbereich ausgelesen, in dem die im Argument übergebene Kennlinie liegt.

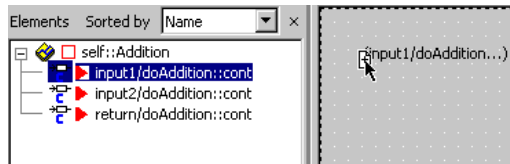
## 4.1.4 Erzeugen von Blockdiagrammen

Nachdem die Schnittstelle für eine Komponente definiert worden ist, werden die Argumente und Rückgabewerte oder die Eingabe- und Ausgabeparameter in der Fläche „Elements“ des Blockdiagrammeditors dargestellt. Sie können nun im Zeichenbereich angeordnet und mit anderen grafischen Elementen, wie etwa Variablen oder Operatoren, verbunden werden.

*Elemente, Operatoren und Verbindungen*

### **Anordnen eines Schnittstellenelements:**

- Wählen Sie das gewünschte Element aus der Fläche „Elements“ aus und ziehen Sie es an die gewünschte Stelle im Zeichenbereich.



Das Element wird im Zeichenbereich angeordnet.

- Sie können das Element im Zeichenbereich auswählen und ziehen, um es an eine andere Position zu bringen.

## Erzeugen eines Basiselements:

- Klicken Sie auf die Schaltfläche für das Element, das Sie erzeugen möchten, um den Mauszeiger mit dem entsprechenden Elementtyp zu laden.

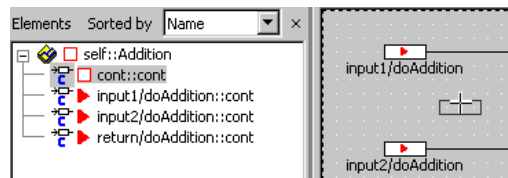
Die Schaltflächen für die Elemente befinden sich in der zweiten Reihe der Schaltflächenleiste.

Der zugehörige Elementeditor wird automatisch geöffnet und erlaubt es dem Anwender, die Elementeigenschaften unmittelbar anzupassen.

### Hinweis

Wenn das Öffnen des Elementeditors unmittelbar nach dem Erzeugen eines Elements unerwünscht ist, können Sie unten im Editor die Option **Always show Edit Dialog for new elements** oder im Knoten „Confirmation Dialogs“ (s. Seite 47) des Optionsfensters die Option **Edit primitive Element** deaktivieren.

- Klicken Sie in den Zeichenbereich, um das neue Element darin anzuordnen.



Im Diagramm erscheint das Element an dem Platz, auf den Sie geklickt haben. Sie können seine Position ändern, indem Sie es an eine andere Stelle ziehen.

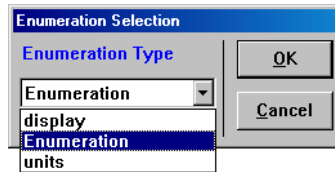
In der Liste „Elements“ wird ein neues Element hinzugefügt

Sie können die Eigenschaften eines Elements nachträglich ändern, indem Sie mit der rechten Maustaste darauf klicken. Daraufhin erscheint ein Kontextmenü mit verschiedenen Befehlen. Näheres dazu finden Sie im Kapitel „Bearbeiten von Elementen“ auf Seite 481.

## Einfügen einer Enumeration:



- Klicken Sie auf die Schaltfläche **Enumeration**. Das Auswahlfenster „Enumeration Selection“ wird geöffnet.



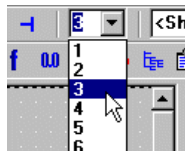
Alle in der Datenbank definierten Enumerationen werden im Kombikästchen „Enumeration Type“ angezeigt.

- Wählen Sie aus dem Kombikästchen die gewünschte Enumeration.
- Klicken Sie auf **OK**, um das Auswahlfenster zu schließen.

Der Mauszeiger wird mit dem Enumerations-typ geladen. Der Elementeditor wird geöffnet.

- Passen Sie die Elementeigenschaften nach Ihren Wünschen an und drücken Sie **OK**.
- Klicken Sie in das Zeichenfeld.
- Die Enumeration wird platziert.
- Geben Sie einen Namen für die Enumeration ein und drücken Sie <ENTER>.

## Anordnen eines Operators:



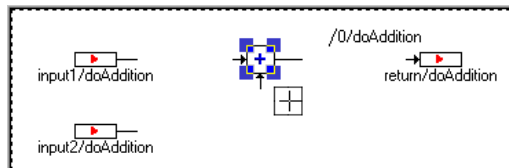
- Wählen Sie aus dem Kombikästchen „Argument Size“ die Anzahl der Eingänge des Operators aus.

### Hinweis

Für einige Operatoren (z.B. Division, Subtraktion, Vergleichsoperatoren) kann die Zahl der Eingänge nicht geändert werden. In diesen Fällen bleibt die Auswahl wirkungslos.

- Wählen Sie aus der Symbolleiste den Operator aus, den Sie erzeugen möchten, um den Mauszeiger mit diesem Operator zu laden.
- Klicken Sie in den Zeichenbereich, um den Operator darin anzuordnen.

Der Operator wird zum Diagramm hinzugefügt. Sie können seine Position korrigieren, indem Sie ihn an eine andere Stelle ziehen.



Der Informationsfluss in Diagrammen wird durch Verbinden der Elemente im Zeichenbereich bestimmt.

### Herstellen einer Verbindung zwischen Diagrammelementen:

- Klicken Sie mit der rechten Maustaste in den Zeichenbereich, jedoch *nicht* auf irgendein Element,

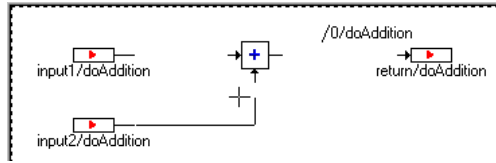
oder



- klicken Sie auf die Schaltfläche **Connect**.  
Der Verbindungsmodus wird gestartet und der Cursor verwandelt sich in ein Fadenkreuz.
- Um eine Verbindung zu beginnen, klicken Sie auf den Anschluss des ersten Diagrammelements, das Sie verbinden möchten.
- Um eine Verbindung zu erzeugen, bewegen Sie den Cursor zum Endpunkt der Verbindung.

Eine Linie folgt dem Weg der Bewegung Ihres Cursors. Jedesmal, wenn Sie in den Zeichenbereich klicken, jedoch nicht auf ein Element, erzeugen Sie einen Verankerungspunkt für die Verbindungslinie. Durch Klicken mit der rech-

ten Maustaste während des Ziehens einer Verbindungslinie wird der letzte Verankerungspunkt gelöscht.



- Um die Verbindung fertigzustellen, klicken Sie auf einen Anschluss des Diagrammelements, das Sie verbinden möchten.

Die Verbindung wird hergestellt und es wird eine Linie gezeichnet.

Das Aussehen der Linie richtet sich nach den verbundenen Elementen.

Farbe/Stil	Bedeutung
schwarz/durchgezogen	Verbindung zwischen zwei numerischen Anschlüssen
schwarz/gestrichelt	Verbindung zwischen zwei logischen Anschlüssen
schwarz/Strich-Punkte	Kontrollfluss-Verbindung (siehe Seite 225)
farbig (Standard: grün)	Kommentarlinie; die Sequenzierung für die Anweisung oder Operation ist noch nicht gelöst (s. Kapitel 4.1.6). Die Farbe dieser Linien kann im ASCET-Optionsfenster gewählt werden (s. „Knoten „Colors““ auf Seite 57).
rot	fehlerhafte Verbindung, z.B. zwischen numerischen und logischen Anschlüssen

Wenn Sie ein Element hin- und herbewegen, folgt die Verbindungslinie dieser Bewegung. Es kann jedoch erforderlich sein, den Verlauf einer Linie zu ändern, damit das Diagramm klar und übersichtlich bleibt. Sie können dies erreichen, indem Sie einfach die Linie ziehen.

### Verbindungsmodus beenden:

Der Verbindungsmodus ist nach Fertigstellung einer Verbindung beendet. Er kann aber auch vorher beendet werden, und zwar wie folgt.

- Klicken Sie nochmals mit der rechten Maustaste in den Zeichenbereich (nicht auf ein Element)

oder

- klicken Sie erneut auf die Schaltfläche **Connect**.

Der Verbindungsmodus wird beendet.

### Arrays, Matrizen, Kennlinien und -felder

In diesem Abschnitt ist beschrieben, wie Sie Arrays, Matrizen und Kennlinien/-felder anlegen. Wie Sie die Daten dieser Elemente bearbeiten, ist in Kapitel 4.11 „Bearbeiten von Daten“ beschrieben.

#### Erzeugen eines Arrays oder einer Matrix:

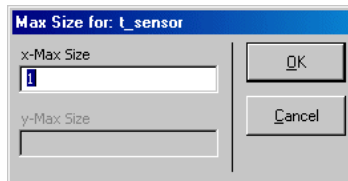


- Klicken Sie auf die Schaltfläche **Array** oder **Matrix**.

Der Elementeditor wird geöffnet. Da Arrays und Matrizen als Variable angelegt werden, ist das Attribut **volatile** eingestellt.

- Passen Sie die Elementeigenschaften nach Ihren Wünschen an und drücken Sie **OK**.

Das Dialogfenster „Max size“ wird geöffnet.



- Passen Sie die maximale Anzahl von Elementen in den Feldern „x-Max Size“ und „y-Max Size“ an.

Das Feld „y-Max Size“ ist nur verfügbar, wenn eine Matrix angelegt wurde.

- Klicken Sie **OK**.

Wenn Sie die maximale Größe überschritten haben, wird folgende Fehlermeldung angezeigt:

\*-Max size exceeded. Reduce to limit <limit>.

- Übernehmen Sie das Limit mit **OK**.

Oder

- Kehren Sie mit **Cancel** zum Fenster „Max size“ zurück.
- Platzieren Sie das Element im Zeichenbereich.

### Erzeugen von normalen oder Festkennlinien/-feldern:



- Wählen Sie im Kombikästchen in der Schaltflächenleiste den Tabellentyp **Normal** oder **Fixed**.
- Klicken Sie auf die Schaltfläche **One D Table Parameter** oder **Two D Table Parameter**.  
Der Elementeditor wird geöffnet. Da alle Kennlinien/-felder als Parameter angelegt werden, ist das Attribut **non-volatile** eingestellt und das Feld „Memory“ deaktiviert.
- Passen Sie die Elementeneigenschaften nach Ihren Wünschen an und drücken Sie **OK**.  
Das Dialogfenster „Max size“ wird geöffnet.
- Passen Sie die maximale Anzahl von Abtastpunkten in den Feldern „x-Max Size“ und „y-Max Size“ an.

#### Hinweis

*Festkennlinien/-felder müssen auf jeder Achse mindestens 2 Stützstellen haben. Andernfalls kommt es bei der Überprüfung der Monotonie zu einer Fehlermeldung.*

Das Feld „y-Max Size“ ist nur verfügbar, wenn ein Kennfeld angelegt wurde.

- Klicken Sie **OK**.

Wenn Sie die maximale Größe überschritten haben, wird folgende Fehlermeldung angezeigt:

\*-Max size exceeded. Reduce to limit <limit>.

- Übernehmen Sie das Limit mit **OK**.

*Oder*

- Kehren Sie mit **Cancel** zum Fenster „Max size“ zurück.

- Platzieren Sie das Element im Zeichenbereich.

### Erzeugen von Distributionen:

---

Distributionen werden für die Verwendung von Gruppenkennlinien und -feldern (siehe „Gruppentabellen und Verteilungen“ auf Seite 502) benötigt.



- Klicken Sie auf die Schaltfläche **Distribution**.

Der Elementeditor wird geöffnet. Da Distributionen als Parameter angelegt werden, ist das Attribut **non-volatile** eingestellt und das Feld „Memory“ deaktiviert.

- Passen Sie die Elementeigenschaften nach Ihren Wünschen an und drücken Sie **OK**.

Das Dialogfenster „Max size“ wird geöffnet.

- Passen Sie die maximale Anzahl von Stützstellen im Feld „x-Max Size“ an.

Das Feld „y-Max Size“ ist nicht verfügbar.

- Klicken Sie **OK**.

Wenn Sie die maximale Größe überschritten haben, wird folgende Fehlermeldung angezeigt:

x-Max size exceeded. Reduce to limit 2048.

- Übernehmen Sie das Limit mit **OK**.

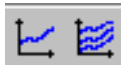
Oder

- Kehren Sie mit **Cancel** zum Fenster „Max size“ zurück.

- Platzieren Sie das Element im Zeichenbereich.

### Erzeugen von Gruppenkennlinien/-feldern:

---



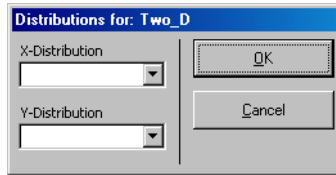
- Wählen Sie im Kombikästchen in der Schaltflächenleiste den Tabellentyp **Group**.

- Klicken Sie auf die Schaltfläche **One D Table Parameter** oder **Two D Table Parameter**.

Der Elementeditor wird geöffnet. Da alle Kennlinien/-felder als Parameter angelegt werden, ist das Attribut **non-volatile** eingestellt und das Feld „Memory“ deaktiviert.



- Passen Sie die Elementeigenschaften nach Ihren Wünschen an und drücken Sie **OK**.  
Das Fenster „Distribution for“ wird geöffnet.



- Wählen Sie in den Kombikästchen „x-Distribution“ und „y-Distribution“ geeignete Distributionen.

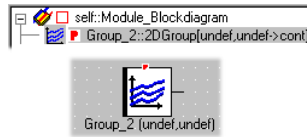
Das Kombikästchen „y-Distribution“ ist nur für Gruppenkennfelder verfügbar.

- Klicken Sie **OK**.

Wenn Sie auf **Cancel** klicken, wird keine Distribution ausgewählt. Die Tabelle wird dennoch angelegt, aber mit `undef` gekennzeichnet.

Die Distribution muss dann vor dem Bearbeiten der Tabelle mit **Element** → **Edit Distribution** hinzugefügt werden.

- Platzieren Sie das Element im Zeichenbereich.



## Kontrollflussanweisungen

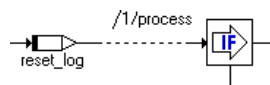
### If-Abfragen verwenden:



- Klicken Sie auf die Schaltfläche **If then** oder **If then else**, um den Mauszeiger mit dem entsprechenden Block zu laden.

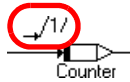
- Platzieren Sie den Block im Zeichenbereich.

Der If...Then-Block und der If...Then...Else-Block sehen fast gleich aus, nur hat der If...Then...Else-Block zwei Kontrollflusszweige.



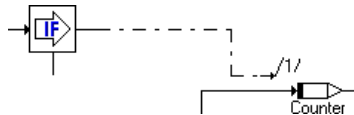
- Verbinden Sie den Eingang mit einem logischen Element.
- Spezifizieren Sie die Aktionen für die Zweige.

- Klicken Sie mit der rechten Maustaste auf den Sequenzaufruf, der mit einem Zweig verbunden werden soll, und wählen Sie aus dem Kontextmenü **Connector**.



Der Sequenzaufruf wird zu einem Verbinder. Die farbige Linie einer ungelösten Sequenzierung wird nun schwarz.

- Verbinden Sie den gewünschten Zweig mit dem Verbinder.



Sie können einen Zweig auch mit mehreren Aktionen verbinden. Bearbeiten Sie in dem Fall die Nummern der Verbinder (siehe Seite 249); wie bei Sequenzaufrufen darf keine Nummer doppelt vorkommen.

- Verfahren Sie für den anderen Zweig des If...Then...Else-Blocks ebenso.

### Schalter verwenden:

- Wählen Sie aus dem Kombikästchen „Argument Size“ die Anzahl der Verzweigungen des Schalters aus.

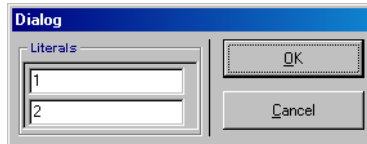
Sie müssen mindestens zwei Verzweigungen wählen. Der Default-Zweig wird *nicht* mitgezählt.



- Klicken Sie auf die Schaltfläche **Switch**, um den Mauszeiger mit dem entsprechenden Block zu laden.
- Platzieren Sie den Block im Zeichenbereich.

- Wenn Sie die Werte für die Alternativen bearbeiten wollen, klicken Sie mit der rechten Maustaste auf den Block und wählen Sie **Edit Literals** aus dem Kontextmenü.

Ein Editorfenster öffnet sich; es enthält ein Eingabefeld für jede Verzweigung.



- Geben Sie im Feld „Literals“ die Werte für die Verzweigungen an.
  - Schließen Sie den Editor mit **OK**, um die Änderungen zu übernehmen.
- Verbinden Sie den Eingang oben am Block mit einem Element vom Typ `sdisc` oder `udisc`.

### Hinweis

*Wenn Sie den Eingang mit einem Anschluss vom Typ `cont` verbinden, wird bei der Codegenerierung ein Fehler angezeigt.*

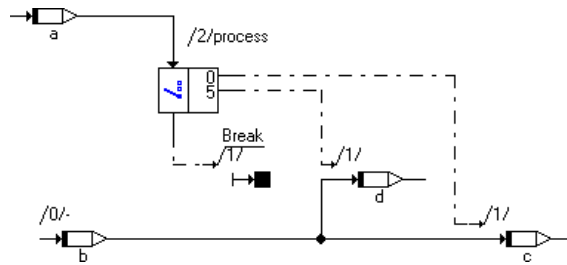
- Spezifizieren Sie die Aktionen für die Zweige.
- Klicken Sie mit der rechten Maustaste auf den Sequenzaufruf, der mit einem Zweig verbunden werden soll, und wählen Sie aus dem Kontextmenü **Connector**.

Der Sequenzaufruf wird zu einem Verbinder.

- Verbinden Sie den gewünschten Zweig mit dem Verbinder.

Sie können einen Zweig auch mit mehreren Aktionen verbinden. Bearbeiten Sie in dem Fall die Nummern der Verbinder (siehe Seite 249); wie bei Sequenzaufrufen darf keine Nummer doppelt vorkommen.

- Verfahren Sie für die anderen Zweige ebenso.

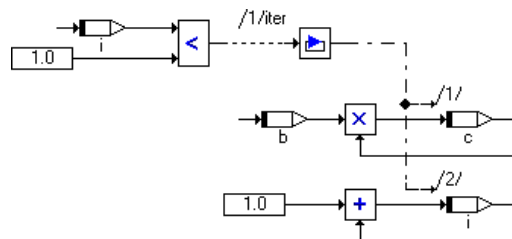
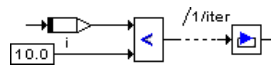


### While-Schleife verwenden:

Das einzige Schleifenkonstrukt, das in Blockdiagrammen zur Verfügung steht, ist die `while`-Schleife.



- Klicken Sie auf die Schaltfläche **While**, um den Mauszeiger mit dem entsprechenden Block zu laden.
- Platzieren Sie den Block im Zeichenbereich.
- Spezifizieren Sie die Bedingung für die Schleife.
- Verbinden Sie die Bedingung mit dem Eingang der Schleife.
- Spezifizieren Sie die Aktion für die Schleife.
- Klicken Sie mit der rechten Maustaste auf den Sequenzaufruf, der mit dem Ausgang der Schleife verbunden werden soll, und wählen Sie aus dem Kontextmenü **Connector**.  
Der Sequenzaufruf wird zu einem Verbinder.
- Verbinden Sie den Ausgang der Schleife mit dem Verbinder.



Sie können den Ausgang auch mit mehreren Aktionen verbinden. Bearbeiten Sie in dem Fall die Nummern der Verbinder (siehe Seite 249); wie bei Sequenzaufrufen darf keine Nummer doppelt vorkommen.

### **Hinweis**

*Achten Sie unbedingt darauf, unendliche oder für Echtzeitanwendungen ungeeignete Schleifen zu vermeiden!*

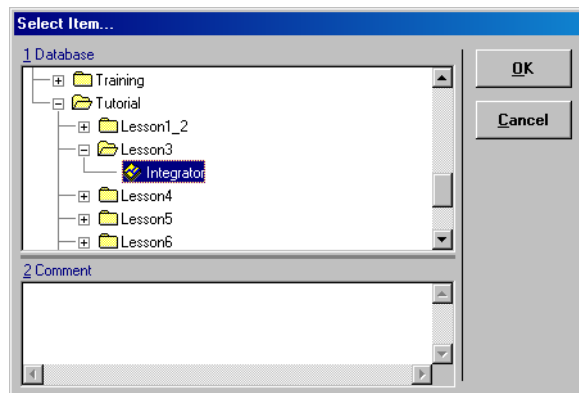
### Komponenten als komplexe Elemente

Komponenten können als komplexe Elemente innerhalb anderer Komponenten verwendet werden. Während Basiselemente in der Komponente definiert sind, in der sie enthalten sind, werden komplexe Elemente durch Referenz aufgenommen, d.h. wenn die aufgenommene Komponente geändert wird, so sind diese Änderungen in der aufnehmenden Komponente wirksam. Sie können andere Diagrammelemente mit den Eingängen und Ausgängen der aufgenommenen Komponente verbinden.

### **Aufnehmen einer Komponente als komplexes Element:**

Um eine andere Komponente aufzunehmen, gehen Sie wie folgt vor.

- Wählen Sie **Element** → **Add Item**  
oder
- wählen Sie aus dem Kontextmenü **Add Item**.  
Das Dialogfenster „Select Item“ öffnet sich. Es zeigt den Inhalt der aktuellen Datenbank.



- Wählen Sie aus der Liste „1 Database“ die Komponente aus, die Sie hinzufügen möchten.
- Klicken Sie auf **OK**, um die Komponente hinzuzufügen.

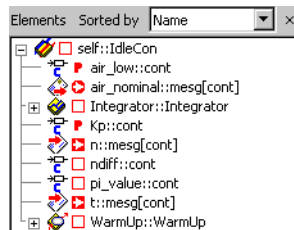
Die Komponente wird zur Liste „Elements“ hinzugefügt, wobei ihr Instanzname hervorgehoben wird, um ein Bearbeiten an Ort und Stelle zu ermöglichen.

- Bearbeiten Sie den Instanznamen und drücken Sie <ENTER>.
- Ziehen Sie die Komponente in den Zeichenbereich, um sie zum Diagramm hinzuzufügen.
- Verbinden Sie die Anschlüsse der Komponente genauso wie die Anschlüsse anderer Diagrammelemente.

### Hinweis

Als Alternative zum Hinzufügen von Datenbankeinträgen unter Verwendung der hier beschriebenen Menüauswahl können Sie Einträge vom Komponentenmanager zum Blockdiagrammeditor ziehen (Drag & Drop).

Wenn Sie eine Komponente aufnehmen, wird deren Instanzname in der Fläche „Elements“ dargestellt. Das Pluszeichen vor dem Instanznamen (⊢ ⊕) signalisiert, dass die Baumstruktur weiter aufgeklappt werden kann.



Wenn Sie dieselbe Komponente nochmals aufnehmen, wird eine neue Instanz der Komponente erzeugt, und es wird ein eindeutiger Instanzname zugewiesen. Im Diagramm wird der Instanzname unterhalb des grafischen Blockes angezeigt. In der Liste „Elements“ werden sowohl der Instanzname als auch der Klassenname im folgenden Format angezeigt: <instance-Name>::<className>. Sie können die Breite der Liste ändern, um ihren gesamten Inhalt zu betrachten.

## Kommentare und Notizen

---

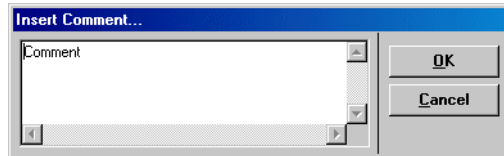
Kommentare haben keinerlei Einfluss auf die Funktionalität einer Komponente. Sie können nur erläuternden Text enthalten, der helfen kann, Ihr Softwaremodell zu dokumentieren.

### Hinzufügen eines Kommentars:

---



- Klicken Sie auf die Schaltfläche **Comment**.  
Es erscheint ein Textfeld.



- Geben Sie den Text für Ihren Kommentar ein.
- Klicken Sie auf **OK**.  
Der Mauszeiger wird mit dem Kommentar geladen.
- Klicken Sie in den Zeichenbereich, um den Kommentar darin anzuordnen.

Sie können die Position des Kommentars später korrigieren, indem Sie ihn an eine andere Stelle ziehen. Sie können den Text des Kommentars bearbeiten, indem Sie mit der rechten Maustaste darauf klicken und aus dem Kontextmenü **Edit Comment** wählen.

Sie können auch an die gesamte Komponente oder an eine aufgenommene Komponente eine Notiz anhängen. Die Notizen werden in einem separaten Editorfenster eingegeben. Bei der automatischen Erzeugung einer Dokumentation werden Notizen mit in die Dokumentation eingefügt.

### Bearbeiten der Notizen einer Komponente:

---

- Wenn Sie die Notizen der editierten Komponente bearbeiten wollen, wählen Sie **Component** → **Notes**.

*Oder*

- Markieren Sie in der Liste „Elements“ oder in der Zeichenfläche die aufgenommene Komponente, deren Notiz Sie bearbeiten wollen.

- Wählen Sie aus dem Kontextmenü dieser Komponente **Notes**.

Der Notizeneditor für den gewählten Eintrag wird geöffnet. Einzelheiten dazu siehe „Notizen“ auf Seite 747.

#### 4.1.5 Bearbeiten von Blockdiagrammen

---

Im vorliegenden Abschnitt werden die allgemeinen Bearbeitungsmerkmale des Zeichenbereichs sowie die Merkmale, die das Aussehen von Diagrammelementen ändern, beschrieben.

In manchen Fällen ist es von Nutzen, durchgeführte Aktionen im Diagramm zurückzuverfolgen oder in eine frühere Version des Diagramms zurückzukehren, um dort die Arbeit erneut aufzunehmen. Dazu werden die durchgeführten Aktionen im Diagramm Schritt für Schritt in der internen Zwischenablage aufgezeichnet. Die Möglichkeit einer Rückkehr in eine frühere Diagrammversion besteht allerdings nur, solange der Benutzer sich im geöffneten Editor befindet. Bei Verlassen des Editors wird nur die zuletzt bearbeitete Version abgespeichert, die aufgezeichneten Versionen in der Zwischenablage werden gelöscht.

##### **Rückgängigmachen der letzten Aktion:**

---

- Wählen Sie **View → Undo**

oder



- klicken Sie auf die Schaltfläche **Undo**

oder

- Drücken Sie <CTRL> + <Z>.

Die zuletzt durchgeführte Aktion wird rückgängig gemacht.

##### **Hinweis**

*Bei einer Wiederaufnahme der Arbeit in einer früheren Version des Diagramms werden alle später vorgenommene Aktionen gelöscht und die neuen Aktionen aufgezeichnet.*



## Wiederherstellen einer rückgängig gemachten Aktion:

---

- Wählen Sie **View** → **Redo**

oder



- klicken Sie auf die Schaltfläche **Redo**

oder

- Drücken Sie <CTRL> + <Y>.

Die zuletzt rückgängig gemachte Aktion wird erneut ausgeführt.

## Elemente betrachten

---

### Ansehen aller Vorkommen eines Elements:

---

- Wählen Sie entweder im Zeichenbereich oder in der Fläche „Elements“ das Element aus.

- Wählen Sie **Element** → **Show Occurrences**

oder

- wählen Sie **Show Occurrences** aus dem Kontextmenü.

Alle Vorkommen des Elements in dem aktuellen Programm werden hervorgehoben.

### Ansehen aller mit einem Objekt verbundenen Elemente:

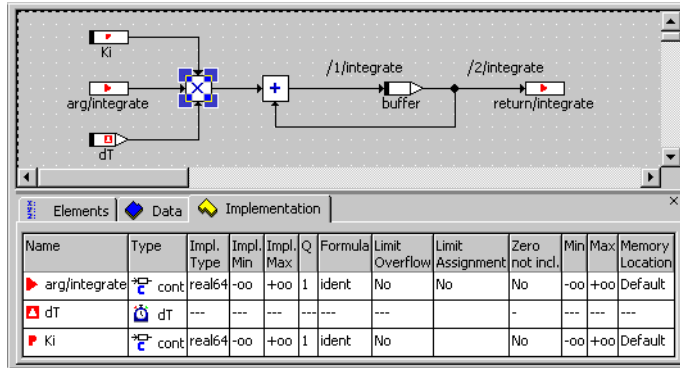
---

Sie brauchen nicht in die Browser-Ansicht zu wechseln, wenn Sie die Implementierungen der direkt mit einem grafischen Objekt (Element, Operator, Verbindung) verbundenen Elemente ansehen oder bearbeiten wollen. Von besonderem Interesse ist diese Funktion bei der Überprüfung der Implementierungen der Ein- und Ausgänge eines Operators, daher ist nur dieser Fall im Folgenden beschrieben.

- Klicken Sie mit der rechten Maustaste auf ein grafisches Objekt und wählen Sie aus dem Kontextmenü **Browse Connected Elements**.

Ein Feld mit den Registern „Elements“, „Data“ und „Implementation“ wird eingeblendet. In diesem Feld werden alle direkt mit dem grafischen Objekt verbundenen Elemente aufgelistet.

Wenn das grafische Objekt ein Element ist, wird es mit angezeigt. Operatoren sind in der Liste nicht enthalten.



Die Register haben gleiche Funktionalität und Aufbau wie die des Komponentenmanagers (s. Kapitel 2.1.3). Das Register „Implementation“ ist standardmäßig aktiviert.

- Arbeiten Sie in den Registern wie in Kapitel 2.3.2 beschrieben.

Insbesondere können zur Bearbeitung der Elementeigenschaften per Doppelklick die jeweiligen Editoren geöffnet werden.

- Klicken Sie auf das **X** rechts oben im Feld



oder

- wählen Sie **View** → **Show/ Hide** → **Connected Elements**, um das Feld auszublenden.
- Sie können das Feld außer über das Kontextmenü auch mit **View** → **Show/ Hide** → **Connected Elements** wieder einblenden, allerdings wird dann der Inhalt nicht an die aktuelle Auswahl im Zeichenbereich angepasst.

### Umbenennen oder Löschen eines Diagrammelements:

---

- Wählen Sie **Element** → **Rename**, um das Element umzubenennen.
- Wählen Sie **Element** → **Delete**, um das Element zu entfernen.

Bevor Sie ein Element löschen, müssen Sie sein Vorkommen aus dem Diagramm löschen. Sie können den Befehl **Show Occurences** verwenden, um die zu entfernenden Vorkommen im Diagramm auswählen zu lassen.

- Wählen Sie **Component** → **Delete Unused Elements**, um alle Elemente zu löschen, die in der Fläche „Elements“ aufgelistet sind, jedoch in einem Diagramm keine Vorkommen haben.

### Ersetzen eines Diagrammelements:

---

- Wählen Sie einen Operator aus der Symbolleiste aus, um den Mauszeiger mit einem neuen Diagrammelement zu laden.
- Klicken Sie auf einen vorhandenen Operator im Diagramm, um ihn durch den neuen Operator zu ersetzen.

Es wird ein Dialogfenster mit einer Bestätigungsaufforderung angezeigt.

- Klicken Sie auf **Yes**, um das Diagrammelement zu ersetzen.

Auf dieselbe Weise können Sie auch ein Element ersetzen. In dem Fall müssen Sie das Element auf das zu ersetzende Diagrammelement ziehen. Sie können auch einen Operator im Diagramm durch ein Element ersetzen und umgekehrt.

### Ausschneiden, Kopieren und Einfügen eines Diagrammelements:

---

- Klicken Sie auf das auszuschneidende oder zu kopierende Diagrammelement.

Ein ausgewähltes Element wird durch um seinen Rand herum angeordnete Anfasser gekennzeichnet.

- Wählen Sie **Edit** → **Cut**

*oder*

- drücken Sie <CTRL> + <X>, um das ausgewählte Diagrammelement zu entfernen und in der Zwischenablage abzulegen.

- Wählen Sie **Edit** → **Copy**

*oder*

- drücken Sie <CTRL> + <C>, um das ausgewählte Diagrammelement in die Zwischenablage zu kopieren, ohne es aus dem Zeichenbereich zu entfernen.

- Wählen Sie **Edit** → **Paste**

*oder*

- drücken Sie <CTRL> + <V>, um das Diagrammelement in der Nähe der bisherigen Position einzufügen.
- Ziehen Sie das neue Diagrammelement an die gewünschte Stelle.

ASCET verwendet eine eigene interne Zwischenablage für grafische Informationen, so dass diese Operationen keinen Einfluss auf die Zwischenablage von Windows haben. Sie können auch Diagrammelemente zwischen Komponenten kopieren, doch es werden nur die grafischen Informationen kopiert, und Schnittstellenelemente sind ausgeschlossen.

#### **Löschen einer Verbindung oder eines Elements:**

---

- Wählen Sie die Verbindung oder das Element im Zeichenbereich.

- Wählen Sie **Edit** → **Delete**

*oder*

- drücken Sie <DEL>, um die ausgewählte Verbindung bzw. das ausgewählte Element aus dem Diagramm zu löschen.

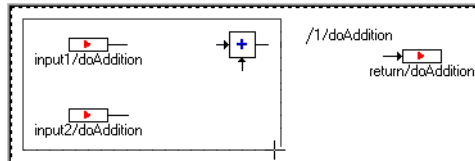
#### **Auswählen mehrerer Diagrammelemente:**

---

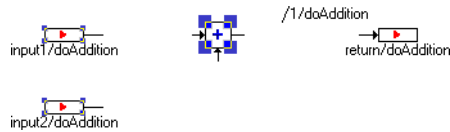
- Klicken Sie auf alle Diagrammelemente, die Sie auswählen möchten, und halten Sie dabei die Taste <CTRL> gedrückt.

*Oder*

- Ziehen Sie ein Rechteck um die Elemente herum, die Sie im Zeichenbereich auswählen möchten.



Die Diagrammelemente innerhalb des Rechtecks werden ausgewählt, was durch Anfassers an ihren Ecken angezeigt wird.



- Sie können die Gruppe genauso ausschneiden, kopieren, einfügen, löschen oder bewegen wie ein einzelnes Diagrammelement.

### *Aussehen von Diagrammelementen*

#### **Ändern des Aussehens eines Diagrammelements:**

Diese Befehle stehen nur über das Kontextmenü eines Vorkommens einer aufgenommenen Komponente zur Verfügung, die im Zeichenbereich ausgewählt wurde. Sie beeinflussen nur das Aussehen des jeweiligen Vorkommens.

#### **Hinweis**

*Anders als die im Abschnitt „Layout von aufgenommenen Komponenten“ beschriebenen Einstellungen können diese nicht ins Standard-Layout übernommen werden.*

- Wählen Sie **Show/Hide Name**, um den Instanznamen des Vorkommens zu zeigen oder zu verbergen.

- Wählen Sie **Ports** → **Get/Set**, um die Get- und Set-Anschlüsse des Vorkommens zu zeigen oder zu verbergen.

Dieser Befehl ist unter dem Namen **Get/ Set Ports** auch für Arrays, Matrizen und Kennlinien/-felder verfügbar.

### **Bearbeiten der Sichten eines Diagrammelements:**

---

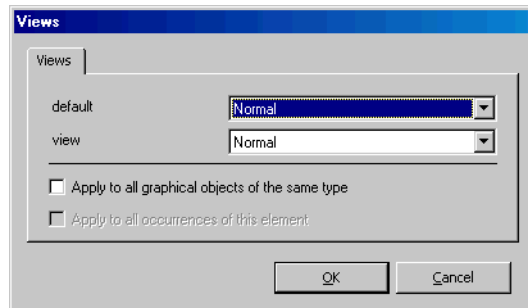
Sie können die Darstellung einzelner Diagrammelemente (Basiselemente, aufgenommene Komponenten, Operatoren, Verbindungslinien usw.) in den verschiedenen Sichten (siehe Kapitel 7.3) bearbeiten.

- Wählen Sie das Element, dessen Darstellung Sie bearbeiten wollen.
- Wählen Sie **Edit** → **Views**

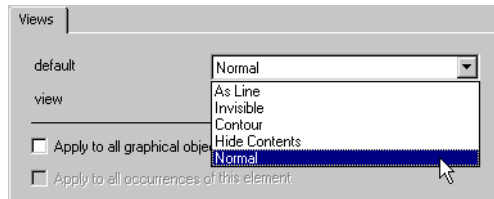
oder

- klicken Sie mit der rechten Maustaste auf das Diagrammelement und wählen Sie aus dem Kontextmenü **Views**.

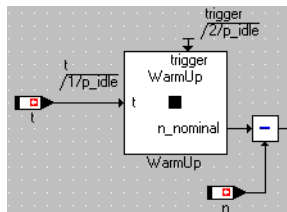
Das Dialogfenster „Views“ wird geöffnet. Es zeigt alle in der aktuellen Datenbank definierten Sichten.



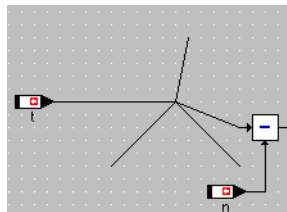
- Öffnen Sie das Kombikästchen der gewünschten Sicht.



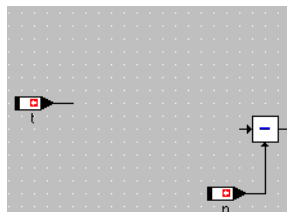
- Wählen Sie aus dem Kombikästchen die gewünschte Darstellung für das Diagrammelement.



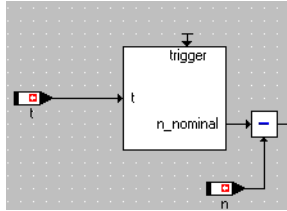
- Normal - die Standardeinstellung, bei der alle vorandenen Elemente wie Symbole, Namen, Sequenzaufrufe etc. angezeigt werden.



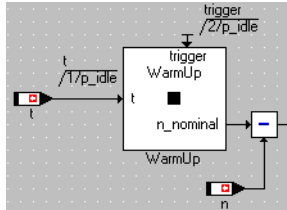
- As Line - Das Element wird durch eine oder mehrere Linien von der Mitte zu den Pins ersetzt.



- Invisible - Das Element verschwindet aus der grafischen Darstellung und kann im Zeichenbereich nicht mehr ausgewählt werden. Es ist aber weiterhin in der Liste „Elements“ vorhanden.



- **Contour** - Nur die Umrisse des Elements und ggf. die Namen der Anschlüsse werden dargestellt. Sequenzaufrufe, Name, Symbol etc. verschwinden.



- **Hide Content** - Die Bildschirmdarstellung ist gegenüber **Normal** unverändert, aber die Komponente oder Hierarchie kann aus dem Zeichenbereich heraus nicht mehr geöffnet werden.

Der Inhalt einer so markierten Hierarchie ist in der Dokumentation nicht enthalten, der Inhalt einer Komponente ist in der Dokumentation enthalten.

- Aktivieren Sie die Option **Apply to all graphical objects of the same type**, wenn die Auswahl für alle Diagrammelemente dieses Typs gelten soll.
- Aktivieren Sie die Option **Apply to all occurrences of this element**, wenn die Auswahl für alle Vorkommen dieses Elements gelten soll.
- Klicken Sie **OK** um Ihre Auswahl zu übernehmen.

Das Element wird in der gewählten Sicht entsprechend Ihrer Auswahl dargestellt.

Der Befehl steht für alle Diagrammelemente zur Verfügung. Sichten sind in Kapitel 7.3 auf Seite 736 erläutert.

Nicht immer ist im Darstellungsmodus *As Line* zu erkennen, dass es sich um ein Element handelt. Daher wird folgende Warnung angezeigt, wenn Sie ein Element im Modus *As Line* oder eine Verbindung zu einem solchen Element auswählen:

CAUTION: Element shown in "As Line" view could be affected!



## *Layout von aufgenommenen Komponenten*

---

Wenn Sie in Ihre Komponente eine andere Komponente aufnehmen (s. Seite 229) und im Zeichenbereich ablegen, wird das im Layout-Editor definierte Standard-Layout der aufgenommenen Komponente angezeigt. Wenn dieses Standard-Layout Ihren Bedürfnissen nicht entspricht, haben Sie zwei Möglichkeiten.

Zum einen können Sie das Standard-Layout im Layout-Editor anpassen (s. Kapitel 4.13). Änderungen im Layout-Editor haben jedoch keinen Einfluss auf individuelle Vorkommen der Komponente in Blockdiagrammen. Existierende Diagramme bleiben unverändert; Sie müssen die Vorkommen manuell ersetzen, um das geänderte Layout zu laden.

Zum anderen können Sie im Blockdiagramm das Layout eines bestimmten Vorkommens anpassen, wenn Sie entweder im ASCET-Optionsfenster die Option **Activate flexible layout** aktiviert haben (s. „Optionen für Blockdiagramme“ auf Seite 56) oder das flexible Layout für diese Komponente einzeln aktiviert haben (siehe „Flexibles Layout aktivieren (lokal):“ auf Seite 553). Dabei wird nur das Layout des bearbeiteten Vorkommens geändert, weder das Standard-Layout noch das Layout weiterer Vorkommen derselben Komponente im Diagramm werden automatisch angepasst. Sie können also bei Bedarf jedem Vorkommen derselben Komponente unterschiedliche Layouts zuweisen.

Sie können die Größe eines Blocks ändern, die Anschlüsse verschieben sowie anzeigen oder verstecken. Dabei sind folgende Grenzen zu beachten:

- Ein Vorkommen muss mindestens die Größe eines Additionsoperators mit zwei Eingängen haben.
- Darüber hinaus wird die Mindestgröße eines Vorkommens durch die Anzahl der sichtbaren Anschlüsse begrenzt: zwei Anschlüsse können nicht dieselbe Position einnehmen.
- Wenn das Standard-Layout eine Symbolgrafik enthält, wird diese abgeschnitten, falls die Größe des Vorkommens kleiner wird als die Grafik.

Wenn Sie ein so bearbeitetes Vorkommen wie auf Seite 235 beschrieben kopieren oder ausschneiden und an anderer Stelle einfügen, erhält das eingefügte Vorkommen das Layout des kopierten/ausgeschnittenen Vorkommens.

### **Größe eines Vorkommens bearbeiten:**

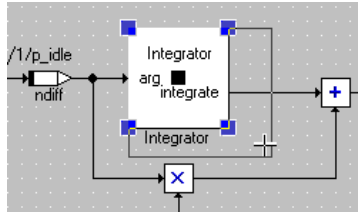
---

Sie können die Größe eines Vorkommens entweder manuell bearbeiten oder automatisch die minimale Größe einstellen.

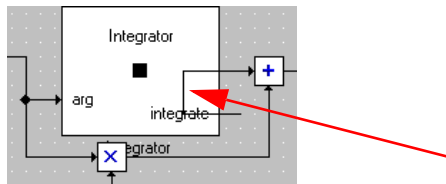
- **Markieren Sie im Zeichenbereich das Vorkommen der aufgenommenen Komponente, das Sie bearbeiten wollen.**

An den Ecken erscheinen vier Anfasser.

- Ziehen Sie an den Anfassern, um die gewünschte Größe für das Vorkommen einzustellen.

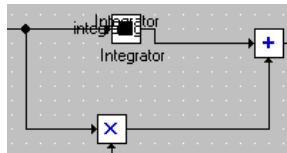


Andere Elemente werden nicht mit verschoben. Existierende Verbindungslinien bleiben erhalten, haben aber möglicherweise einen unübersichtlichen Verlauf.



- Ordnen Sie bei Bedarf die Diagrammelemente neu an, damit die Übersichtlichkeit gewahrt bleibt.
- Klicken Sie mit der rechten Maustaste auf das Vorkommen und wählen Sie aus dem Kontextmenü **Minimal Size**.

Die Größe des Vorkommens wird entsprechend der Anzahl der Anschlüsse minimiert. Die Namen der Anschlüsse oder des Vorkommens werden dabei ignoriert.

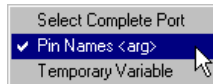
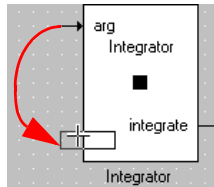


- Wählen Sie **Show/Hide Name** aus dem Kontextmenü des Vorkommens, um den Namen der Komponente ein- oder auszublenden.

### Anschlüsse bearbeiten:

Sie können die Anschlüsse eines Vorkommens verschieben sowie die Namen ein- oder ausblenden.

- Ziehen Sie den Anschluss, den Sie verschieben wollen, mit der Maus an die gewünschte Position.



Sie können den Anschluss überall dort ablegen, wo noch kein anderer Anschluss ist.

- Klicken Sie mit der rechten Maustaste auf den Anschluss und wählen Sie **Pin Names <name>** aus dem Kontextmenü, um die Anzeige des Anschlussnamens ein- oder auszuschalten.

Wenn der Anschlussname angezeigt wird, ist **Pin Names <pin name>** mit einem Haken markiert.

- Wählen Sie aus dem Kontextmenü des Vorkommens **Show Pin Names**, um die Namen aller Anschlüsse des Vorkommens zu zeigen.
- Wählen Sie **Hide Pin Names**, um die Namen aller Anschlüsse des Vorkommens zu verbergen.

- Wählen Sie **Unconnected Ports**, um alle nicht verbundenen Anschlüsse des Vorkommens zu zeigen oder zu verbergen. Die Anschlüsse einer Methode können nur gemeinsam angezeigt oder versteckt werden.

### Hinweis

Auf diese Weise werden Anschlüsse nur verborgen, nicht entfernt. Operationen wie **Minimal Size** verhalten sich, als seien die Anschlüsse sichtbar.

### Ports anzeigen/verstecken:

Sie haben die Möglichkeit, die Darstellung einer Komponente über das Kontextmenü **Ports** zu modifizieren. Anschlüsse einer Komponente könne hier methodenweise hinzugefügt oder entfernt werden.

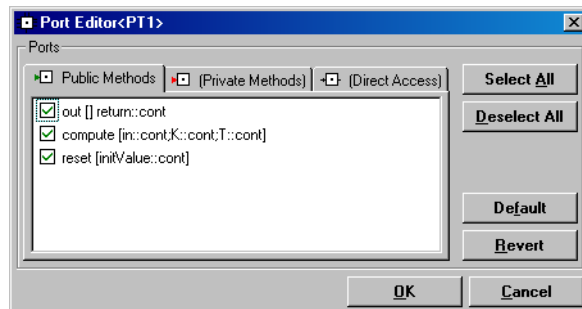
### Hinweis

Die Anschlüsse einer Methode können der Darstellung nur gemeinsam hinzugefügt oder entfernt werden.

#### 1. Öffentliche Methoden

- Wählen Sie aus dem Kontextmenü des Vorkommens **Ports** → **Methods**.

Der Port-Editor öffnet sich; das Register „Public Methods“ wird angezeigt. Es enthält eine Liste aller öffentlichen Methoden der aufgenommenen Komponente. Die ausgewählten Methoden sind mit einem Häkchen markiert.



## 2. Private Methoden

- Wählen Sie aus dem Kontextmenü des Vorkommens **Ports** → **Methods**.
- Wählen Sie das Register „Private Methods“.

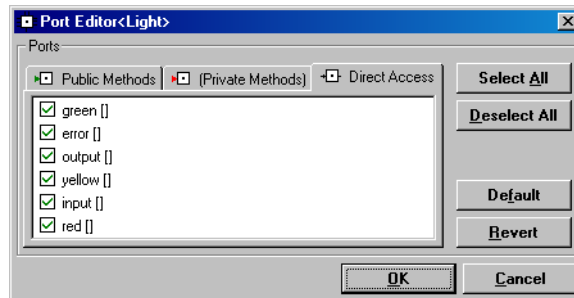
## 3. Direct Access

- Wählen Sie aus dem Kontextmenü des Vorkommens **Ports** → **Direct Access**.

Der Port-Editor öffnet sich; das Register „Direct Access“ wird angezeigt. Es enthält eine Liste aller Direktzugriffsmethoden der aufgenommenen Komponente (z.B. die Ein-/Ausgänge eines Zustandsautomaten).

*Oder*

- Wählen Sie aus dem Kontextmenü des Vorkommens **Ports** → **Methods**.
- Wählen Sie das Register „Direct Access“.



## 4. Bearbeiten

- Doppelklicken Sie auf einen nicht markierten Eintrag, um diesen auszuwählen.
- Klicken Sie auf **Select All**, um alle Einträge auszuwählen.
- Doppelklicken Sie auf einen markierten Eintrag, um die Auswahl dieses Eintrags aufzuheben.
- Klicken Sie auf **Deselect All**, um die Auswahl aller Einträge aufzuheben.

- Klicken Sie auf **Revert**, um sämtliche bisher gemachten Änderungen zu widerrufen.
- Klicken Sie auf **Default**, um die im Layout-Editor vorgegebene Einstellung wiederherzustellen.
- Wechseln Sie in eins der anderen Register, um weitere Einstellungen vorzunehmen,

*oder*

- klicken Sie auf **Cancel**, um den Vorgang abbrechen,

*oder*

- übernehmen Sie die Auswahl mit **OK**.

Die Anschlüsse der markierten Methoden/Prozesse werden im Vorkommen angezeigt, die der nicht markierten Methoden/Prozesse werden entfernt.

Wenn Sie einen Anschluss entfernen, der mit einem anderen Element verbunden ist, wird die Verbindungslinie mit entfernt.

Die Positionen für neu hinzugefügte Anschlüsse werden automatisch bestimmt. Eingänge werden auf der linken Seite angelegt, Ausgänge auf der rechten Seite. Wenn Sie Anschlüsse hinzufügen wollen, für die die aktuelle Größe des Vorkommens keinen Platz bietet, wird das Layout automatisch vergrößert.

### **Änderungen als neues Standard-Layout verwenden:**

---

Wenn Sie das geänderte Layout eines Vorkommens als neues Standard-Layout verwenden wollen, gehen Sie wie folgt vor.

- Wählen Sie aus dem Kontextmenü des Vorkommens **Use as Default Layout**.

Folgende Abfrage wird angezeigt:

Do you want to use the current layout as the default layout for all future occurrences of this class?

- Brechen Sie den Vorgang mit **Cancel** ab

*oder*

- bestätigen Sie mit **OK**.

Das geänderte Layout wird zum neuen Standard-Layout. Es ist im Layout-Editor der Komponente verfügbar.

Wenn Sie neue Vorkommen der Komponente anlegen, wird das neue Standard-Layout verwendet. Existierende Vorkommen in vorhandenen Diagrammen werden jedoch *nicht* aktualisiert.

### Standard-Layout wieder herstellen:

Solange Sie nicht den Befehl **Use as Default Layout** aufgerufen haben, können Sie das im Layout-Editor der Komponente definierte Standard-Layout wieder herstellen.

- Wählen Sie aus dem Kontextmenü eines Vorkommens die Funktion **Default Layout**.

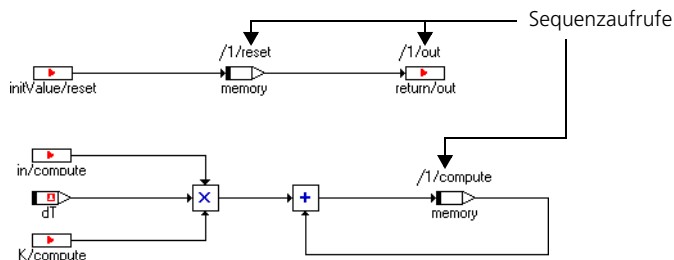
Das im Layout-Editor der Komponente definierte Standard-Layout wird wieder hergestellt.

Wenn dabei verbundene Anschlüsse entfernt werden, werden die Verbindungslinien mit entfernt.

Wenn verbundene Anschlüsse verschoben werden, werden die Verbindungen mit verschoben. Sie bleiben erhalten.

## 4.1.6 Sequenzaufrufe

Mit jeder Zuweisungsoperation oder jedem Methodenaufruf einer aufgenommenen Komponente ist ein Sequenzaufruf verknüpft. Jeder Sequenzaufruf stellt eine Anweisung in einem ASCET Diagramm dar. Sequenzaufrufe bestimmen den Kontrollfluss in Diagrammen, indem sie jede Anweisung einer Methode zuweisen und die Reihenfolge der Anweisungen innerhalb einer Methode bestimmen.

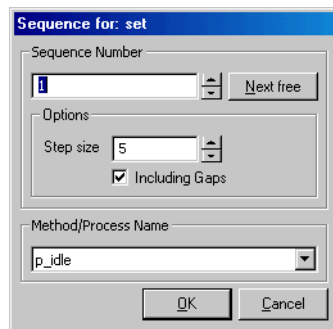


Im Blockdiagrammeditor werden Verbindungslinien zwischen Elementen und/ oder Operatoren als farbige Linien angezeigt, solange die Sequenzierung für die mit dem Element verknüpfte Anweisung oder Operation noch nicht gelöst ist. Die Farbe weist darauf hin, dass die Sequenzierung noch aufgelöst werden muss. Sie wird in Schwarz geändert für alle Linien, denen ein Sequenzaufruf zuwiesen worden ist.

Sie können die Sequenzaufufe einzeln oder zu mehreren bearbeiten, manuell oder automatisch.

### *Einzelne Sequenzaufufe bearbeiten*

Für die Bearbeitung und Konfiguration einzelner Sequenzaufufe wird der Sequenzeditor verwendet.



Dieser enthält folgende Elemente:

- Feld „Sequence Number“ – hier kann eine beliebige Nummer für den Sequenzaufruf eingegeben werden.

Standardmäßig für neue Sequenzaufufe die Nummer 1 angezeigt, für bereits zugewiesene Sequenzaufufe die aktuelle Nummer.

- Schaltfläche **Next free** – hiermit wird die nächste noch nicht vergebene Nummer nach bestimmten Regeln eingestellt (s. Seite 250).
- Feld „Step size“ – hier wird die Schrittweite für die Suche nach der nächsten freien Nummer eingestellt.

Standardmäßig ist der Wert der Option „Sequence Step Size“ aus dem ASCET-Optionsfenster (siehe „Knoten „Sequencing““ auf Seite 58) voreingestellt. Wenn Sie den Wert im Sequenzeditor ändern, wird der Wert im Optionsfenster angepasst.



- Option **Including Gaps** – hiermit wird festgelegt, ob Lücken zwischen vorhandenen Sequenznummern bei der Verwendung von **Next free** mit einbezogen werden oder nicht.

Voreingestellt ist der Wert der Option **Use Gaps** aus dem ASCET-Optionsfenster (siehe „Knoten „Sequencing““ auf Seite 58). Wenn Sie die Einstellung der Option **Including Gaps** im Sequenzeditor ändern, wird die Einstellung im Optionsfenster angepasst.

- Kombikästchen „Method/Process Name“ – hier wird der Sequenzaufruf einem Prozess / einer Methode zugeordnet.

Voreingestellt ist für neue Sequenzaufrufe der in der Fläche „Diagrams“ des Blockdiagrammeditors ausgewählte Prozess bzw. die Methode. Für bereits zugewiesene Sequenzaufrufe wird der aktuelle Prozess bzw. die Methode angezeigt.

Wenn Sie die Auswahl im Sequenzeditor ändern, wird die Auswahl in der Fläche „Diagrams“ ebenfalls geändert.

- **OK** übernimmt die geänderten Einstellungen und schließt den Sequenzeditor; **Cancel** schließt den Sequenzeditor, ohne die Einstellungen zu übernehmen.

### **Sequenzaufruf im Sequenzeditor bearbeiten:**

---

- Klicken Sie im Zeichenbereich mit der rechten Maustaste auf einen Sequenzaufruf, um das Kontextmenü zu öffnen.
- Wählen Sie aus dem Kontextmenü **Edit**, um den Sequenzeditor zu öffnen.
- Wählen Sie aus dem Kombikästchen „Method/Process Name“ den Prozess/die Methode für den Sequenzaufruf.
- Geben Sie im Feld „Step size“ die Schrittweite für die automatische Bestimmung der Sequenznummer ein.
- Aktivieren Sie die Option **Including Gaps**, wenn bei der automatischen Bestimmung der Sequenznummer Lücken zwischen vorhandenen Nummern berücksichtigt werden sollen.
- Stellen Sie im Feld „Sequence Number“ die Nummer des Sequenzaufrufs ein

*oder*

- klicken Sie auf **Next free**, um die nächste für den gewählten Prozess/die Methode noch freie Nummer einzustellen.
- Klicken Sie auf **OK**.

Es wird überprüft, ob die eingestellte Kombination aus Nummer und Prozess/Methode bereits vergeben ist. Wenn nicht, wird die Kombination dem Sequenzaufruf zugewiesen und im Blockdiagramm angezeigt.

„Step size“ und **Including Gaps** werden in das ASCET-Optionsfenster übernommen.

Für die Ermittlung der Nummer bei Verwendung von **Next free** gelten folgende Regeln:

1. Nur ganzzahlige Vielfache des Wertes im Feld „Step size“ (entspricht „Sequence Step Size“ im Knoten „Sequencing“ des ASCET-Optionsfensters, s. Seite 58) werden berücksichtigt. Wenn beispielsweise der Wert 5 eingestellt ist, werden nur die Nummern 5, 10, 15, 20, ... geprüft.
2. Wenn die Option **Including Gaps** aktiviert ist, werden eventuell vorhandene Lücken zwischen bereits vorhandenen Sequenznummern aufgefüllt. Bedingung 1 gilt trotzdem; Lücken, die kein ganzzahliges Vielfaches des Wertes im Feld „Sequence Step Size“ sind, werden nicht gefüllt.

Wenn z.B. die Nummern 1–3, 5–9 und 11 schon vergeben sind und im Feld „Sequence Step Size“ der Wert 5 angegeben ist, wird nicht die Nummer 4, sondern 10 vergeben.

### Hinweis

*Das System speichert die zuletzt vergebene Sequenznummer. Lücken unterhalb dieser Nummer werden **nicht** aufgefüllt!*

*Beim Verlassen des Editors oder durch das Zurücksetzen eines (Seite 253) oder mehrerer Sequenzaufrufe (Seite 257) wird die gespeicherte Nummer zurückgesetzt, die automatische Nummerierung beginnt wieder bei 1.*

3. Wenn die Option **Including Gaps** nicht aktiviert ist, wird nach der höchsten vorhandenen Nummer das nächste ganzzahlige Vielfache des Wertes im Feld „Sequence Step Size“ gesucht.

Im obigen Beispiel (1–3, 5–9 und 11 belegt) wird die Nummer 15 vergeben.

## Einzelne Sequenzaufufe automatisch zuweisen:

---

Sie können einzelne Sequenzaufufe auf folgende Weise einfach und schnell bearbeiten:

- Wählen Sie in der Fläche „Diagrams“ den Prozess/ die Methode, der der Sequenzaufuf zugewiesen werden soll.
- Doppelklicken Sie im Zeichenbereich auf den Sequenzaufuf, den Sie bearbeiten wollen,

oder

- klicken Sie mit der rechten Maustaste auf den Sequenzaufuf.
- Wählen Sie **Next Number** aus dem Kontextmenü.

In beiden Fällen können Sie sowohl einen noch nicht zugewiesener Sequenzaufuf als auch einen bereits bearbeiteten auswählen.

### Hinweis

*Wenn keine Methode/kein Prozess ausgewählt wurde, öffnen der Doppelklick und das Kontextmenü **Next Number** den Sequenzeditor.*

Dem Sequenzaufuf wird der ausgewählte Prozess/die Methode sowie die nächste freie Nummer zugewiesen.

Es gelten die auf Seite 250 beschriebenen Regeln, allerdings unter Verwendung der im ASCET-Optionsfenster eingestellten Werte für „Sequence Step Size“ und **Use gaps**.

## Einzelne Sequenzaufufe inkrementieren/dekrementieren:

---

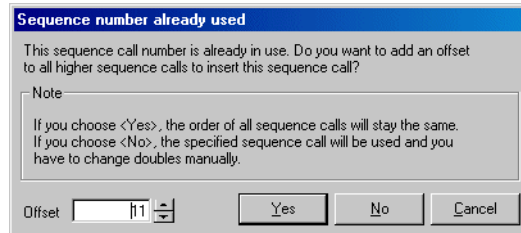
- Klicken Sie mit der rechten Maustaste auf den Sequenzaufuf, dessen Nummer Sie bearbeiten wollen.
- Wählen Sie **Change** → **Increment** aus dem Kontextmenü, um die Sequenznummer des Aufrufs um 1 zu erhöhen.

- Wählen Sie **Change** → **Decrement**, um die Sequenznummer des Aufrufs um 1 zu verringern.

### Bereits vergebene Nummer zuweisen:

---

Wenn Sie im Sequenzeditor oder beim Inkrementieren/Dekrementieren eine bereits vorhandene Kombination aus Prozess/Methode und Nummer zuweisen wollen, wird folgende Warnung angezeigt.



Sie haben die Möglichkeit, die existierende Nummer sowie alle höheren Nummern um einen einstellbaren Offset zu verschieben. Die Reihenfolge der verschobenen Sequenzaufrufe bleibt erhalten.

- Stellen Sie im Feld „Offset“ den Wert für die Verschiebung der Sequenzaufrufe ein.  
Voreingestellt ist der Wert der Option „Sequence Shift Offset“ aus dem ASCET-Optionsfenster (s. „Knoten „Sequencing““ auf Seite 58).
- Klicken Sie auf **Yes**.

Der bearbeitete Sequenzaufruf erhält die im Sequenzeditor eingestellten Werte. Der andere Sequenzaufruf sowie die Sequenzaufrufe mit höherer Nummer, die zum selben Prozess/Methode gehören, werden um den unter „Offset“ eingestellten Wert verschoben.

*Oder*

- Klicken Sie auf **No**, um die vorhandene Kombination dennoch zuzuweisen.

Das kann sinnvoll sein, wenn das erste Vorkommen der Kombination irrtümlich zustande kam, z.B. durch einen Tippfehler bei der Nummer.

#### Hinweis

*Sie müssen dann die doppelten Sequenzaufrufe manuell bearbeiten, um den Konflikt aufzulösen.*

*Ansonsten werden bei der Codegenerierung entsprechende Fehlermeldungen erzeugt.*

#### *Oder*

- Klicken Sie **Cancel**, um zum Sequenzeditor zurückzukehren.
- Stellen Sie eine noch nicht vergebene Kombination ein.

### **Einzelnen Sequenzaufruf zurücksetzen:**

---

Um einen einzelnen Sequenzaufruf zurückzusetzen, gehen sie wie folgt vor:

- Klicken Sie mit der rechten Maustaste auf einen Sequenzaufruf.
- Wählen Sie **Change** → **Reset** aus dem Kontextmenü.

Die aktuellen Werte des Sequenzaufrufs werden zurückgesetzt, die Verbindungslinie wird wieder farbig dargestellt.

Gleichzeitig wird die intern gespeicherte zuletzt vergebene Nummer gelöscht, so dass automatische Zuweisungen wieder beim niedrigsten möglichen Wert beginnen.

### **Bewegen zwischen Sequenzaufrufen:**

---

- Wählen Sie einen Sequenzaufruf im Zeichenbereich aus.
- Wählen Sie **Sequence Call** → **Next Seq. Call**, um den nächstfolgenden Aufruf in der Sequenz auszuwählen.

- Wählen Sie **Sequence Call** → **Last Seq. Call**, um den vorhergehenden Aufruf in der Sequenz auszuwählen.

### **Ändern der Sichtbarkeit einzelner Sequenzaufrufe:**

---

- Klicken Sie mit der rechten Maustaste auf den Eingangsanschluss des Elements, mit dem der Sequenzaufruf verknüpft ist.
- Deaktivieren Sie im Kontextmenü den Befehl **Sequence Call**.

*Oder*

- Klicken Sie mit der rechten Maustaste auf den Sequenzaufruf.
- Wählen Sie im Kontextmenü **Hide**.  
Der Sequenzaufruf wird versteckt.
- Klicken Sie mit der rechten Maustaste auf den Eingangsanschluss des Elements, mit dem der Sequenzaufruf verknüpft ist.
- Aktivieren Sie im Kontextmenü den Befehl **Sequence Call**.  
Der Sequenzaufruf wird angezeigt.
- Wählen Sie **Select Complete Port**, um den Anschluss zu markieren, mit dem der Sequenzaufruf verknüpft ist.

Der Eingangsanschluss des betreffenden Diagrammelements wird blau dargestellt. Dieses Merkmal ist nützlich, um in komplexen Diagrammen Sequenzaufufe verfolgen zu können.

### **Folge von geschützten Sequenzaufrufen erstellen:**

---

- Wählen Sie **Atomic** → **Start**, um eine Folge von geschützten Sequenzaufrufen zu starten.  
Eine Folge von geschützten Sequenzaufrufen kann in einer Echtzeitumgebung nicht unterbrochen werden.
- Wählen Sie **Atomic** → **Stop**, um eine Folge von geschützten Sequenzaufrufen zu beenden.

## Mehrere Sequenzaufrufe bearbeiten

---

In komplexen Diagrammen ist manchmal eine große Zahl von Sequenzaufrufen erforderlich. In diesem Falle ist es nützlich, wenn die Sequenzaufrufe in Gruppen bearbeitet werden können. Sie können alle Sequenzaufrufe ausgewählter Blöcke, einzelner Prozesse oder Methoden oder eines kompletten Diagramms ändern.

### Hinweis

Achten Sie darauf, dass Sie **nur Sequenzaufrufe** auswählen. Wenn auch Verbinder (siehe „Verbinder“ auf Seite 259) ausgewählt werden, funktionieren die automatischen Zuweisungen nicht.

Wenn ausschließlich Verbinder ausgewählt werden, funktioniert lediglich **Sequence Calls** → **Sequencing - Ignore Info**.

### Automatisches Zuweisen von Sequenzaufrufen:

---

- Wählen Sie in der Fläche „Diagrams“ den Prozess/die Methode aus, mit der die Sequenzaufrufe verknüpft werden sollen.
- Wählen Sie alle Elemente aus, denen Sie Sequenzaufrufe zuweisen möchten.

Sequenzaufrufe können immer nur einem Prozess/einer Methode auf einmal automatisch zugewiesen werden, deshalb sollten Sie nur die Elemente auswählen, die mit demselben Prozess/ derselben Methode verknüpft werden sollen.

### Hinweis

Wenn Sie keine Elemente auswählen, kommt es bei Vorhandensein mehrerer Prozesse/Methoden zu unvollständigen und fehlerhaften Zuweisungen.

- Wählen Sie **Sequence Calls** → **Sequencing - Ignore Info**.

Der Befehl analysiert das Diagramm und weist entsprechend dem eingebauten Sequenzierungsalgorithmus die Sequenzaufrufe zu.

Der Algorithmus der automatischen Sequenzierung setzt eine auf dem Datenfluss basierende Herangehensweise voraus, d.h. die Elemente werden in der durch den Datenfluss vorgegebenen Reihenfolge sequenziert. Wenn das Diagramm in einer anderen Reihenfolge sequenziert werden soll, müssen die Sequenzaufrufe von Hand eingestellt werden.

### **Automatisches Zuweisen von Sequenzaufrufen ab einer bestimmten Nummer:**

---

- Wählen Sie in der Fläche „Diagrams“ den Prozess/die Methode aus, mit der die Elemente verknüpft werden sollen.
- Wählen Sie alle Elemente aus, denen Sie Sequenzaufufe zuweisen möchten.
- Wählen Sie **Sequence Calls** → **Sequencing Starting With**.
- Geben Sie eine Zahl in das Eingabefeld ein.

Die ausgewählten Elemente werden mit dem ausgewählten Prozess/Methode sequenzalisiert, beginnend bei der angegebenen Sequenznummer.

### **Anhängen von Sequenzaufrufen an eine existierende Sequenz:**

---

Es ist möglich, Sequenzaufufe an eine früher definierte Sequenz anzuhängen, d.h. an eine Anzahl von Sequenzaufrufen, die bereits einem Prozess/einer Methode zugewiesen worden sind. In diesem Falle erhält der erste der neu zugewiesenen Sequenzaufufe eine Nummer, die um eins höher ist als die des letzten in der früher definierten Sequenz.

- Wählen Sie alle Elemente aus, die Sie an eine bereits definierte Sequenz anhängen möchten.
- Wählen Sie in der Fläche „Diagrams“ die Methode aus, mit der die Elemente verknüpft werden sollen.
- Wählen Sie **Sequence Calls** → **Sequencing Appending**.

Die ausgewählten Sequenzaufufe werden für den ausgewählten Prozess/die Methode an die bereits definierte Sequenz angehängt.



## Mehrere Sequenzaufufe verschieben:

---

Wenn Sie die Sequenznummern einer Gruppe von Sequenzaufrufen verschieben möchten, gehen Sie dazu wie folgt vor:

- Klicken Sie mit der rechten Maustaste auf den Sequenzaufruf mit der niedrigsten Nummer.
- Wählen Sie **Change** → **Shift by offset** aus dem Kontextmenü.

Die Sequenznummer des Aufrufs und aller Aufrufe mit höherer Nummer, die mit derselben Methode oder demselben Prozess verknüpft sind, werden um den in der Option „Sequence Shift Offset“ im ASCET-Optionsfenster (s. „Knoten „Sequencing““ auf Seite 58) eingestellten Wert nach oben verschoben.

## Sequenzaufufe skalieren:

---

Sie haben die Möglichkeit, alle Sequenzaufufe eines Prozesses/einer Methode oder des ganzen Diagramms zu skalieren.

- Wählen Sie in der Fläche „Diagrams“ den Prozess/die Methode, deren Sequenzaufufe Sie skalieren wollen.
- Wählen Sie **Sequence Calls** → **Scale To Step Size** → **For Method/Process**.

Die Sequenzaufufe der Methode/des Prozesses werden entsprechend dem unter „Sequence Step Size“ im ASCET-Optionsfenster (s. Seite 58) eingetragenen Wert skaliert.

*Oder*

- Wählen Sie **Sequence Calls** → **Scale To Step Size** → **For Diagram**, um alle Sequenzaufufe des Diagramms zu skalieren.

## Zurücksetzen von mehreren Sequenzaufrufen:

---

- Wählen Sie **Sequence Calls** → **Reset** → **For Diagram**, um alle Sequenzaufufe des aktuellen Diagramms zurückzusetzen.

- Wählen Sie **Sequence Calls** → **Reset** → **For Method/Process**, um alle Sequenzaufufe zurückzusetzen, die der Methode/dem Prozess zugewiesen sind, der aktuell in der Fläche „Diagrams“ ausgewählt ist.
- Wählen Sie **Sequence Calls** → **Reset** → **For Selected Blocks**, um alle Sequenzaufufe zurückzusetzen, die den gegenwärtig im Zeichenbereich ausgewählten Elementen zugewiesen sind.

Bei allen Befehlen wird zunächst eine Bestätigung gefordert, und anschließend werden die betreffenden Sequenznummern auf 0 gesetzt und die Sequenznamen gelöscht. Die Verbindungslinien werden wieder farbig dargestellt.

### Ändern der Sichtbarkeit von mehreren Sequenzaufrufen:

---

- Wählen Sie **Sequence Calls** → **Hide** → **For Diagram**, um alle Sequenzaufufe im aktuellen Diagramm zu verbergen.
- Wählen Sie **Sequence Calls** → **Hide** → **For Method/Process**, um alle Sequenzaufufe zu verbergen, die der gegenwärtig ausgewählten Methode/dem Prozess zugewiesen sind.
- Wählen Sie **Sequence Calls** → **Hide** → **For Selected Blocks**, um alle Sequenzaufufe zu verbergen, die den gegenwärtig im Zeichenbereich ausgewählten Elementen zugewiesen sind.
- Wählen Sie **Sequence Calls** → **Hide** → **Unused**, um alle gegenwärtig nicht verwendeten Sequenzaufufe zu verbergen.

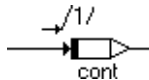
Der Befehl **Show** kehrt die Wirkung des Befehls **Hide** um, wobei dieselben vier Optionen zur Verfügung stehen.

## Verbinder

Verbinder werden verwendet, um eine Zuweisung mit einer Kontrollflussanweisung zu verbinden, wie etwa `If...Then` oder `If...Then...Else`.

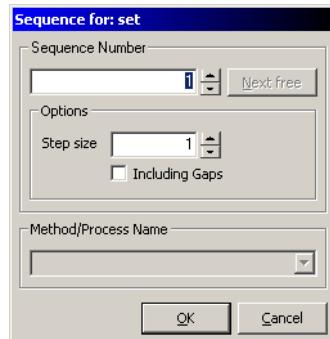
### Verbinder erstellen:

- Klicken Sie mit der rechten Maustaste auf den Sequenzaufruf, den Sie zu einem Verbinder machen wollen.
- Wählen Sie **Connector** aus dem Kontextmenü.



Der Sequenzaufruf wird zu einem Verbinder. Dieser kann mit einer Kontrollflussanweisung verbunden werden.

Verbinder können wie Sequenzaufrufe im Sequenzeditor bearbeitet werden (siehe „Sequenzaufruf im Sequenzeditor bearbeiten:“ auf Seite 249). Allerdings sind die Schaltfläche **Next free** und das Feld „Method/Process Name“ deaktiviert. Der Doppelklick auf den Verbinder und das Kontextmenü **Next Number** öffnen ebenfalls den Sequenzeditor.



Die meisten Möglichkeiten zur automatischen Bearbeitung sind für Verbinder *nicht* vorhanden.

## Automatisches Zuweisen von Verbindern:

---

### Hinweis

*Dies ist die einzige automatische Zuweisung, die für Verbinder funktioniert.*

- Wählen Sie in der Fläche „Diagrams“ den Prozess/die Methode aus, mit der die Sequenzaufrufe verknüpft werden sollen.
- Wählen Sie alle Verbinder aus, die in dem gewählten Prozess/der Methode verwendet werden sollen.

### Hinweis


*Achten Sie unbedingt darauf, dass Sie **nur Verbinder** auswählen, da die automatische Zuweisung andernfalls nicht funktioniert.*

- Wählen Sie **Sequence Calls** → **Sequencing - Ignore Info**.

Der Befehl analysiert das Diagramm und weist entsprechend dem eingebauten Algorithmus die Nummern der Verbinder zu.

### 4.1.7 Implementation-Casts in Blockdiagrammen

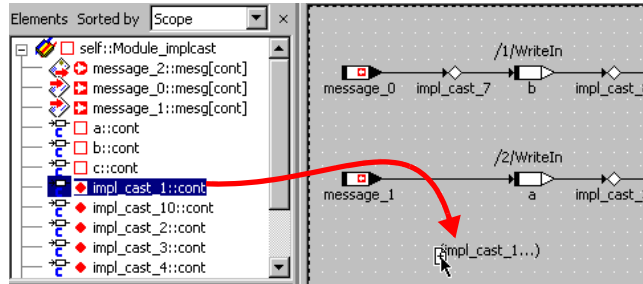
---

Im Blockdiagrammeditor können Implementation-Casts (siehe „Implementation-Casts“ im ASCET-Referenzhandbuch) wie alle anderen Elemente durch die zugehörige Schaltfläche in der Schaltflächenleiste (hier: **Implementation cast**, ) eingefügt werden. Einmal erzeugt, können sie aus der Elementliste durch Drag&Drop in den Zeichenbereich eingefügt und dort wie alle anderen Elemente verbunden werden.

### Hinweis

*Implementation-Casts können **nicht** auf logische Elemente angewendet werden. Wenn Sie einen Implementation-Cast mit einem logischen Element oder Anschluss verbinden, wird die Verbindungslinie rot dargestellt, um auf den Fehler hinzuweisen.*

Sequenzaufrufe gibt es für Implementation-Casts nicht, die korrekte Reihenfolge wird von der Codegenerierung aus dem Kontext ermittelt.



Im Blockdiagrammeditor gibt es noch eine weitere, sehr komfortable Möglichkeit, Implementation-Casts einzufügen. Diese ist speziell für bereits existierende arithmetische Berechnungsketten vorteilhaft. Über das Kontextmenü der arithmetischen Operatoren  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $\text{abs}$  und  $\text{neg}$  haben Sie die Möglichkeit, durch Auswahl von **Add Implementation Casts** automatisch solche Implementation-Casts für alle Ein- und Ausgänge der Operation einzufügen.

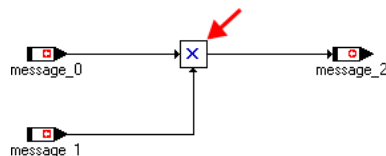
#### Automatisches Einfügen von Implementation-Casts an Operatoren:

##### Hinweis

Das Verfahren funktioniert nur, wenn **keiner** der Ein- und Ausgänge des Operators direkt mit einem Implementation-Cast verbunden ist. Andernfalls erscheint folgende Fehlermeldung:

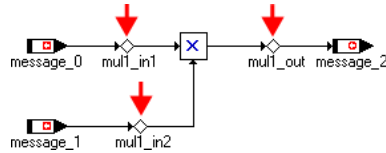
This operator is already connected to at least one implementation cast. Please specify further implementation casts individually.

- Markieren Sie den Operator, für den Implementation-Casts hinzugefügt werden sollen.



- Klicken Sie mit der rechten Maustaste auf den Operator und wählen Sie **Add Implementation Cast** aus dem Kontextmenü.

Wenn im Zeichenbereich genügend Platz vorhanden ist, werden auf allen Verbindungen mit Ein- und Ausgängen des Operators Implementation-Casts eingefügt.



Wenn nicht genügend Platz vorhanden ist, wird folgende Warnung angezeigt:

The layout is too tight to place an implementation cast automatically. Please rearrange the diagram.

- Verlängern Sie die Verbindungen zum Operator und versuchen Sie es erneut.

Die Implementation-Casts werden automatisch benannt, und zwar nach folgendem Schema:

`<operator><m>_<pin type><n>`

- `<operator>` kann, abhängig vom ausgewählten Operator, die Werte `add`, `sub`, `mul`, `div`, `abs` oder `neg` annehmen.
- `<m>` ist die Nummer des Operators. Der erste Operator eines Typs, für den auf diese Weise Implementation-Casts erzeugt werden, bekommt die Nummer 1, bei weiteren Operatoren desselben Typs wird hochgezählt.
- `<pin type>` ist die Bezeichnung des Operatoranschlusses, der mit dem Implementation-Cast verbunden ist, d.h. `in` für Eingänge und `out` für Ausgänge.
- `<n>` ist die Nummer des Implementation-Casts. Mit Eingängen verbundene Implementation-Casts und mit dem Ausgang verbundene Implementation-Casts werden getrennt nummeriert.
  - Der Implementation-Cast am ersten Eingang des Operators bekommt die Nummer 1, weitere Eingänge werden hochgezählt. Hat der Operator nur einen Eingang, entfällt die Nummer für den Implementation-Cast.

- Ist der Operatorausgang mit mehreren Elementen verbunden, werden die Implementation-Casts auf den Verbindungen bei 1 beginnend nummeriert.  
Ist der Operatorausgang mit einem Element verbunden, entfällt die Nummer.

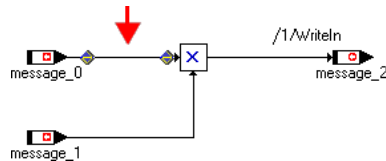
### Automatisches Einfügen von Implementation-Casts auf einer Verbindung:

Ebenso können Sie auf allen Verbindungslinien (Datenpfaden) von arithmetischen Werten über das Kontextmenü einen Implementation-Cast einfügen.

#### Hinweis

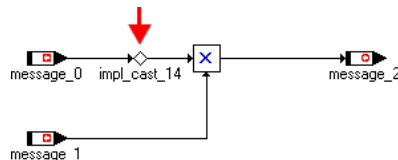
Dies gilt **nicht** für Verbindungen mit anderen Operatoren als +, -, \*, /, abs, neg, max, min und mux, Verbindungen mit logischen Elementen oder Kontrollfluss-Verbindungslinien.

- Wählen Sie die Verbindung, auf dem Sie einen Implementation-Cast einfügen wollen.



- Klicken Sie mit der rechten Maustaste auf die Verbindung und wählen Sie **Add Implementation Cast** aus dem Kontextmenü.

Es wird automatisch ein Implementation-Cast auf der Verbindung eingefügt.



Die Implementation-Casts werden automatisch benannt, und zwar nach folgendem Schema:

`impl_cast_<n>`

<n> ist die Nummer des Implementation-Casts. Der erste automatisch auf einer Verbindung erzeugte Implementation-Cast bekommt keine Nummer, der zweite bekommt die Nummer 1, weitere automatisch auf beliebigen Verbindungen erzeugte Implementation-Casts werden hochgezählt.

Dieses Schema gilt für alle Implementation-Casts, die *nicht* automatisch für einen Operator erzeugt wurden (s. Seite 261).

### Anzeige der Implementation-Casts ein-/ausschalten:

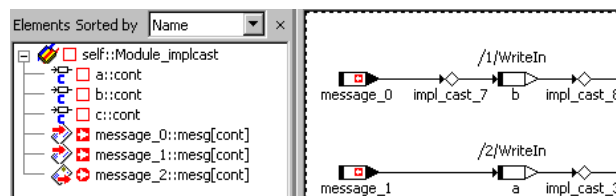
Standardmäßig werden die Implementation-Casts wie alle Elemente in der Liste „Elements“ angezeigt. Sie haben jedoch die Möglichkeit, die Anzeige abzuschalten.

- Öffnen Sie das Untermenü **View** → **Show/Hide**.

Wenn die Anzeige eingeschaltet ist, ist die Menüfunktion **Impl. Casts in Elements List** mit einem Häkchen versehen.

- Wählen Sie **View** → **Show/Hide** → **Impl. Casts in Elements List**.

Die Implementation-Casts werden in der Liste „Elements“ nicht mehr angezeigt, das Häkchen an der Menüfunktion wird entfernt.



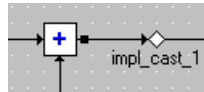
Im Zeichenbereich werden die Implementation-Casts nach wie vor entsprechend den Einstellungen für die aktuelle Sicht angezeigt. Das Kontextmenü der Implementation-Casts enthält jedoch nur noch wenige Funktionen.

- Wählen Sie **View** → **Show/Hide** → **Impl. Casts in Elements List** erneut, um die Anzeige wieder einzuschalten.

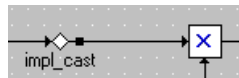
Implementation-Casts können in Verbindung mit temporären Variablen verwendet werden. Dabei ist folgendes zu beachten:



- Wenn ein Implementation-Cast auf eine temporäre Variable folgt, wird die temporäre Variable nicht vom Implementation-Cast beeinflusst.



- Temporäre Variablen unmittelbar nach einem Implementation-Cast werden unterstützt.



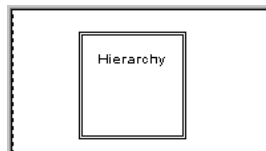
#### 4.1.8 Grafische Hierarchien

Um komplexe Diagramme übersichtlicher zu organisieren, können Sie Teile eines Diagramms zu einem Hierarchieblock gruppieren, der dann auf der obersten Ebene des Diagramms nur als Symbol sichtbar ist. Die Tatsache, ob sich irgendein Diagrammelement innerhalb eines Hierarchieblocks befindet oder nicht, hat keinerlei Einfluss auf seine Funktionalität; Hierarchien dienen nur zur grafischen Strukturierung. Hierarchien können verschachtelt werden, so dass Hierarchieblöcke andere Hierarchieblöcke enthalten können.

##### Hinzufügen einer neuen Hierarchie:



- Klicken Sie auf die Schaltfläche **Hierarchy** in der Werkzeugleiste des Blockdiagrammeditors, um den Mauszeiger mit einer Hierarchie zu laden.
- Klicken Sie im Zeichenbereich auf die Stelle, wo Sie die Hierarchie anordnen möchten.  
Der Hierarchieblock wird zum Diagramm hinzugefügt.



## Diagrammelemente in einen Hierarchieblock umwandeln:

---

- Wählen Sie die Diagrammelemente aus, die Sie in einem Hierarchieblock anordnen möchten.
- Klicken Sie auf die Schaltfläche **Hierarchy**.  
Ein Hierarchieblock, der die ausgewählten Elemente enthält, wird zum Diagramm hinzugefügt.

Die ausgewählten Diagrammelemente werden innerhalb eines neu erzeugten Hierarchieblocks angeordnet. Für jede Linie, die ein Diagrammelement außerhalb des Hierarchie Rahmens mit einem innerhalb desselben verbindet, wird ein Anschluss mit einem Standardnamen erzeugt. Es ist nicht möglich, Diagrammelemente z.B. durch Drag&Drop direkt in einen Hierarchieblock zu bewegen. Sie müssen stets über die Zwischenablage kopiert oder auf die hier beschriebene Weise umgewandelt werden.

## Bewegen von Elementen in einen und aus einem Hierarchieblock:

---

- Kopieren oder verschieben Sie Diagrammelemente aus der höheren Ebene mit **Cut** oder **Copy** in die Zwischenablage.
- Doppelklicken Sie auf die Hierarchie, die die Elemente aufnehmen soll.  
Der Inhalt der Hierarchie wird angezeigt.
- Fügen Sie die Diagrammelemente mit **Paste** in den Hierarchieblock ein.
- Doppelklicken Sie in den Zeichenbereich, um wieder in die höhere Ebene zu gelangen.

## Auflösen eines Hierarchieblocks:

---

- Klicken Sie mit der rechten Maustaste auf den Hierarchieblock und wählen Sie **Resolve** aus dem Kontextmenü.  
Der Hierarchieblock wird entfernt, und alle darin enthaltenen Diagrammelemente werden in das Diagramm eine Ebene weiter oben gezogen.

*Oder*

- Wählen Sie einen Hierarchieblock aus.

- Wählen Sie **Edit** → **Delete**.

Der Hierarchieblock wird mit allen darin enthaltenen Diagrammelementen gelöscht.

### **Aussehen eines Hierarchieblocks ändern:**

---

- Klicken Sie mit der rechten Maustaste in den Block und wählen Sie aus dem Kontextmenü **Rename Hierarchy**, um den Hierarchieblock umzubenennen.

Sie werden aufgefordert, einen Namen für die Hierarchie einzugeben.

- Geben Sie den Namen in das Eingabefeld ein und klicken Sie auf **OK**.

- Klicken Sie mit der rechten Maustaste in den Block und wählen Sie aus dem Kontextmenü **Show/Hide Name**.

Dadurch wird der Name der Hierarchie unsichtbar. Wird der Befehl erneut gewählt, erscheint der Name wieder.

- Klicken Sie mit der rechten Maustaste in den Block und wählen Sie **Change Icon**, um der Hierarchie eine Symboldatei zuzuweisen.

Es wird eine Dateiauswahl angezeigt, aus dem Sie eine Bitmap-Bilddatei im Format TIFF, PCX oder BMP auswählen können.

- Wählen Sie die gewünschte Bilddatei aus und klicken Sie auf **OK**.

Die Bilddatei wird zusammen mit dem Diagramm in der Datenbank von ASCET gespeichert.

- Wählen Sie den Hierarchieblock aus und ändern Sie die Größe des Rahmens, indem Sie die Anfassers ziehen.

- Wählen Sie aus dem Kontextmenü **Set To Default Size** aus, um zur Standardgröße des Hierarchieblocks zurückzukehren.

Die Verfahrensweise beim Navigieren zwischen Hierarchie-Ebenen ist dieselbe wie beim Navigieren zwischen aufgenommenen Komponenten eines Diagramms.

### **Navigieren zwischen Hierarchie-Ebenen:**

---

- Doppelklicken Sie auf den Hierarchieblock.

*Oder*

- Klicken Sie mit der rechten Maustaste in den Block und wählen Sie aus dem Kontextmenü **Next Level**.

Sie bewegen sich ins Innere des Hierarchieblocks, d.h. der Inhalt des aktuellen Blocks wird im Zeichenbereich dargestellt.

- Doppelklicken Sie innerhalb der Hierarchie auf den Zeichenbereich (nicht auf ein Diagrammelement).

Sie verlassen die Hierarchie, und der Block, den Sie verlassen haben, ist ausgewählt.

### **Hinzufügen von Eingangs- und Ausgangsanschlüssen zur Hierarchie:**

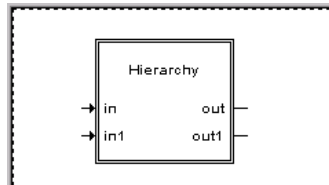
---

Der Datenfluss zwischen den verschiedenen Hierarchie-Ebenen erfolgt über Eingangs- und Ausgangsanschlüsse. Dabei handelt es sich einfach um Verbindungslinien, die über die Ebenen hinweg reichen.

- Klicken Sie mit der rechten Maustaste auf den Hierarchieblock.
- Wählen Sie aus dem Kontextmenü **Add Inpin** oder **Add Outpin**.

Sie können eine beliebige Anzahl von Eingangs- und Ausgangsanschlüssen zum Hierarchierahmen hinzufügen. Die Eingangs- und Ausgangsanschlüsse werden durch Pfeil-

symbole dargestellt, die den Namen des Anschlusses innerhalb des Hierarchieblocks enthalten.



### Ändern des Aussehens von Eingangs- und Ausgangsanschlüssen:

---

- Klicken Sie mit der rechten Maustaste auf einen Eingangs- oder Ausgangsanschluss.
- Wählen Sie aus dem Kontextmenü **Rename Pin**, um den Namen des Anschlusses zu ändern.
- Geben Sie den Namen in das Eingabefeld ein und klicken Sie auf **OK**.
- Wählen Sie **Pin Name**, um den Namen des Anschlusses unsichtbar zu machen.
- Wählen Sie nochmals denselben Befehl, um ihn wieder sichtbar zu machen.
- Wählen Sie **Remove Pin**, um den Eingangs- oder Ausgangsanschluss zu entfernen.
- Klicken Sie mit der rechten Maustaste auf den Hierarchieblock und wählen Sie **Show Pin Names** oder **Hide Pin Names**, um alle Namen von Anschlüssen des Hierarchieblocks zu zeigen oder zu verbergen.

Wenn Sie ein Diagrammelement mit einem Anschluss verbinden und danach den Anschluss mit einem anderen Diagrammelement innerhalb des Hierarchieblocks verbinden, so ist dies dasselbe, als ob diese zwei Elemente direkt verbunden würden.

## 4.1.9 Navigieren zwischen Diagrammen

---

### *Komponenten mit mehreren Diagrammen*

---

Eine Komponentenspezifikation für eine Klasse oder ein Modul kann aus mehr als einem Diagramm bestehen. Dieses Merkmal ist für das Strukturieren komplexer Spezifikationen von Nutzen. Ein Diagramm kann entweder öffentlich sein, d.h. nur öffentliche Methoden enthalten, oder privat, d.h. nur private Methoden enthalten.

Auf öffentliche Methoden kann von anderen Komponenten aus zugegriffen werden, auf private Methoden nicht. Auf private Methoden kann nur aus der Komponente heraus zugegriffen werden. Ein Sonderfall sind private Diagramme, die Aktionen und Bedingungen enthalten. Diese werden nur in Zustandsautomaten verwendet.

Jede als Blockdiagramm spezifizierte Komponente hat mindestens ein öffentliches Diagramm, das `Main` genannt wird. Klassen können eine beliebige Anzahl von öffentlichen oder privaten Diagrammen haben, Module hingegen haben nur öffentliche Diagramme.

#### **Erzeugen eines neuen Diagramms:**

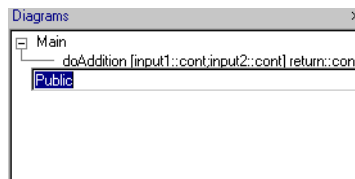
---

- Wählen Sie **Diagram** → **Add Diagram** → **Public** oder **Private**

oder

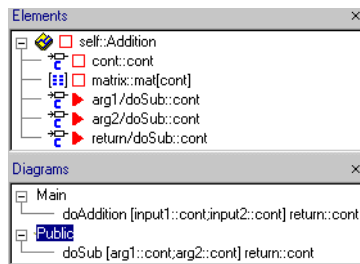
- wählen Sie in der Fläche „Diagrams“ aus dem Kontextmenü **Add Diagram** → **Public** oder **Private**.

In der Fläche „Diagrams“ wird ein neues öffentliches oder privates Diagramm hinzugefügt. Der Name des neuen Diagramms wird automatisch ausgewählt.



- Geben Sie einen Namen ein und drücken Sie <ENTER>.  
Sie können das Diagramm auch später umbenennen. Wählen Sie dazu **Diagram** → **Rename Diagram**.
- Fügen Sie die erforderlichen Methoden oder Prozesse hinzu, um die Funktionalität des Diagramms zu spezifizieren.

In der Liste „Elements“ werden die Schnittstellenelemente, d.h. Argumente, lokale Variable und Rückgabewert, nur für die Methoden in dem aktuellen Diagramm dargestellt. Andere Elemente, wie etwa Variable oder Parameter, können für verschiedene Diagramme verwendet werden und sind in der Liste „Elements“ immer sichtbar.



Im Blockdiagrammeditor arbeiten Sie immer nur an jeweils einem Diagramm. Falls eine Komponente mehr als nur ein Diagramm enthält, können Sie zwischen Diagrammen umschalten, indem Sie ein neues Diagramm in den Zeichenbereich laden.

### Diagramm laden:

- Wählen Sie ein Diagramm.
- Wählen Sie **Diagram** → **Load Diagram**  
*oder*

- wählen Sie aus dem Kontextmenü **Load Diagram**.

Das ausgewählte Diagramm wird in den Zeichenbereich des Blockdiagrammeditors geladen.

Falls in dem aktuellen Diagramm nicht gesicherte Änderungen vorhanden sind, werden Sie aufgefordert, das aktuelle Diagramm zu sichern, bevor das neue Diagramm geladen wird.

### **Löschen eines Diagramms aus einer Komponente:**

---

- Wählen Sie in der Liste „Diagrams“ das zu löschende Diagramm aus.

- Wählen Sie **Diagram** → **Delete Diagram**  
oder

- wählen Sie aus dem Kontextmenü **Delete Diagram**.

Das Diagramm wird zusammen mit sämtlichen darin enthaltenen Methoden/Prozessen aus der Komponente gelöscht.

#### **Hinweis**

*Das oberste Diagramm im Feld „Diagrams“ kann nicht gelöscht werden.*

### **Bewegen von Methoden zwischen Diagrammen:**

---

- Wählen Sie in der Fläche „Diagrams“ die Methode aus, die Sie verschieben möchten.



- Wählen Sie **Diagram** → **Move**, um die Methode zu verschieben.

Im Fenster „Specification“ wird eine Liste aller Diagramme für die aktuelle Komponente angezeigt.



- Wählen Sie das Diagramm aus, in das Sie die Methode verschieben möchten, und klicken Sie auf **OK**.

Die ausgewählte Methode wird in das Zieldiagramm verschoben. Wenn die Methode im ursprünglichen Diagramm noch Sequenzauf-rufe hat, werden diese zurückgesetzt.

Sie können Methoden nur dann verschieben, wenn ihre Schnittstellenelemente keine Vorkommen in dem Diagramm haben. Andernfalls entfernen Sie zuerst die Vorkommen von Schnittstellenelementen und verschieben Sie danach die Methode.

### *Navigieren zwischen Komponenten*

Wenn Sie im Blockdiagrammeditor eine Komponente bearbeiten, die andere Komponenten beinhaltet, können Sie diese Unterkomponenten bearbeiten, ohne über den Komponentenmanager gehen zu müssen, um den Editor auf einer enthaltenen Komponente zu öffnen.

### **Navigieren zwischen verschiedenen Ebenen eines Blockdiagramms:**

- Wählen Sie im Zeichenbereich oder in der Fläche „Elements“ die zu bearbeitende Komponente aus.
- Wählen Sie **Element** → **Edit Component**  
*oder*

- wählen Sie aus dem Kontextmenü der Komponente **Edit Component**

oder

- doppelklicken Sie auf die aufgenommene Komponente.

Für die ausgewählte Komponente wird ein Editor geöffnet.

Falls die aufnehmende Komponente nicht gesicherte Änderungen enthält, wird ein neues Editorfenster geöffnet. Andernfalls wird die Unterkomponente in das ursprüngliche Editorfenster geladen.

- Um sich in einer Einschluhhierarchie aufwärts zu bewegen, doppelklicken Sie in den Zeichenbereich, jedoch nicht auf ein Diagrammelement.

Sie gelangen eine Ebene nach oben. Wenn Sie die Unterkomponente geändert haben, wird für die vorhergehende Komponente ein neues Editorfenster geöffnet, und das Fenster für die Unterkomponente bleibt geöffnet.

Die hier beschriebenen Mechanismen werden auch zum Navigieren zwischen verschiedenen Ebenen von grafischen Hierarchien in derselben Komponente verwendet.

#### 4.1.10 Analysieren von Komponenten

---

Nach dem Erzeugen eines Blockdiagramms können Sie mit der erzeugten Komponente auch experimentieren, um zu sehen, ob diese wie beabsichtigt funktioniert. In diesem Abschnitt wird beschrieben, wie man sich auf das Offline-Experimentieren mit einer Komponente vorbereitet und wie die Experimentierumgebung gestartet wird.

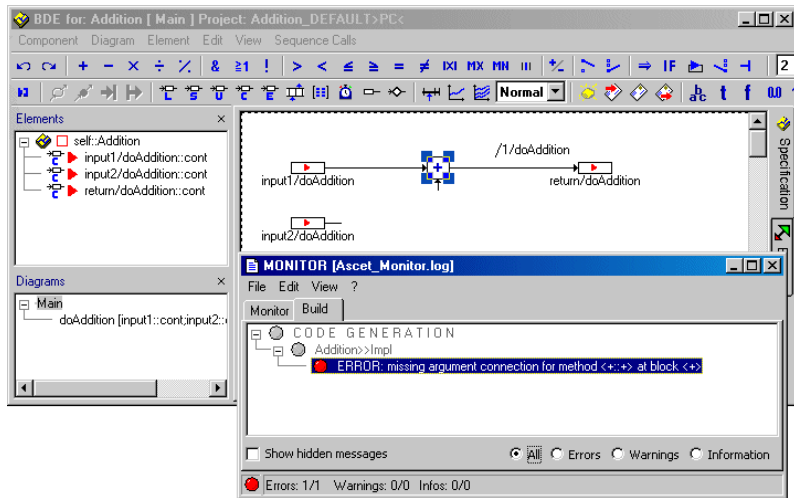
##### **Hinweis**

*Für ein einzelnes ASCET-Modul ohne Projektkontext kann Code nur mit dem Codegenerator `Physical Experiment` (s. Seite 397) generiert und simuliert werden. Für jeden anderen Codegenerator muss das Modul in ein Projekt integriert werden. Dafür kann zu jeder Klasse und jedem Modul auch ein sogenanntes Standard- oder Default-Projekt definiert werden. Nur auf diese Weise sind die benötigten Implementierungen verfügbar. Ohne Projektkontext fehlt die Konvertierungsformel sowie sämtliche Implementierungen importierter Größen.*

## Analysieren eines Diagramms:

- Wählen Sie **Diagram** → **Analyze Diagram**, um das aktuelle Diagramm zu analysieren.

Das Diagramm wird auf syntaktische Fehler geprüft. Nicht zugewiesene Sequenzaufrufe, fehlende Verbindungen usw. werden im Monitorfenster (siehe Kapitel 2.4 „Das ASCET-Monitorfenster“) gemeldet.



- Klicken Sie auf eine Fehlermeldung im Monitorfenster, Register „Build“, um den Fehler im Blockdiagrammeditor automatisch hervorheben zu lassen.

## Generieren von Code für eine Komponente:



- Wählen Sie **Component** → **Generate Code** oder
- klicken Sie auf die Schaltfläche **Generate Code**.

Der C-Code für die aktuelle Komponente wird generiert.

Das System zeigt eventuelle Fehlermeldungen an, die vom Codegenerator zurückgegeben

werden. Auch in diesem Fall können Sie das betreffende Diagrammelement finden, indem Sie auf eine Fehlermeldung doppelklicken.

Wenn der Code für eine Komponente erfolgreich generiert worden ist, können Sie die Offline-Experimentierumgebung für die Komponente öffnen.

Der durch ASCET generierte Code wird in der Datenbank gespeichert, wo er vom Benutzer nicht betrachtet werden kann. Es ist jedoch möglich, den Code in eine Datei zu schreiben, wo er mit einem beliebigen Texteditor betrachtet werden kann.

### Ansehen des generierten Codes:

---

- Generieren Sie den Code für die aktuelle Komponente wie oben beschrieben.
- Wählen Sie **Component** → **File Out Generated Code** → **Flat** oder **Recursive**, um den generierten Code im Dateisystem zu speichern.

Die generierten Dateien werden in dem ausgewählten Verzeichnis abgelegt. Wurde Code rekursiv im Dateisystem gespeichert, wird für jede Komponente eine separate Datei angelegt. Die Dateinamen der generierten Dateien werden im Monitorfenster von ASCET aufgelistet.

- Öffnen Sie die Dateien mit einem Texteditor, um sich diese anzusehen.

Wenn Sie **Flat** wählen, wird nur der Code für die aktuell ausgewählte Komponente in einer Datei abgelegt; wenn Sie **Recursive** wählen, wird der Code für die durch die aktuelle Komponente referenzierten Komponenten ebenfalls in eine Datei geschrieben.

Eine andere Variante zur Ansicht des generierten Codes besteht darin, direkt aus ASCET einen benutzerdefinierten externen Editor (z.B. Notepad, Code-wright, etc.) bzw. einen Internet-Browser zu nutzen. Da der Link zum externen Editor über ein File System realisiert ist, muss eine Verknüpfung der Dateien mit den Endungen „\*.c“ und „\*.h“ zum Editor in der Windows-Konfiguration erfolgen.

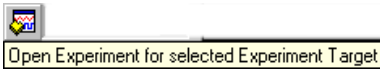
- Wählen Sie **Component** → **View Generated Code**.

C-Code wird generiert und im externen Editor angezeigt.

Der Editor kann im ASCET-Optionsfenster, Knoten „ASCII Editor“ (s. Seite 61), gewählt werden.

### Starten eines Offline-Experiments:

---



- Wählen Sie **Component** → **Open Experiment**

oder

- klicken Sie auf die Schaltfläche **Open Experiment for selected Experiment Target**.

Code wird generiert und mit dem für das aktuelle Target spezifischen Compiler kompiliert, und die Experimentierumgebung für die Komponente wird geöffnet. Die generierten Dateien werden im cgen-Verzeichnis abgelegt.

Die Experimentierumgebung wird in „Die Experimentierumgebung“ auf Seite 603 beschrieben.

Unter manchen Umständen werden die globalen Elemente (s. „Definieren der globalen Kommunikation“ auf Seite 416) im Standardprojekt nicht korrekt aktualisiert. In diesem Fall wird im ASCET Monitorfenster folgende Fehlermeldung ausgegeben:

```
Error: need export for imported element <name> with
type <type>
```

Um den Fehler zu korrigieren, gehen Sie wie folgt vor:

### Definieren von globalen Elementen im Standardprojekt:

---

- Wählen Sie im Blockdiagrammeditor **Component** → **Default Project** → **Resolve Globals**, um die globalen Elemente aufzulösen.

- Wählen Sie **Component** → **Default Project** → **Delete Unused Globals**, um unbenutzte globale Elemente zu entfernen.

Sie können jetzt das Experiment erneut starten.

#### 4.1.11 Datenaustausch

---

Ein Datensatz ist immer mit der Komponente verknüpft, von der er stammt. Um den Datensatz zu importieren, verwenden Sie den Befehl **File → Import** im Komponentenmanager. Dieser Befehl schreibt den importierten Datensatz zurück auf die Komponente, zu der er gehört. Für das Exportieren von Datensätzen gelten dieselben Regeln wie für das Exportieren von Datenbankeinträgen. Einzelheiten dazu siehe Abschnitt „Exportieren von Ordnern und Datenbankeinträgen“ auf Seite 94.

##### **Exportieren des Datensatzes einer Komponente:**

---

- Wählen Sie **Component → Export Data**.  
Das Dateiauswahlfenster wird geöffnet.
- Wählen Sie Datei- und Pfadnamen aus.
- Klicken Sie auf **OK**.  
Der Datensatz wird in die ausgewählte Datei geschrieben.

Es ist möglich, Teile eines Diagramms in einer Datei zu speichern und diese Diagrammelemente in eine andere Komponente zu importieren. Alle Diagrammelemente mit Ausnahme von Schnittstellenelementen können in einer Datei gespeichert werden.

##### **Speichern von Teilen eines Diagramms in einer Datei:**

---

- Wählen Sie den Teil des Diagramms aus, den Sie in eine Datei schreiben möchten.
- Wählen Sie **Edit → Copy**.
- Wählen Sie **Edit → File Out Buffer**.  
Das Dateiauswahlfenster wird geöffnet.
- Wählen Sie einen Pfad aus und geben Sie einen Dateinamen mit der Erweiterung `.asc` ein.
- Klicken Sie auf **OK**.  
Die in der Zwischenablage abgelegten Diagrammelemente werden in der vorgegebenen Datei gespeichert.

### Lesen eines Diagrammteils aus einer Datei:

---

- Wählen Sie **Edit** → **File In Buffer**, um das Diagramm in eine andere Komponente zu laden.  
Das Dateiauswahlfenster wird geöffnet.
- Wählen Sie die Datei aus, die die zu importierenden Diagrammelemente enthält.
- Klicken Sie auf **OK**.  
Die importierten Diagrammteile werden in die Zwischenablage der Datenbank geschrieben.
- Wählen Sie **Edit** → **Paste**.  
Die Diagrammelemente aus der Datei werden in das aktuelle Diagramm der aktuellen Komponente geschrieben.

Es ist möglich, die Daten einer Tabelle oder eines Arrays in eine Datei zu schreiben und sie wieder einzulesen. Die Daten werden in einem durch Tabs getrennten ASCII-Format geschrieben, so dass sie mit einem beliebigen Tabellenkalkulationsprogramm oder Textverarbeitungssystem gelesen und bearbeitet werden können.

### Schreiben der Daten eines Array oder einer Tabelle in eine Datei:

---

- Wählen Sie in der Fläche „Elements“ die Tabelle oder den Array aus.
- Wählen Sie **Element** → **File Out Data**.  
Es wird das Dateiauswahlfenster geöffnet.
- Wählen Sie einen Pfad und einen Dateinamen mit der Erweiterung `.dat` aus.
- Klicken Sie auf **OK**.  
Die Daten der ausgewählten Tabelle oder des Arrays werden in die Datei geschrieben.

### Einlesen der Daten eines Arrays oder einer Tabelle aus einer Datei:

---

- Wählen Sie in der Fläche „Elements“ die Tabelle oder den Array aus.
- Wählen Sie **Element** → **File In Data**.  
Das Dialogfenster von Windows zur Dateiauswahl wird geöffnet.

- Wählen Sie die Datei aus, die die Daten enthält, die Sie einlesen wollen.
- Klicken Sie auf **Öffnen**.  
Die Daten aus der Datei werden in die ausgewählte Tabelle oder den Array geschrieben.

#### 4.1.12 Verwendung des Blockdiagrammeditors

---

##### *Speichern von Diagrammen*

---

##### **Speichern einer Komponentenspezifikation:**

---

Um die spezifizierte Komponente zu sichern, gehen Sie wie folgt vor:

- Wählen Sie **Diagram** → **Store to Cache**.  
*oder*
- wählen Sie in der Fläche „Diagrams“ aus dem Kontextmenü **Store to Cache**.  
Die Komponente wird im Cache der Datenbank gesichert. Dieser Befehl bewirkt keine permanente Speicherung in der Datenbank.
- Um die Änderungen dauerhaft zu speichern, wählen Sie im Komponentenmanager **File** → **Save Database**  
*oder*
- klicken Sie auf die Schaltfläche **Save**.  
Durch den Befehl **Save** werden alle vorgenommenen Änderungen in der Datenbank gespeichert.

##### *Betrachten und Drucken von Diagrammen*

---

##### **Ändern der Darstellungsweise eines Blockdiagramms:**

---

- Wählen Sie **View** → **Redraw**, um das Diagramm neu zu zeichnen.  
Manchmal kann ein Diagramm verfälscht werden, wenn Sie Elemente hin- und herbewegen. Durch Wahl dieses Befehls wird das Diagramm bereinigt.

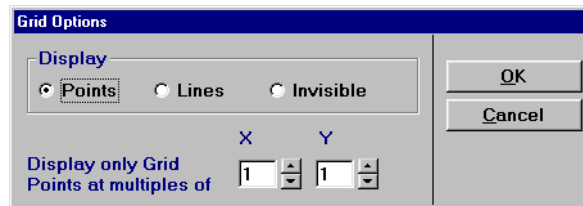


- Wählen Sie im Kombikästchen „Zoom“ einen Wert, um den Zeichenbereich zu skalieren.  
Sie können auch einen beliebigen Wert eingeben.
- Wählen Sie im Kombikästchen „Zoom“ Page Layout, um eine Druckseite zu sehen.  
Die Vergrößerung wird so gewählt, dass die erste Druckseite vollständig sichtbar ist. Die Größenverhältnisse bleiben dabei gewahrt.
- Wählen Sie im Kombikästchen „Zoom“ den Wert 100%, um zur Standardgröße zurückzukehren.

### Ändern des Rasters im Zeichenbereich:

---

- Wählen Sie **View** → **Grid**.  
Das Dialogfenster „Grid Options“ wird geöffnet.



- Klicken Sie auf die Option **Points**, um im Zeichenbereich Rasterpunkte anzeigen zu lassen.
- Klicken Sie auf die Option **Lines**, um Rasterlinien anzeigen zu lassen.
- Klicken Sie auf die Option **Invisible**, um die Rasteranzeige auszuschalten.
- Stellen Sie die Vielfachen des Rasterabstands ein, bei denen die Punkte bzw. Linien erscheinen.

Wenn zum Beispiel die Option **Points** gewählt und die Anzeige auf zwei eingestellt wurde, so ist der Abstand zwischen den Rasterpunkten gleich dem Doppelten des Standardwertes.

- Klicken Sie auf **OK**, um die Rasteroptionen einzustellen.

### Druckbereich einrichten:

---

- Stellen Sie im ASCET-Optionsfenster im Knoten „Paper Size“ (s. Seite 58) die Größe des Papiers ein.

Die Einstellung wird wirksam, wenn Sie den Blockdiagrammeditor das nächste Mal öffnen.

- Wählen Sie **View** → **Page frame Portrait**, um für den Druck das Hochformat zu wählen.
- Wählen Sie **View** → **Page frame Landscape**, um das Querformat zu wählen.

Die Standardeinstellung für das Seitenformat wird im ASCET-Optionsfenster, Knoten „Paper Size“ (Seite 58), Option „Paper Orientation“, gesetzt.

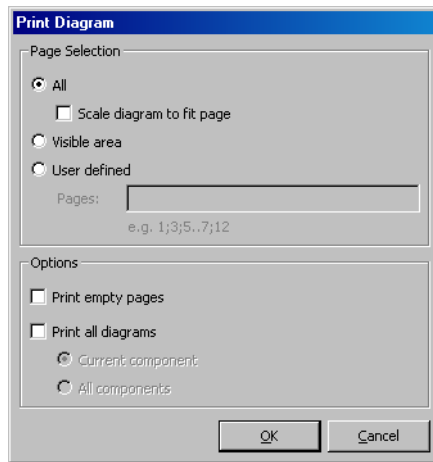
Wenn der Zeichenbereich größer als das gewählte Seitenformat ist, werden die Druckseiten im Diagramm durch gestrichelte Linien angezeigt. Außerdem können Sie im ASCET-Optionsfenster die Anzeige von Seitenzahlen im Blockdiagrammeditor einschalten (Knoten „Block Diagrams“ (Seite 56), **Show Page Number**). Linien und Seitenzahlen werden allerdings von Diagrammelementen verdeckt.

Die Farbe der Linien und der Seitenzahlen wird im ASCET-Optionsfenster, Knoten „Colors“ (Seite 57), Option „Watermark Color“, gesetzt.

## Drucken eines Blockdiagramms:

---

- Wählen Sie **View** → **Print Diagram**, um den Zeichenbereich zu drucken.  
Das Fenster „Print Diagram“ öffnet sich.



- Stellen Sie im Fenster „Print Diagram“ die Optionen für den Druck ein.
- Klicken Sie auf **OK**.  
Das Fenster „Printer Selection“ öffnet sich.
- Wählen Sie im Feld „Printers“ einen Drucker.  
Mit **Setup** können Sie die Druckereinrichtung ändern.
- Klicken Sie auf **OK**, um die Auswahl zu übernehmen.  
Die Komponente wird gemäß Ihren Einstellungen gedruckt.

Das Fenster „Print Diagrams“ enthält folgende Optionen:

- **All**  
Der gesamte Zeichenbereich wird gedruckt.
- **Scale diagram to fit page**  
Skaliert das Diagramm, so dass es auf eine Druckseite passt. Nur verfügbar, wenn **All** aktiviert ist.

- **Visible area**  
Der sichtbare Teil des Zeichenbereichs wird gedruckt.
- **User defined**  
Ein benutzerdefinierter Teil des Zeichenbereichs wird gedruckt.

#### Hinweis

Wenn **User defined** aktiviert ist, sind die Optionen **Print empty pages** und **Print all diagrams** wirkungslos.

- „Pages:“  
Eingabefeld für die Seiten, die gedruckt werden sollen. Einzelne Seiten werden durch ; getrennt, Anfang und Ende eines Bereichs durch zwei Punkte. Nur verfügbar, wenn **User defined** aktiviert ist.
- **Print empty pages**  
Leere Seiten werden gedruckt.
- **Print all diagrams**  
Alle Diagramme der Komponente werden gedruckt.
- **Current component**  
Nur die Diagramme der editierten Komponente werden gedruckt. Nur verfügbar, wenn **Print all diagrams** aktiviert ist.
- **All components**  
Auch die Diagramme von aufgenommenen Komponente werden gedruckt. Nur verfügbar, wenn **Print all diagrams** aktiviert ist.

#### *Verwendung von referenzierten Komponenten*

Es gibt mehrere Befehle, die auf referenzierte Komponenten angewandt werden können, d.h. Komponenten, die in die aktuelle Komponente aufgenommen sind. Diese haben dieselbe Funktion wie die Befehle im Menü **Component**, wirken sich jedoch nur auf die aktuell ausgewählte aufgenommene Komponente aus.

#### **Bearbeiten einer aufgenommenen Komponente:**

- Wählen Sie in der Liste „Elements“ oder im Zeichenbereich eine aufgenommene Komponente.

- Wählen Sie **Element** → **Edit Component**, um den Blockdiagrammeditor für die ausgewählte Komponente zu öffnen.

Dieser Befehl hat dieselbe Wirkung wie ein Doppelklicken auf die referenzierte Komponente.

- Wählen Sie **Element** → **Notes**, um die Hinweise für die importierte Komponente zu betrachten.

Mit diesem Befehl erhalten Sie eine Lesansicht der mit der Komponente verknüpften Hinweise, d.h. Sie können die Hinweise in dieser Ansicht nicht bearbeiten. Hinweise sind im Abschnitt „Notizen“ auf Seite 747 erläutert.

- Wählen Sie **Element** → **Export Data**, um den Datensatz der referenzierten Komponente zu exportieren.

Dieser Befehl hat dieselbe Wirkung wie **Component** → **Export Data**, bezieht sich jedoch nur auf die ausgewählte referenzierte Komponente.

## 4.2 Der Zustandsautomaten-Editor

---

Ein *Zustandsautomat* besteht aus einem Zustandsdiagramm und zusätzlichen normalen Blockdiagrammen. Das Zustandsdiagramm ist ein spezielles Blockdiagramm, in dem die Zustände als Rechtecke mit gerundeten Ecken und die Übergänge durch gerichtete Bögen dargestellt sind. Jeder Zustand kann ein *Hierarchiezustand* sein, d.h. ein Zustand, der ein anderes Zustandsdiagramm enthält. Die Aktionen und Bedingungen eines Zustandsautomaten werden in zusätzlichen Blockdiagrammen oder in ESDL spezifiziert. Zusätzliche öffentliche Methoden können in gesonderten Diagrammen spezifiziert werden. Was ein Zustandsautomat ist und wie er funktioniert, ist im Detail im Kapitel „Zustandsautomaten“ im ASCET-Referenzhandbuch beschrieben.

Das Spezifizieren von Zustandsautomaten erfolgt ähnlich wie das Spezifizieren von Blockdiagrammen, und Blockdiagramme können auch als Teile eines Zustandsautomaten verwendet werden. Viele Teile der Funktionsbeschreibung des Blockdiagrammeditors gelten ebenso für den Zustandsautomaten-Editor. Diese Teile werden hier nicht gesondert beschrieben, sodass Sie sich, bevor Sie das vorliegende Kapitel lesen, mit dem Spezifizieren von Blockdiagrammen im Blockdiagrammeditor vertraut gemacht haben sollten. Der Blockdiagrammeditor ist in Kapitel 4.1 auf Seite 186 beschrieben.

Das Spezifizieren eines Zustandsautomaten besteht aus den folgenden Schritten, die in den Abschnitten dieses Kapitels beschrieben werden:

- Zeichnen des Zustandsdiagramms (d.h. Anordnen der Zustände und Übergänge).
- Spezifizieren der Bedingungen und Aktionen.
- Zuordnen der Bedingungen und Aktionen zu den Zuständen und Übergängen im Zustandsdiagramm.
- Spezifikation von öffentlichen Methoden in einem separaten Diagramm.

### *Spezielle Menüfunktionen*

---

Die Menüfunktionen des Zustandsautomaten-Editors entsprechen im wesentlichen denen des Blockdiagrammeditors (siehe „Beschreibung der Menüfunktionen“ auf Seite 193). Nur das Menü **Diagram** enthält einige spezielle Funktionen.

- **Diagram**
  - *Add Diagram*  
Erzeugen eines neuen Diagramms.  
(*Public* → enthält nur öffentliche Methoden,  
*Actions/Conditions BDE* → enthält Aktionen u. Bedingungen in Form von Blockdiagrammen,  
*Actions/Conditions ESDL* → enthält Aktionen u. Bedingungen in ESDL)
  - *Add Trigger*  
Erzeugen eines Triggers.
  - *Add Action*  
Erzeugen einer Aktion.
  - *Add Condition*  
Erzeugen einer Bedingung.
  - *Create State Code Comments*  
Erzeugt im ESDL-Code von Zuständen und Übergängen einen Kommentar in der ersten Zeile (Seite 319).

## 4.2.1 Zeichnen des Zustandsdiagramms

---

### **Erstellen eines neuen Zustandsautomaten:**

---

- Wählen Sie im Komponentenmanager einen Ordner für den neuen Zustandsautomaten aus.

- Wählen Sie **Insert** → **State Machine**

*oder*



- klicken Sie auf die Schaltfläche **Insert State Machine**.

Ein neuer Zustandsautomat wird angelegt.

- Geben Sie einen Namen für den Zustandsautomaten ein.

- Wählen Sie **Component** → **Edit Item**

*oder*

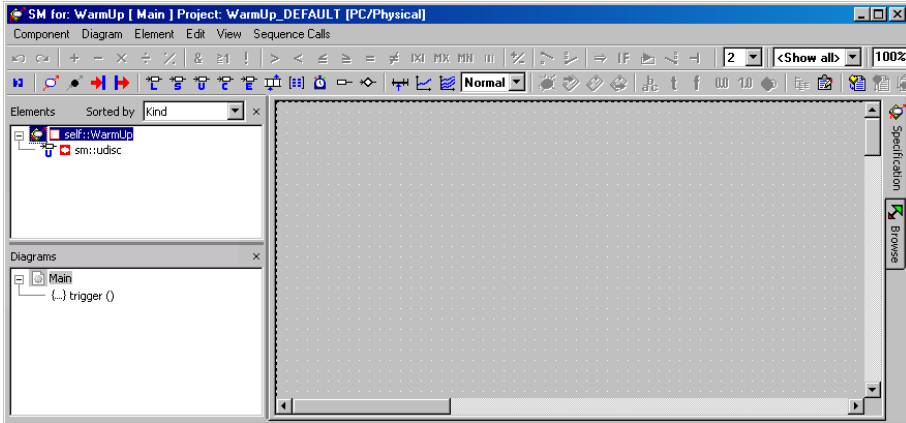
- drücken Sie <ENTER>

*oder*

- doppelklicken Sie in der Liste „Elements“ auf den Namen des Zustandsautomaten.

Der Zustandsautomaten-Editor wird geöffnet.

Bei der Standardeinstellung ist das Zustandsdiagramm geöffnet, und die Operatorschaltflächen in der Werkzeugleiste sind gesperrt. Ein Zustandsdiagramm enthält keine Operatoren.



### Anordnen der Zustände für das Diagramm:

---



- Klicken Sie auf die Schaltfläche **State** (Zustand), um den Mauszeiger mit einem neuen Zustand zu laden.
- Klicken Sie in den Zeichenbereich auf die Stelle, wo Sie den Zustand anordnen möchten.

An der Stelle, auf die Sie geklickt haben, erscheint ein Zustandssymbol.

- Wiederholen Sie diese beiden Schritte für alle anderen anzuordnenden Zustände.

Jeder Zustandsautomat muss einen Anfangszustand haben, in dem sich der Zustandsautomat befindet, wenn die Klasse zu Beginn aktiviert wird.

### Startzustand bestimmen:

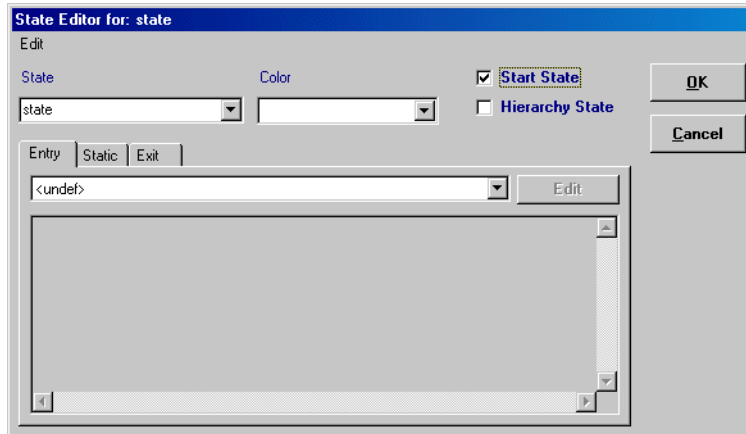
---

- Klicken Sie mit der rechten Maustaste auf den Zustand, der als Anfangszustand fungieren soll, und wählen Sie aus dem Kontextmenü **Edit State**.

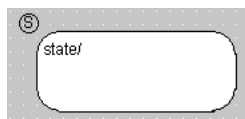
*Oder*



- Doppelklicken Sie auf den Zustand.  
Der Zustandseditor (Dialogfenster „State Editor for:“) wird geöffnet.



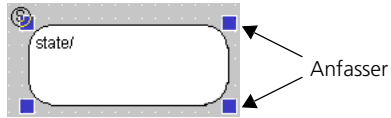
- Setzen Sie die Option **Start State** aktiv.
- Klicken Sie auf **OK**.  
An dem ausgewählten Zustand wird links oben ein eingekreistes 'S' angezeigt, um darauf hinzuweisen, dass es sich um den Anfangszustand handelt.



### Bearbeiten eines Zustands:

- Um Zustandssymbole zu bewegen, ziehen Sie sie einfach im Zeichenbereich in die gewünschte Position.  
Wenn Sie ein Symbol bewegen, folgen die an ihm beginnenden oder endenden Verbindungslinien automatisch dieser Bewegung, ihr Verlauf wird den Erfordernissen entsprechend geändert.

- Um die Größe eines Zustandssymbols zu ändern, klicken Sie einmal darauf.



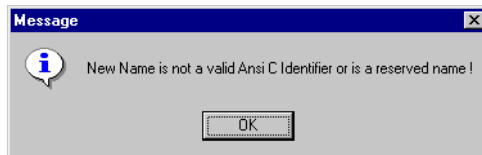
- Ziehen Sie an den Anfassern, bis das Symbol die gewünschte Größe hat.
- Sie können Zustandssymbole ebenso wie andere Diagrammelemente unter Verwendung der Zwischenablage ausschneiden, kopieren und einfügen (s. Seite 235).

Standardmäßig bekommt der erste Zustand den Namen `state`, der nächste ist `state_1`, usw. Sie können die Namen nach Belieben ändern, allerdings darf jeder Name nur einmal vorkommen und muss ANSI C-konform sein.

#### Zustände umbenennen:

- Wählen Sie im Zeichenbereich den Zustand, den Sie umbenennen wollen.
- Öffnen Sie den Zustandseditor (Seite 288).
- Tragen Sie im Feld „State“ einen Namen ein.
- Klicken Sie auf **OK**.

Wenn Sie einen nicht ANSI C-konformen Namen eingegeben haben, wird eine Fehlermeldung angezeigt.



- Bestätigen Sie die Fehlermeldung und geben Sie einen gültigen Namen ein.

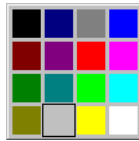
Wenn Sie einen bereits vorhandenen Namen eingegeben haben, werden automatisch die Zeichen `_n` angehängt. `n` ist dabei die kleinste für diesen Namen noch nicht vergebene Zahl.

Der neue Name wird im Zustandssymbol angezeigt.

### Zustände farbig darstellen:

---

Standardmäßig ist jeder neu hinzugefügte Zustand weiß. Sie können die Farbe wie folgt ändern:



- Öffnen Sie den Zustandseditor für den Zustand, den Sie bearbeiten wollen.
- Wählen Sie aus dem Kombikästchen „Color“ eine Farbe aus.

Das Zustandssymbol wird in der gewählten Farbe dargestellt.

### Layout eines Zustands kopieren:

---

Sie können das Layout eines Zustands, also seine Größe und Farbe, einem anderen Zustand zuweisen.

- Klicken Sie mit der rechten Maustaste in den Zustand, dessen Layout Sie kopieren wollen.
  - Wählen Sie aus dem Kontextmenü **State Layout** → **Copy Layout**.
- Größe und Farbe des Zustands werden in die Zwischenablage kopiert.
- Klicken Sie mit der rechten Maustaste in den Zustand, dem Sie das Layout zuweisen wollen.
  - Wählen Sie aus dem Kontextmenü **State Layout** → **Paste Layout**.

Dem Zustand wird das Layout zugewiesen, er hat jetzt dieselbe Größe und Farbe wie der erste.

Neben den Zuständen können Sie im Diagramm Knoten (s. Abschnitt „Knoten“ im ASCET-Referenzhandbuch) verwenden.

## Knoten erstellen:

---



- Klicken Sie auf die Schaltfläche **Junction** (Knoten), um den Mauszeiger mit einem Knoten zu laden.
- Klicken Sie in den Zeichenbereich auf die Stelle, wo Sie den Knoten anordnen möchten. An der Stelle, auf die Sie geklickt haben, erscheint ein unbeschrifteter Kreis als Symbol für den Knoten.
- Wiederholen Sie diese beiden Schritte, wenn Sie weitere Knoten benötigen.

## Knoten bearbeiten:

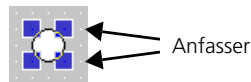
---

Anders als bei Zuständen können Sie einem Knoten keine Aktionen zuordnen. Sie können ihn lediglich verschieben und die Größe und Farbe verändern.

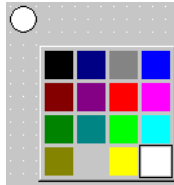
- Um Knoten zu bewegen, ziehen Sie sie einfach im Zeichenbereich in die gewünschte Position.  
Wenn Sie einen Knoten bewegen, folgen die an ihm beginnenden oder endenden Verbindungslinien automatisch dieser Bewegung, ihr Verlauf wird den Erfordernissen entsprechend geändert.
- Wenn sich ein Knoten und ein Zustand oder zwei Knoten überlappen, werden beide Umrandungen rot dargestellt. Ziehen Sie eins der Objekte so weit weg, dass sie nebeneinander liegen.



- Um die Größe eines Knotens zu ändern, klicken Sie einmal darauf.



- Ziehen Sie an den Anfassern, bis das Symbol die gewünschte Größe hat.
- Sie können Knoten ebenso wie andere Diagrammelemente unter Verwendung der Zwischenablage ausschneiden, kopieren und einfügen (s. Seite 235).
- Um die Farbe des Knotens zu ändern, klicken Sie mit der rechten Maustaste auf den Knoten und wählen Sie **Edit Color** aus dem Kontextmenü.



Ein Auswahlfeld mit den möglichen Farben öffnet sich.

- Wählen Sie eine Farbe.
- Der Knoten wird in der gewünschten Farbe dargestellt.

### Übergang herstellen:



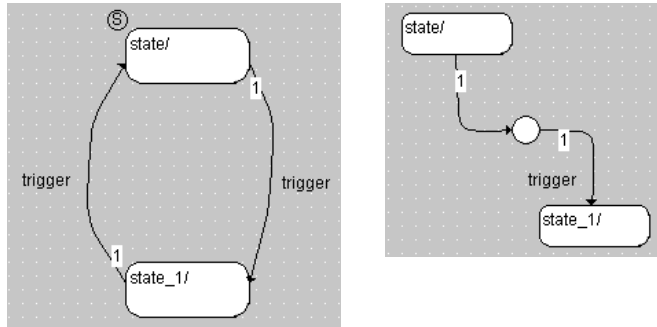
- Klicken Sie auf die Schaltfläche **Connect** oder
- klicken Sie mit der rechten Maustaste in den Zeichenbereich (nicht auf einen Zustand).  
Der Verbindungsmodus wird aktiviert, der Cursor verwandelt sich in ein Fadenkreuz.
- Klicken Sie in das Zustandssymbol oder den Knoten, an dem die Verbindung beginnen soll, und ziehen Sie den Mauszeiger zum nächsten Zustand oder Knoten.  
Es wird eine der Bewegung des Mauszeigers folgende Linie gezogen. Die Linie beginnt am Rand des Symbols, in das Sie geklickt haben.

#### Hinweis

*Es ist nicht möglich, einen Übergang aus einem Knoten in sich selbst oder in einen anderen Knoten herzustellen.*

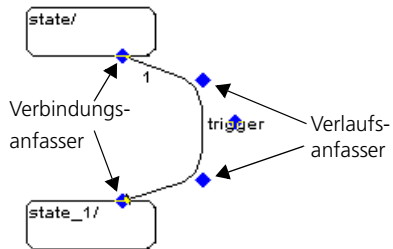
- Klicken Sie in das Symbol, mit dem Sie eine Verbindung herstellen möchten.

Zwischen den zwei Symbolen wird ein Übergang gezeichnet. Er ist mit einer Pfeilspitze versehen und zeigt vom Symbol des ersten Zustands/Knotens zum Symbol des zweiten Zustands/Knotens.



### Verlauf eines Übergangs ändern:

- Klicken Sie auf den Übergang, damit die Verlaufs- und Verbindungsanfasser im Diagramm erscheinen.



- Ziehen Sie die Verlaufsanfasser, um den Verlauf des Übergangs zu ändern.

- Ziehen Sie die Verbindungsanfasser, um die Lage der Verbindungsstelle am Zustandssymbol zu ändern.

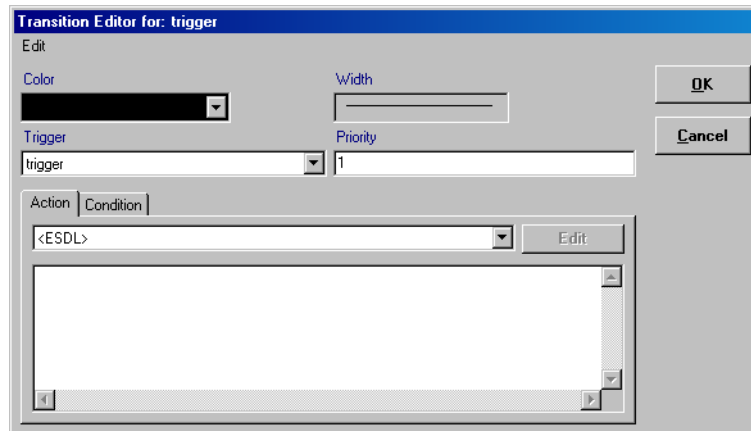
Sie können eine vorhandene Verbindungsstelle einfach durch Ziehen der Verbindungsanfasser einem anderen Quellen- oder Zielzustand zuordnen.

### Aussehen eines Übergangs ändern:

- Klicken Sie mit der rechten Maustaste auf einen Übergang und wählen Sie aus dem Kontextmenü **Edit Transition**

oder

- doppelklicken Sie auf den Übergang.  
Der Übergangseditor (Dialogfenster „Transition Editor“) wird geöffnet.



- Wählen Sie aus dem Kombikästchen „Color“ eine Farbe aus.
- Wählen Sie aus dem Kombikästchen „Width“ die Linienstärke aus.



- Klicken Sie auf **OK**, um die von Ihnen gewählten Optionen zu bestätigen.

Der Übergang wird in der gewählten Farbe und mit der gewählten Linienstärke dargestellt.

Standardmäßig ist in einem neuen Zustandsautomaten ein Trigger angelegt. Sie können weitere Trigger hinzufügen, sollten dabei aber beachten, dass die Verwendung mehrerer Trigger größeren Programmcode zur Folge hat (siehe „Optimiert für Codegröße“ im ASCET-Referenzhandbuch).

#### **Einfügen eines Triggers:**

---

- Wählen Sie **Diagram** → **Add Trigger**, um einen neuen Trigger zu erzeugen.

Der Trigger wird zur Fläche „Diagrams“ hinzugefügt. Sein Name ist hervorgehoben, um ein Bearbeiten an Ort und Stelle zu ermöglichen.

- Geben Sie einen Namen für den Trigger ein und drücken Sie <ENTER>.

### 4.2.2 Hierarchiezustände

---

Zustandsdiagramme können hierarchisch sein (s. „Hierarchie“ im ASCET-Referenzhandbuch), d.h. ein Zustand kann ein anderes Zustandsdiagramm enthalten. Wenn der Zustandsautomat in einen Hierarchiezustand eintritt, beginnt er mit dem Anfangszustand des in dem Hierarchiezustand enthaltenen Unterdiagramms. Falls jedoch der Hierarchiezustand eine Geschichte hat (s. „Geschichte“ im ASCET-Referenzhandbuch), so wird bei einem erneuten Eintritt in den Hierarchiezustand derjenige Unterzustand aktiviert, in dem sich das Unterdiagramm befand, als der Hierarchiezustand zum letzten Mal verlassen wurde.

Dabei wird zwischen geschlossenen und offenen Hierarchiezuständen unterschieden. Im Gegensatz zum geschlossenen Hierarchiezustand, wo das Unterdiagramm in einer neuen Zeichenebene erstellt wird, befindet sich beim offenen Hierarchiezustand das Unterdiagramm in der gleichen Zeichenebene. Die Funktionalität beider Hierarchiearten ist jedoch die gleiche. Somit entfällt unter Verwendung von offenen Hierarchiezuständen das Wechseln der verschiedenen Zeichenebenen.

Innerhalb eines Zustandsdiagramms können gleichzeitig offene und geschlossene Hierarchiezustände vorhanden sein, ein einzelner Zustand kann jedoch nur eine Hierarchieart verwenden. Eine fehlerhafter Aufbau, bei dem Zustände in der Zeichenebene überlappen, wird durch eine Farbänderung an den Zustandssymbolen angezeigt.

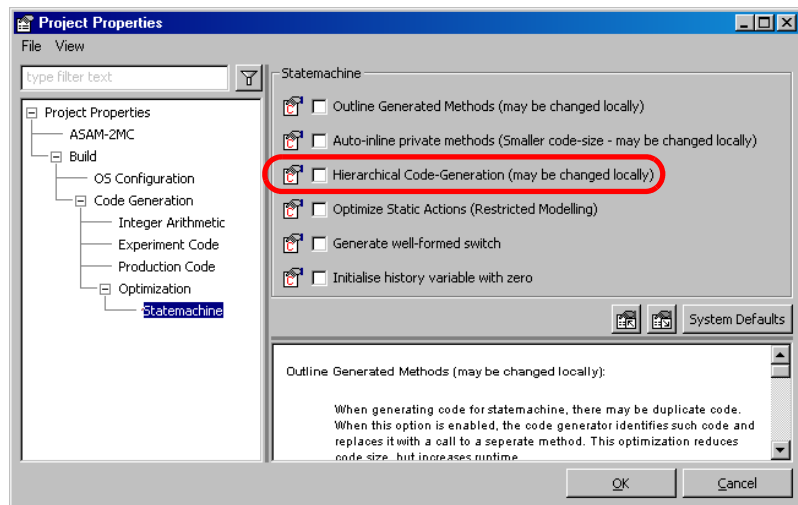


Der Code für einen hierarchischen Zustandsautomaten kann wahlweise flach (eine einzige `switch`-Anweisung für alle (Basis-)Zustände) oder hierarchisch (gemäß der Hierarchie verschachtelte `switch`-Anweisungen) generiert werden; dies kann eine merkliche Reduzierung der Codegröße bedeuten (siehe Abschnitt „Hierarchische Codegenerierung“ im ASCET-Referenzhandbuch).

### Hierarchische Codegenerierung ein-/ausschalten:

Um hierarchische Codegenerierung ein-/auszuschalten, gehen Sie wie folgt vor.

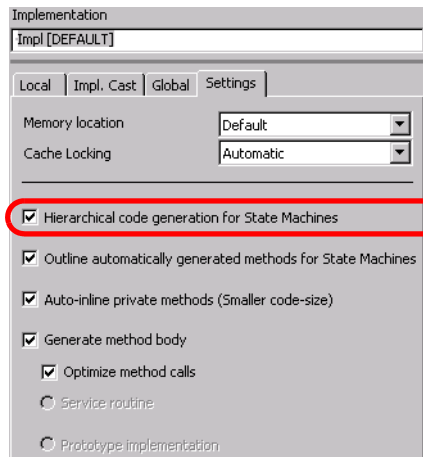
1. ausschalten
  - Öffnen Sie das Projekt, zu dem Ihr Zustandsautomat gehört.  
Das kann auch das Standardprojekt sein.
  - Öffnen Sie im Projekteditor das Fenster „Project Properties“ (Seite 439).
  - Deaktivieren Sie im Knoten „Statemachine“ (Seite 453) die Option **Hierarchical Code-Generation (may be changed locally)**, um für alle Zustandsautomaten im Projekt die hierarchische Codegenerierung auszuschalten.



Die Einstellungen der einzelnen Zustandsautomaten sind in diesem Fall nicht von Bedeutung.

## 2. einschalten

- Aktivieren Sie die Option **Hierarchical Code-Generation (may be changed locally)**, um für alle Zustandsautomaten im Projekt die hierarchische Codegenerierung einzuschalten.
- Schließen Sie das Fenster „Project Properties“.  
Die globale Einstellung in den Projekteigenschaften allein reicht nicht, damit hierarchischer Code generiert wird.
- Öffnen Sie mit **Component** → **Edit Implementation** den Implementierungseditor jedes Zustandsautomaten im Projekt, für den hierarchischer Code generiert werden soll.
- Aktivieren Sie im Register „Settings“ die Option **Hierarchical code generation for State Machines**.



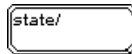
- Schließen Sie den Implementierungseditor.  
Nur wenn beide Optionen aktiviert sind, findet hierarchische Codegenerierung statt.  
Sie können mit Hilfe dieser Option einzelne Zustandsautomaten von der hierarchischen Codegenerierung ausnehmen.

### Hinzufügen eines geschlossenen Hierarchiezustands:

---

- Fügen Sie den Zustand, der ein Hierarchiezustand werden soll, dem übergeordneten Diagramm hinzu.
- Klicken Sie mit der rechten Maustaste auf den Zustand und wählen Sie aus dem Kontextmenü **Edit State**.
- Aktivieren Sie im Zustandseditor die Option **Hierarchy State**.
- Klicken Sie auf **OK**.

Dadurch wird der Zustand zu einem Hierarchiezustand, was durch die doppelten Linien des Zustandssymbols angezeigt wird.



### Wechseln zwischen Hierarchie-Ebenen:

---

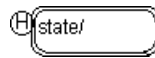
- Doppelklicken Sie auf einen Hierarchiezustand *oder*
- klicken Sie mit der rechten Maustaste auf den Hierarchiezustand und wählen Sie aus dem Kontextmenü **Next Level**.  
Sie bewegen sich eine Ebene abwärts, d.h. es wird das Zustandsdiagramm dargestellt, das in dem Hierarchiezustand enthalten ist. Sie können hier den Inhalt des Hierarchiezustands spezifizieren.
- Wenn Sie sich in einem Hierarchiezustand befinden, doppelklicken Sie in den Zeichenbereich (nicht auf ein Diagrammelement).  
Sie bewegen sich wieder eine Ebene aufwärts.

### Einrichten eines Hierarchiezustands mit einer Geschichte:

---

- Klicken Sie mit der rechten Maustaste auf den Hierarchiezustand und wählen Sie aus dem Kontextmenü **History**.

Der Hierarchiezustand hat nunmehr eine Geschichte. Das heißt, beim Eintritt in den Hierarchiezustand wird der Unterzustand aktiviert, in dem er sich befand, als er verlassen wurde, auch wenn das nicht der Startzustand der Hierarchie ist.



Mit Anschlüssen an Symbolen von Hierarchiezuständen ist es möglich, Zustände zu verbinden, die sich nicht in derselben Hierarchie befinden. Eine Verbindung zwischen Zuständen über einen Anschluss ist dasselbe, als wenn die Zustände direkt miteinander verbunden wären.

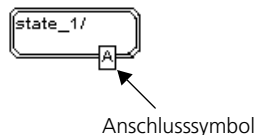
### Hinzufügen eines Anschlusses zu einem Hierarchiezustand:

---

- Klicken Sie mit der rechten Maustaste auf einen Hierarchiezustand.
- Wählen Sie aus dem Kontextmenü **Add Pin**.

Am Zustandssymbol wird ein Anschluss mit einem Standardnamen erzeugt. Ein entsprechender Anschluss wird innerhalb der Hierarchie gezeichnet.

Sie können nun einen Übergang mit dem Anschluss am Zustandssymbol verbinden und dann den Anschluss innerhalb der Hierarchie mit einem Zustand verbinden.

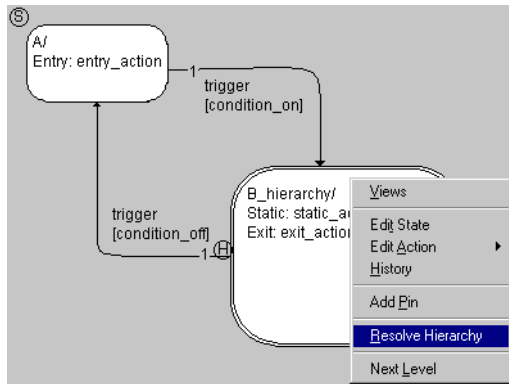


### Auflösen eines Hierarchiezustandes:

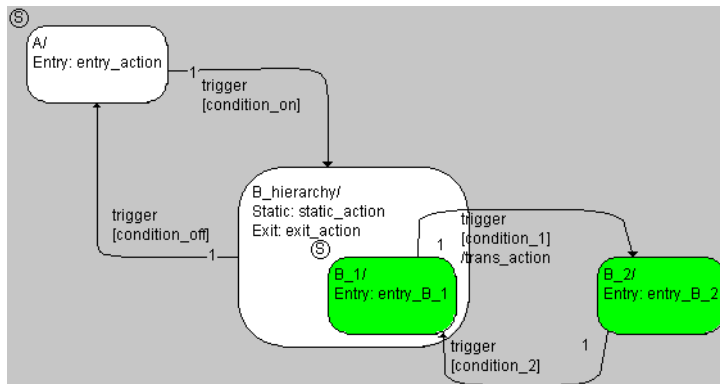
---

- Klicken Sie mit der rechten Maustaste auf einen Hierarchiezustand.

- Wählen Sie aus dem Kontextmenü **Resolve Hierarchy**.



Alle Elemente und ihre Verknüpfungen werden aus dem Hierarchiezustand in die übergeordnete Zeichenebene an die gleiche Position kopiert. Dabei entsteht ein zweites unabhängiges Zustandsdiagramm mit Anfangszustand. Die doppelten Linien, die den geschlossenen Hierarchiezustand markiert haben, werden entfernt. Wenn der ausgewählte Zustand eine Geschichte hatte, wird auch diese entfernt.



Der Zustandsautomat ist nach dem Auflösen der Hierarchie in den meisten Fällen unübersichtlich, auch hat er (meist) nicht mehr dieselbe Funktionalität wie vorher. So waren in dem hier gezeigten Beispiel die Zustände `B_1` und `B_2` anfangs im geschlossenen Hierarchiezustand `B_hierarchy` enthalten. Jetzt ist nur noch `B_1` in `B_hierarchy` enthalten, wenn also der Übergang von `B_1` nach `B_2` (oder umgekehrt) stattfindet, werden andere Aktionen ausgeführt (siehe Abschnitt „Semantik: Hierarchische Zustandsautomaten“ im ASCET-Referenzhandbuch).

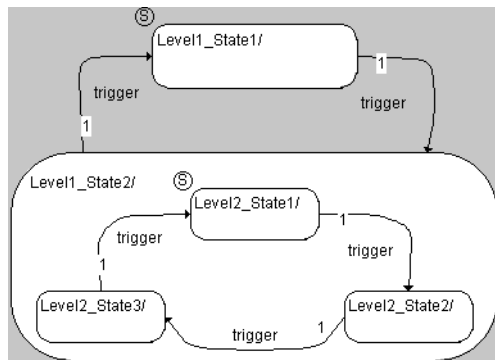
Um die gewünschte Funktionalität herzustellen und das Diagramm übersichtlicher zu machen, ist eine Nachbearbeitung erforderlich.

### *Offener Hierarchiezustand*

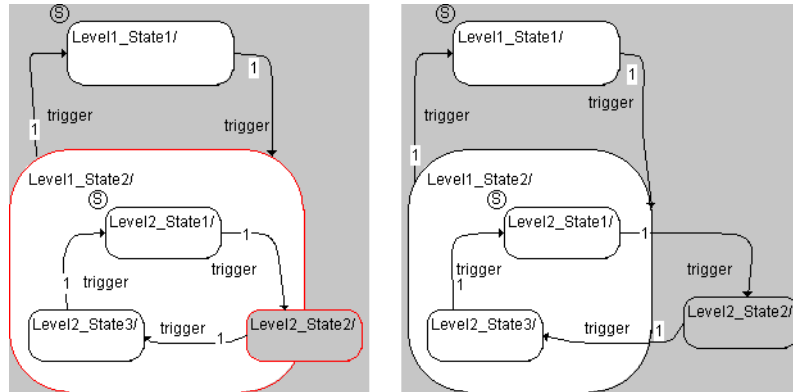
#### **Hinzufügen eines offenen Hierarchiezustandes:**

- Vergrößern Sie die Fläche des Zustandes, der die Hierarchie enthalten soll, durch Ziehen an den Anfassern.
- Fügen Sie alle benötigten Zustände und Übergänge des Unterdiagramms in das Zustandssymbol des Hierarchiezustandes ein.
- Wählen Sie einen Zustand aus, der als Anfangszustand für das Unterdiagramm fungieren soll.

Die Vorgehensweise bei der Erstellung eines Zustandsdiagramms innerhalb eines Zustandsymbols ist mit der Erstellung von einem Zustandsdiagramm im Zeichenbereich identisch.



Übergänge von einem Unterzustand in einen Zustand außerhalb der Hierarchie sind möglich. Dabei ist allerdings zu beachten, dass bei einem Übergang zwischen verschiedenen Hierarchien andere Aktionen abgearbeitet werden als bei einem Übergang innerhalb derselben Hierarchie (siehe auch Beispiel 12 und Beispiel 15 im ASCET-Referenzhandbuch). Ein Zustand wird als außerhalb einer Hierarchie betrachtet, wenn er ganz oder teilweise neben dem Hierarchiezustand liegt (Zustand `Level12_State2` in der Abbildung).



Wie beim geschlossenen Hierarchiezustand können Sie einen Hierarchiezustand mit einer Geschichte einrichten.

#### 4.2.3 Spezifizieren von Bedingungen und Aktionen

Jeder Zustand kann eine Eintrittsaktion, eine statische Aktion und eine Austrittsaktion haben, und mit jedem Übergang sind ein Trigger, eine Bedingung und eine Übergangsaktion verknüpft. Bedingungen (s. „Bedingungen“ im ASCET Referenzhandbuch) und Aktionen (s. „Aktionen“ im ASCET Referenzhandbuch) sind private Methoden, und sie werden auf die gleiche Weise spezifiziert wie die Methoden von anderen Klassen.

Eine Bedingung wird jedesmal geprüft, wenn sich der Zustandsautomat in einem Zustand befindet, der einen Übergang aufweist, mit dem diese Bedingung verknüpft ist. Wenn die Bedingung erfüllt (wahr) ist, findet der Übergang statt. Deshalb hat eine Bedingung stets den Rückgabewert `true` oder `false`. Eintritts-, Austritts- und Übergangsaktionen werden – sofern vorhanden – jedesmal ausgeführt, wenn ein Übergang stattfindet, statische Aktionen werden ausgeführt, wenn kein Übergang stattfindet. Eine ausführliche Beschreibung der Funktionsweise eines Zustandsautomaten finden Sie in Kapitel 2.5.2 bis Kapitel 2.5.6 im ASCET-Referenzhandbuch.

Bedingungen und Aktionen können zum einen in eigenen Diagrammen (ActionCondition-Diagramm) in Form von Blockdiagrammen oder ESDL-Code spezifiziert werden. Der Zustandsautomat enthält dann mindestens zwei Diagramme. Zum anderen können Sie Bedingungen und Aktionen als ESDL-Code auch direkt im Zustands- oder Übergangseditor eingeben; ein separates Diagramm ist dann nicht erforderlich.

Statische Aktionen von Hierarchiezuständen können bezüglich Codegröße optimiert werden; siehe Abschnitt „Statische Aktionen von Hierarchiezuständen“ in Kapitel 2.5.8 des ASCET-Referenzhandbuchs.

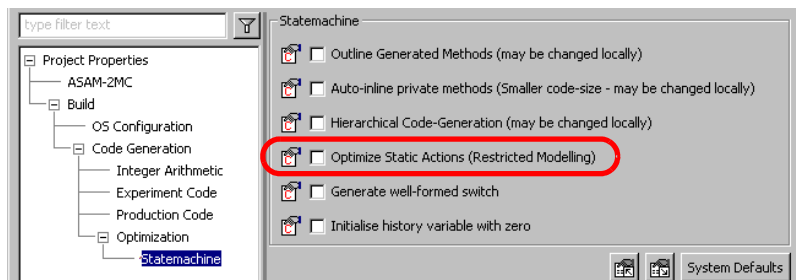
### Optimierung statischer Aktionen ein-/ausschalten:

Um die Optimierung statischer Aktionen von Hierarchiezuständen ein-/auszuschalten, gehen Sie wie folgt vor.

- Öffnen Sie das Projekt, zu dem Ihr Zustandsautomat gehört.  
Das kann auch das Standardprojekt sein.
- Öffnen Sie im Projekteditor das Fenster „Project Properties“ (Seite 439).

#### 1. ausschalten

- Deaktivieren Sie im Knoten „Statemachine“ (Seite 453) die Option **Optimize Static Action (Restricted Modeling)**, um die Optimierung für alle Zustandsautomaten im Projekt auszuschalten.





## 2. einschalten

### Hinweis

Wenn Sie die Optimierung einschalten, gilt folgende Einschränkung bei der Modellierung:

Hat ein Unterzustand des Zustandsautomaten einen direkten Übergang aus seinem Hierarchiezustand heraus, muss dieser Übergang von allen Übergängen aus dem Unterzustand die höchste Priorität haben.

- Aktivieren Sie die Option **Optimize Static Action (Restricted Modeling)**, um die Optimierung für alle Zustandsautomaten im Projekt einzuschalten.

Die Optimierung der statischen Aktionen findet statt. Die dadurch entstehenden Änderungen bei der Codegenerierung können das Verhalten des Zustandsautomaten beeinflussen. Prüfen Sie das Verhalten sorgfältig, wenn Sie die Option für einen existierenden Zustandsautomaten aktivieren.

### Bedingungen und Aktionen in separaten Diagrammen

Sie können Bedingungen und Aktionen als Blockdiagramme oder ESDL-Code in eigenen `ActionCondition`-Diagrammen spezifizieren. Ein Zustandsautomat kann eine beliebige Anzahl `ActionCondition`-Diagramme enthalten; ein solches Diagramm enthält entweder Blockdiagramme oder ESDL-Code.

Zuerst legen Sie die erforderlichen Diagramme an. Danach können die Bedingungen und Aktionen wie normale Blockdiagramme oder ESDL-Code spezifiziert werden. Das Spezifizieren von Blockdiagrammen ist in „Erzeugen von Blockdiagrammen“ auf Seite 217 erläutert, das Spezifizieren von ESDL-Code in „Der ESDL-Editor“ auf Seite 351.

### **Erzeugen eines Diagramms für Aktionen/Bedingungen:**

---

#### 1. Blockdiagramm

- Wählen Sie **Diagram** → **Add Diagram** → **Actions/Conditions BDE**

oder

## 2. ESDL

- klicken Sie mit der rechten Maustaste in die Fläche „Diagrams“ und wählen Sie aus dem Kontextmenü **Add Diagram** → **Actions/Conditions BDE**.

Das Diagramm `ActionCondition_BDE` wird erzeugt.

- Geben Sie dem Diagramm im Feld „Diagrams“ den von Ihnen gewünschten Namen.

- Wählen Sie **Diagram** → **Add Diagram** → **Actions/Conditions ESDL**

*oder*

- klicken Sie mit der rechten Maustaste in die Fläche „Diagrams“ und wählen Sie aus dem Kontextmenü **Add Diagram** → **Actions/Conditions ESDL**.

Das Diagramm `ActionCondition_ESDL` wird erzeugt.

- Geben Sie dem Diagramm im Feld „Diagrams“ den von Ihnen gewünschten Namen.

### **Hinzufügen einer Bedingung:**

---

- Wählen Sie im Feld „Diagrams“ das Diagramm, in dem Sie die Bedingung anlegen wollen.

- Wählen Sie **Diagram** → **Add Condition**

*oder*

- klicken Sie mit der rechten Maustaste in die Fläche „Diagrams“ und wählen Sie aus dem Kontextmenü **Add Condition**.

Es wird eine neue Bedingung erzeugt, und ihr Name wird in der Fläche „Diagrams“ angezeigt.

Eine Bedingung hat einen logischen Rückgabewert. Der Rückgabewert für die Bedingung wird automatisch erzeugt. Wenn Sie das Diagramm öffnen, zu dem die Bedingung gehört, wird der Rückgabewert in der Fläche „Elements“ angezeigt.

- Geben Sie einen Namen für die Bedingung ein und drücken Sie <ENTER>.

Eine Aktion hat standardmäßig keine Argumente und keinen Rückgabewert. Daher werden für eine Aktion keine Schnittstellenelemente erzeugt, doch ansonsten erfolgt das Hinzufügen von Aktionen analog zum Hinzufügen von Bedingungen über das Menü **Diagram → Add Action** bzw. das Kontextmenü **Add → Action**.

### **Öffnen eines Diagramms für Aktionen/Bedingungen:**

---

- Wählen Sie im Feld „Diagrams“ das **ActionCondition**-Diagramm, das Sie öffnen wollen.
- Doppelklicken Sie auf das Diagramm

*oder*

- wählen Sie **Diagram → Load Diagram**.

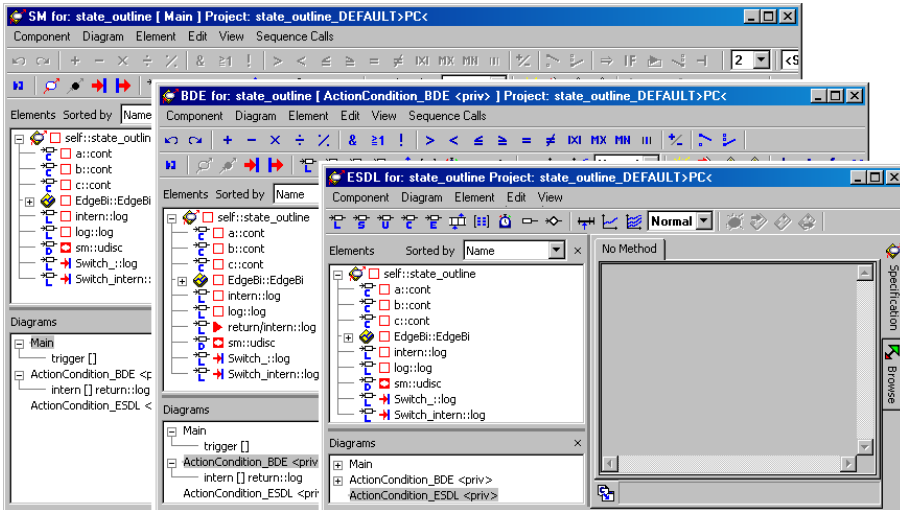
Wenn das aktuell angezeigte Diagramm unsichere Änderungen enthält, werden Sie gefragt, ob Sie die Änderungen sichern wollen.

- Bestätigen Sie die Abfrage mit **Yes**, wenn Sie die Änderungen sichern wollen,

*oder*

- Bestätigen Sie die Abfrage mit **No**, wenn Sie die Änderungen verwerfen wollen.

Das `ActionCondition`-Diagramm wird in einem neuen Editorfenster geladen.



Die Bedingungen und Aktionen werden wie alle anderen Blockdiagramme oder ESDL-Komponenten spezifiziert; siehe Kapitel 4.1 „Der Blockdiagramm-editor“ und 4.4 „Der ESDL-Editor“. Die Menüfunktionen und Schaltflächen für die Codegenerierung sind allerdings deaktiviert, diese sind nur im Zustandsautomaten-Editor verfügbar.

Der Code für die so erzeugten Aktionen und Bedingungen wird wahlweise in Form von eigenen privaten Funktionen generiert oder während der Codegenerierung an ort und Stelle eingefügt (*Auto-Inlining*; siehe „Den Zustandsautomaten optimieren“ im ASCET-Referenzhandbuch).

### Auto-Inlining ein-/ausschalten:

Um Auto-Inlining ein-/auszuschalten, gehen Sie wie folgt vor.

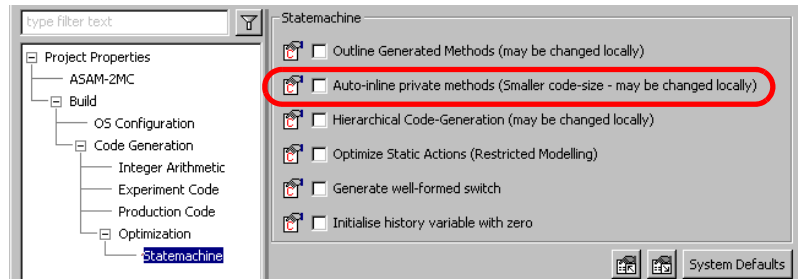
- Öffnen Sie das Projekt, zu dem Ihr Zustandsautomat gehört.

Das kann auch das Standardprojekt sein.

- Öffnen Sie im Projekteditor das Fenster „Project Properties“ (Seite 439).

## 1. ausschalten

- Deaktivieren Sie im Knoten „Statemachine“ (Seite 453) die Option **Auto-inline private methods (Smaller code size - may be changed locally)**, um Auto-Inlining für alle Zustandsautomaten im Projekt auszuschalten.



Die Einstellungen der einzelnen Zustandsautomaten sind in diesem Fall nicht von Bedeutung.

## 2. einschalten

- Aktivieren Sie die Option **Auto-inline private methods (Smaller code size - may be changed locally)**, um Auto-Inlining für alle Zustandsautomaten im Projekt einzuschalten.
- Schließen Sie das Fenster „Project Properties“.  
Die globale Einstellung in den Projekteigenschaften allein reicht nicht, damit Auto-Inlining stattfindet.
- Öffnen Sie mit **Component** → **Edit Implementation** den Implementierungseeditor jedes Zustandsautomaten im Projekt, für den Auto-Inlining eingeschaltet werden soll.
- Aktivieren Sie im Register „Settings“ die Option **Auto-inline private methods (Smaller code size)**.

- Schließen Sie den Implementierungseditor.  
Nur wenn beide Optionen aktiviert sind, kann Auto-Inlining stattfinden.  
Sie können mit Hilfe dieser Option einzelne Zustandsautomaten vom Auto-Inlining ausnehmen.

### *Bedingungen und Aktionen verwenden*

Aktionen und Bedingungen aus einem ActionCondition-Diagramm müssen den Übergängen oder Zuständen, in denen sie verwendet werden sollen, explizit zugewiesen werden.

### **Trigger, Priorität, Bedingung und Aktion zu einem Übergang zuweisen:**

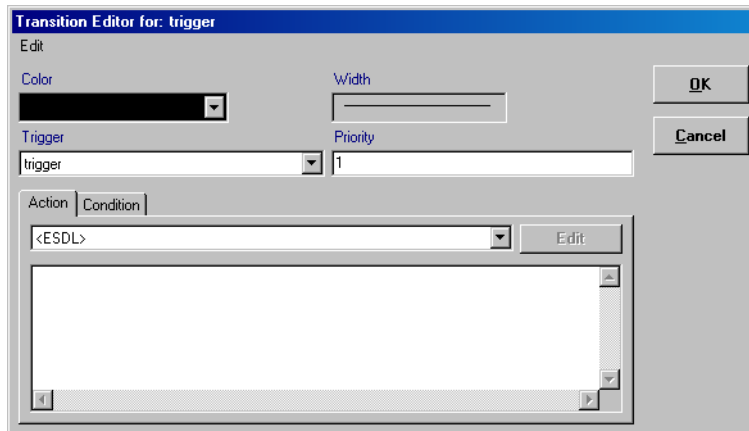
Jeder Übergang muss einen Trigger und eine Priorität haben, jedoch sind die Bedingung und die Übergangsaktion optional.

- Doppelklicken Sie auf den Übergang bzw. das Übergangsegment

oder

- klicken Sie mit der rechten Maustaste auf den Übergang/das Segment und wählen Sie aus dem Kontextmenü **Edit Transition**.

Der Übergangseditor wird geöffnet.



- Geben Sie im Feld „Priority“ eine Zahl ein, um dem Übergang eine Priorität zuzuweisen.

#### **Hinweis**

*Je höher die Zahl gewählt ist, um so höher ist die Priorität des Übergangs.*

Falls mehrere Übergänge/Segmente mit dem gleichen auslösenden Ereignis vorhanden sind, die von einem Zustand oder Knoten wegführen, so werden ihre Bedingungen in der Reihenfolge der Priorität geprüft. Jeder von einem Zustand/Knoten wegführende Übergang mit dem gleichen Trigger muss eine andere Priorität haben.

- Wählen Sie aus dem Kombikästchen „Trigger“ einen Trigger aus.

#### **Hinweis**

*Wenn ein Übergang durch einen Knoten in zwei Übergangsegmente geteilt ist, darf nur eines der Segmente einen Trigger haben.*

Alle im Zustandsdiagramm definierten Trigger sind in diesem Feld aufgelistet.

- Wählen Sie im Register „Action“ aus dem Kombikästchen eine Übergangsaktion aus.

#### **Hinweis**

*Ein in einen Knoten führendes Übergangsegment kann keine Aktion haben. Das Kombikästchen ist deaktiviert.*

Alle in einem ActionCondition-Diagramm spezifizierten Aktionen sind im Kombikästchen aufgelistet und können als Übergangsaktionen zugewiesen werden.

- Wählen Sie im Register „Condition“ aus dem Kombikästchen eine Bedingung aus.

Alle in einem `ActionCondition`-Diagramm spezifizierten Bedingungen sind in diesem Kästchen aufgelistet.

#### **Hinweis**

*Wenn keine Bedingung spezifiziert wird, so bedeutet dies, dass die Bedingung immer wahr ist, d.h. der Übergang findet immer statt.*

- Klicken Sie auf **OK**.

Der Name des Triggers, die Priorität und die Namen von Bedingung und Aktion des Übergangs/Segments werden im Diagramm dargestellt.

Jedem Zustand kann eine Eintritts-, eine statische und eine Austrittsaktion zugeordnet sein. Diese sind alle optional.

#### **Zuweisen von Aktionen zu einem Zustand:**

---

- Doppelklicken Sie auf einen Zustand

*oder*

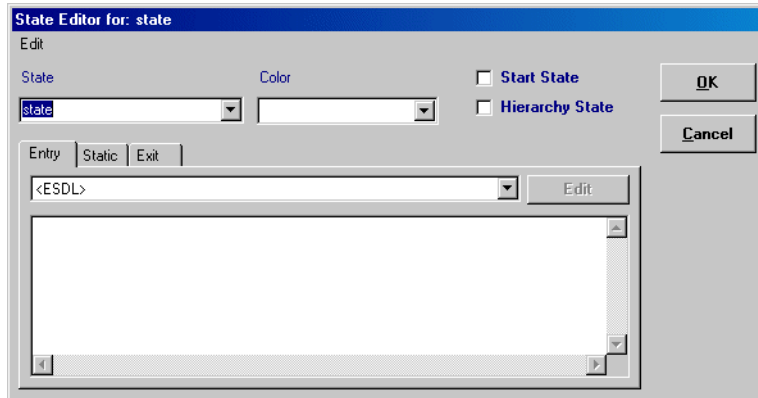
- klicken Sie mit der rechten Maustaste auf den Zustand und wählen Sie aus dem Kontextmenü **Edit State**

*oder*



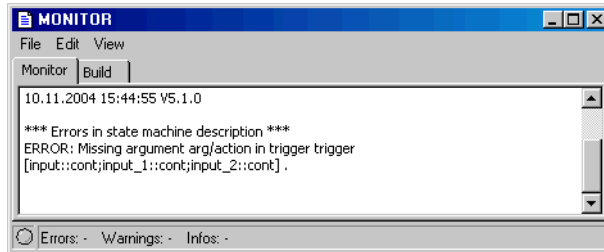
- wählen Sie aus dem Kontextmenü **Edit Action** → **<action>**, um eine bestimmte Aktion zu bearbeiten.

Der Zustandseditor wird geöffnet.



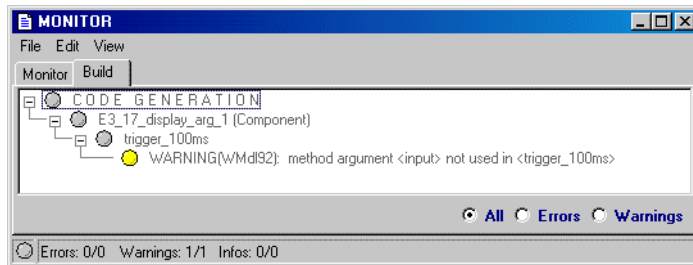
- Wählen Sie im Register „Entry“ aus dem Kombikästchen eine Eintrittsaktion aus.
- Wählen Sie in den Registern „Static“ bzw. „Exit“ die statische Aktion bzw. die Austrittsaktion aus.  
Alle Aktionen können hier verwendet werden.
- Klicken Sie auf **OK**.  
Die Namen der Aktionen werden im Zustandsymbol angezeigt.

Wenn Sie Aktionen oder Bedingungen mit Argumenten zugewiesen haben, wird beim Verlassen des Zustands- oder Übergangseditors überprüft, ob alle relevanten Trigger entsprechende Argumente mit gleichem Namen und Typ haben. Ist das nicht der Fall, wird im ASCET-Monitorfenster eine Fehlermeldung angezeigt.



Erzeugen Sie die in der Fehlermeldung angegebenen Triggerelemente wie auf Seite 322 beschrieben.

Haben Sie umgekehrt ein Triggerargument erzeugt, das in keiner Aktion oder Bedingung verwendet wird, hat das keinen Einfluss auf die Funktion des Zustandsautomaten. Bei der Codegenerierung wird lediglich im Register „Build“ des ASCET-Monitorfensters eine Warnung angezeigt.



Wenn Sie Aktionen oder Bedingungen aus einem ActionCondition-Diagramm zugewiesen haben, wird die Schaltfläche **Edit** in den Registern aktiviert. Sie können diese benutzen, um die zugewiesene Aktion/Bedingung zu bearbeiten.



## Bedingungen/Aktionen bearbeiten:

---

- Öffnen Sie den Zustands- oder Übergangseditor.
- Weisen Sie in einem der Register eine Aktion oder eine Bedingung zu.
- Um die Aktion oder Bedingung zu bearbeiten, klicken Sie auf **Edit**.
- Bestätigen Sie im Fenster „Confirm“ das Abspeichern Ihrer Eingaben im Zustands- oder Übergangseditor.  
Der Zustands-/Übergangseditor wird geschlossen, die Einstellungen übernommen.  
Ein weiteres Fenster „Confirm“ wird angezeigt.
- Bestätigen Sie im neuen Fenster das Abspeichern Ihrer Eingaben im Zustandsdiagramm.  
Das Zustandsdiagramm wird gespeichert.  
Das Diagramm, in dem die zugewiesene Aktion oder Bedingung spezifiziert wurde, wird in einem eigenen Editorfenster geöffnet.

### *Bedingungen und Aktionen im Zustandsdiagramm*

---

Es ist möglich, den ESDL-Code für eine Bedingung oder Aktion im Zustandsdiagramm direkt in den Übergangs- oder Zustandseditor einzugeben. In diesem Falle ist es nicht erforderlich, eine Spezifikation in einem separaten Diagramm zu erstellen, die Schaltfläche **Edit** ist inaktiv.

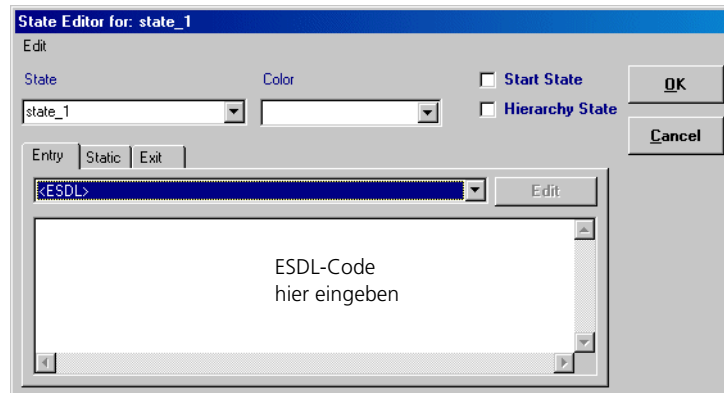
Sie können Variable und Parameter deklarieren, indem Sie im Zustandsautomaten-Editor auf eine Elementschaltfläche klicken und einen Namen in die Fläche „Elements“ eingeben. Diese Variablen können Sie dann im ESDL-Code verwenden, indem Sie ihren Namen eingeben. Ebenso können Sie auf Triggerargumente, importierte Komponenten, deren Methoden oder die in ihnen definierten Elemente Bezug nehmen, indem Sie den Namen eingeben. Auf diese Weise ist es möglich, Verhaltensweisen komplexer Zustandsautomaten mit nur wenigen Zeilen Code zu spezifizieren und ohne dass irgendwelche zusätzlichen Diagramme benötigt werden. Die Sprache ESDL ist im Kapitel „Spezifikation in ESDL“ des ASCET-Referenzhandbuchs beschrieben.

## Spezifizieren von Aktionen im Zustandseditor:

- Klicken Sie mit der rechten Maustaste auf den Zustand, dessen Aktionen Sie spezifizieren möchten.
- Wählen Sie aus dem Kontextmenü **Edit Action** → **Entry**, wenn Sie die Eintrittsaktion bearbeiten wollen.
- Wählen Sie aus dem Kontextmenü **Edit Action** → **Static**, wenn Sie die statische Aktion bearbeiten wollen.
- Wählen Sie aus dem Kontextmenü **Edit Action** → **Exit**, wenn Sie die Austrittsaktion bearbeiten wollen.

Der Zustandseditor wird im entsprechenden Register geöffnet.

- Wählen Sie aus dem Kombikästchen <ESDL>. Das Eingabefeld unter dem Kombikästchen wird aktiviert.



- Geben Sie den ESDL-Code in das Eingabefeld ein.  
Sie können die letzte Eingabe mit <CTRL> + <Z> rückgängig machen.
- Klicken Sie auf **OK**, um Ihre Eingaben zu übernehmen.

Im Menü **Edit** im Zustandseditor stehen Ihnen einige Funktionen zur Verfügung, die das Spezifizieren von Aktionen in ESDL erleichtern.

### **Ausschneiden, Kopieren und Einfügen von ESDL-Code:**

---

- Markieren Sie den Codeabschnitt, den Sie bearbeiten wollen.
- oder*
- Wählen Sie **Edit** → **Select All**, um den gesamten Code der aktuellen Aktion zu markieren.
  - Wählen Sie **Edit** → **Cut**, um den markierten Text auszuschneiden und in die Zwischenablage zu speichern.
  - Wählen Sie **Edit** → **Copy**, um den markierten Text in die Zwischenablage zu kopieren.
  - Wählen Sie **Edit** → **Paste**, um den Code aus der Zwischenablage in ein beliebiges Register einzufügen.

Suchen und Ersetzen von Code ist möglich, siehe Abschnitt „Suchen und Ersetzen von C-Code:“.

Sie können Code aus einer externen Quelldatei in die Aktion einbinden oder den im Zustandseditor spezifizierten Code in eine externe Datei schreiben, siehe Abschnitt „Einbinden einer externen Quelldatei:“.

Sie können den eingegebenen Code auch ausdrucken; siehe Abschnitt „Ausdrucken des C-Codes:“.

Der Übergangseditor, in dem Bedingungen und Übergangskaktionen spezifiziert werden, hat dieselben Funktionalitäten wie der Zustandseditor.

### **Spezifizieren von Bedingungen/Übergangskaktionen im Übergangseditor:**

---

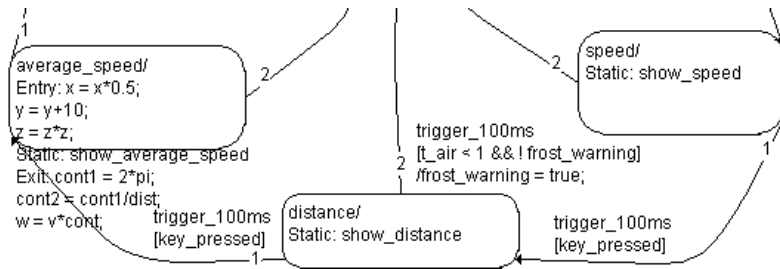
- Klicken Sie mit der rechten Maustaste auf den gewünschten Übergang.
- Wählen Sie aus dem Kontextmenü **Edit Transition**.

*Oder*

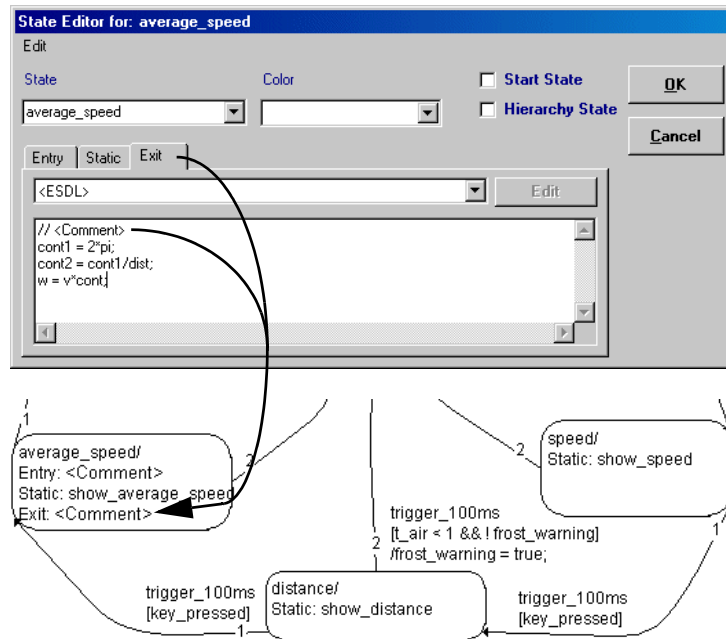
- Doppelklicken Sie auf den Übergang.  
Der Übergangseditor wird geöffnet.

- Wählen Sie im Register „Action“ oder „Condition“ aus dem Kombikästchen <ESDL> aus. Das Eingabefeld unter dem Kombikästchen wird aktiviert.
- Geben Sie den ESDL-Code im Eingabefeld ein.
- Klicken Sie auf **OK**.

Der ESDL-Code jeder Aktion und jeder Bedingung, die am Zustand oder Übergang spezifiziert wurde, wird im Zustandsdiagramm angezeigt. Bei aufwendigen Zustandsautomaten kann es daher schnell unübersichtlich werden.



Dies können Sie vermeiden, indem Sie beim Hinzufügen von Aktionen und Bedingungen in der ersten Zeile einen Kommentar eingeben. Einzeilige Kommentare werden durch zwei vorangestellte Schrägstriche // markiert, Kommentare beliebiger Länge werden in /\* <Kommentar> \*/ eingeschlossen. In diesem Fall werden nur die Kommentare im Diagramm dargestellt.



Sie können die Kommentarzeilen auch nachträglich automatisch einfügen.

### Automatisches Einfügen von Kommentarzeilen:

- Öffnen Sie den Zustandsautomaten im Zustandsautomaten-Editor.
- Wählen Sie **Diagram → Create State Code Comments**.

In jeder Bedingung und jeder Aktion, die im Zustandsdiagramm spezifiziert wurden, wird als erste Zeile ein Kommentar mit dem Text der ersten Code-Zeile eingefügt.

Wenn für einen Zustandsautomaten Code generiert wird, werden Teile von Aktionen und Bedingungen, die am Zustand oder Übergang spezifiziert wurden, unter bestimmten Voraussetzungen als eigene Methoden generiert (*Out-*

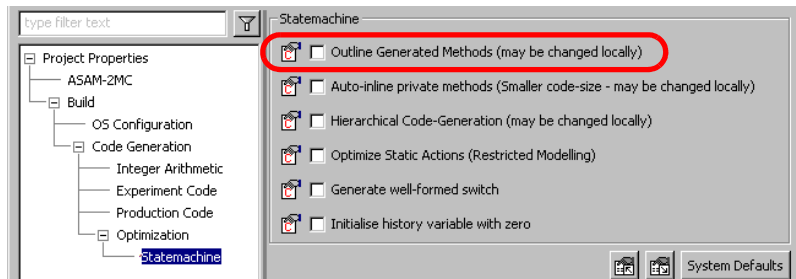
lining). Die Voraussetzungen für Outlining finden Sie im Kapitel „Den Zustandsautomaten optimieren“ im ASCET-Referenzhandbuch. Sie können das Outlining ein- oder ausschalten.

### Outlining von Aktionen/Bedingungen ein-/ausschalten:

Um Outlining ein-/auszuschalten, gehen Sie wie folgt vor.

- Öffnen Sie das Projekt, zu dem Ihr Zustandsautomat gehört.  
Das kann auch das Standardprojekt sein.
- Öffnen Sie im Projekteditor das Fenster „Project Properties“ (Seite 439).
- Deaktivieren Sie im Knoten „Statemachine“ (Seite 453) die Option **Outline Generated Methods (may be changed locally)**, um Outlining für alle Zustandsautomaten im Projekt auszuschalten.

#### 1. ausschalten



Die Einstellungen der einzelnen Zustandsautomaten sind in diesem Fall nicht von Bedeutung.

#### 2. einschalten

- Aktivieren Sie die Option **Outline Generated Methods (may be changed locally)**, um Outlining für alle Zustandsautomaten im Projekt einzuschalten.
- Schließen Sie das Fenster „Project Properties“.  
Die globale Einstellung in den Projekteigenschaften allein reicht nicht, damit Outlining stattfindet.



- Öffnen Sie mit **Component** → **Edit Implementation** den Implementierungseeditor jedes Zustandsautomaten im Projekt, für den Outlining eingeschaltet werden soll.
- Aktivieren Sie im Implementierungseeditor im Register „Settings“ die Option **Outline automatically generated methods for State Machines**.
- Schließen Sie den Implementierungseeditor.  
Die Codegenerierung erzeugt gegebenenfalls den Code für Aktionen/Bedingungen dieses Zustandsautomaten als separate Methoden.

### *Kommunikation mit anderen Komponenten*

---

Ein Zustandsautomat kann mit anderen ASCET-Komponenten kommunizieren. Dafür gibt es mehrere Möglichkeiten: Eingänge und Ausgänge, Triggerargumente und öffentliche Methoden (siehe auch „Zustandsautomaten als Klassen“ im ASCET-Referenzhandbuch).

### **Hinzufügen von Eingängen und Ausgängen zum Zustandsautomaten:**

---



- Klicken Sie auf die Schaltfläche **Input** oder **Output**.
- Im Blockdiagrammeditor für ein `ActionCondition`-Diagramm klicken Sie auf die Stelle im Zeichenbereich, wo Sie den Eingang oder Ausgang anordnen möchten.  
Der Eingang oder Ausgang wird erzeugt. Sie können das Element wie ein Argument oder eine Variable verwenden.

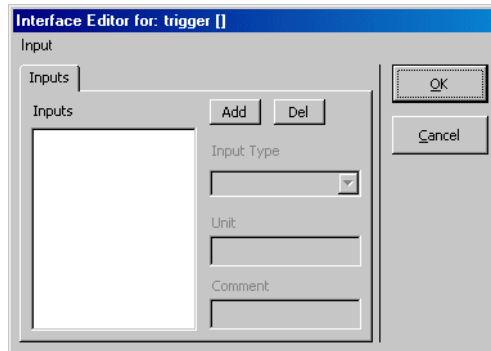
Wenn Sie die externe Kommunikation mit Hilfe von Triggerargumenten durchführen wollen, sind eine Reihe von Schritten notwendig.

Zunächst einmal brauchen Sie ein oder mehrere Triggerargumente. Die Regeln, für welchen Fall welchem Trigger ein Argument hinzugefügt werden muss, sind im Kapitel „Zustandsautomaten als Klassen“ (Seite 87) im ASCET-Referenzhandbuch angegeben.

## Triggerargument hinzufügen:

---

- Öffnen Sie den Schnittstelleneditor für den gewünschten Trigger (s. Seite 202).



Anders als bei den Schnittstellen normaler Methoden gibt es hier nur ein Register, „Inputs“, und ein Menü, **Input**.

- Fügen Sie im Register „Inputs“ die nötigen Argumente hinzu (s. Seite 202).

Das Triggerargument gehört zu einer öffentlichen Methode im selben Diagramm wie das Zustandsdiagramm. Sie können es daher genau wie Variablen oder Parameter verwenden, wenn Sie Bedingungen oder Aktionen im Übergang oder Zustand spezifizieren (siehe „Bedingungen und Aktionen im Zustandsdiagramm“ auf Seite 315).

Wenn Sie das Triggerargument in einer Aktion oder Bedingung verwenden wollen, die in einem `ActionCondition`-Diagramm spezifiziert ist, müssen Sie in der Aktion oder Bedingung ein Argument mit gleichem Namen und gleichem Typ erzeugen. Die Argumente in Triggern und Aktionen/Bedingungen werden entsprechend ihres Namens und Typs aufeinander abgebildet.

## Argumente zu Bedingungen/Aktionen hinzufügen:

---

- Wählen Sie in der Fläche „Diagrams“ die gewünschte Aktion oder Bedingung.
- Öffnen Sie den Schnittstelleneditor.  
Der Schnittstelleneditor für Aktionen/Bedingungen ist derselbe wie für normale Methoden.

- Fügen Sie im Register „Arguments“ die nötigen Argumente hinzu; achten Sie dabei auf die Namens- und Typgleichheit.  
Sie können mit dem so erzeugten Argument Verbindungen herstellen wie mit einem Parameter.
- Bei Bedarf können Sie im Register „Return“ einer Bedingung einen Kommentar zum Rückgabewert eingeben.  
Aktionen können auch Rückgabewerte enthalten; diese werden bei der Verwendung als Aktion ignoriert.

#### 4.2.4 Öffentliche Methoden

---

Sie haben die Möglichkeit, in einem separaten Diagramm öffentliche Methoden zu spezifizieren. Diese Methoden können sowohl von innerhalb als auch von außerhalb des Zustandsautomaten aufgerufen werden.

##### **Erzeugen eines öffentlichen Diagramms:**

---

- Wählen Sie **Diagram** → **Add Diagram** → **Public**  
*oder*
- klicken Sie mit der rechten Maustaste in die Fläche „Diagrams“ und wählen Sie **Add Diagram** → **Public** aus dem Kontextmenü.  
Ein öffentliches Diagramm wird erzeugt.
- Geben Sie dem Diagramm in der Fläche „Diagrams“ den von Ihnen gewünschten Namen.
- Doppelklicken Sie in der Fläche „Diagrams“ auf den Namen des Diagramms.  
Das Diagramm wird in einem eigenen Editorfenster geladen.

##### **Hinzufügen einer öffentlichen Methode:**

---

- Wählen Sie in der Fläche „Diagrams“ ein öffentliches Diagramm.
- Wählen Sie **Diagram** → **Add Method**  
*oder*

- klicken Sie mit der rechten Maustaste in die Fläche „Diagrams“ und wählen Sie aus dem Kontextmenü **Add Method**.

Es wird eine neue Methode erzeugt, und ihr Name wird in der Fläche „Diagrams“ angezeigt.

- Geben Sie einen Namen für die Methode ein und drücken Sie <ENTER>.

Die Methoden werden wie in Kapitel 4.1 „Der Blockdiagrammeditor“ beschrieben spezifiziert. Im folgenden werden einige Anwendungsbeispiele beschrieben, die Liste ist aber nicht vollständig.

#### **Verwenden einer öffentlichen Methode für die externe Kommunikation:**

---

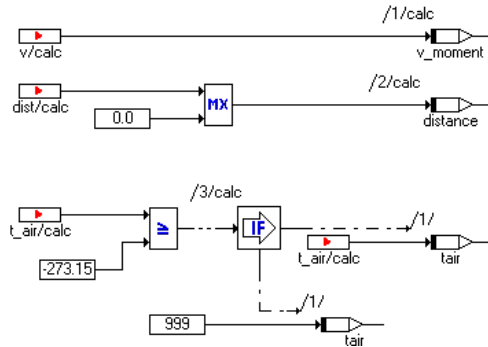
- Erzeugen Sie eine öffentliche Methode in einem öffentlichen Diagramm.
- Fügen Sie die nötige Anzahl Argumente hinzu, wie auf Seite 202 beschrieben.

Die Argumente sind nur innerhalb der Methode lesbar. Um im Zustandsautomaten verfügbar zu sein, müssen ihre Werte Variablen zugewiesen werden.

- Fügen Sie die entsprechende Anzahl von Variablen hinzu.

- Verbinden Sie jedes Argument mit einer Variablen.

Diese Variablen können im gesamten Zustandsautomaten verwendet werden. Die Abbildung enthält Beispiele, die Eingangswerte in der Methode aufzubereiten oder zu überprüfen.

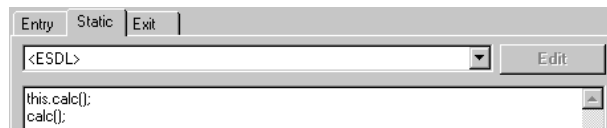


### Verwenden öffentlicher Methoden in Aktionen und Bedingungen:

Eine öffentliche Methode ohne Rückgabewert ist im Zustands- oder Übergangseditor im Auswahlkästchen auf den Aktionsregistern verfügbar. Eine öffentliche Methode mit Rückgabewert ist im Übergangseditor im Auswahlkästchen auf dem Register „Condition“ verfügbar.

- Weisen Sie die öffentliche Methode einem Zustand wie auf Seite 312 beschrieben zu.
- Weisen Sie die öffentliche Methode einer Bedingung wie auf Seite 310 beschrieben zu.

Wenn Sie eine Aktion oder Bedingung am Zustand oder Übergang spezifizieren, können Sie die öffentlichen Methoden des Zustandsautomaten genau wie die Methoden importierter Klassen aufrufen. Der Klassenname wird wahlweise durch `this` oder `self` ersetzt oder ganz weggelassen.



Beide Schreibweisen für den Aufruf der öffentlichen Methode `count` in der Abbildung haben dieselbe Wirkung.

## 4.2.5 Experimentieren mit Zustandsautomaten

---

Sie können mit einem Zustandsautomaten ebenso wie mit jeder anderen Komponente experimentieren. Um die Offline-Experimentierumgebung für einen Zustandsautomaten zu starten, führen Sie die gleichen Schritte aus, die im Abschnitt „Analysieren von Komponenten“ auf Seite 274 dargelegt sind.

### **Analysieren eines Zustandsautomaten:**

---

Sie können wahlweise das Zustandsdiagramm oder eines der zum Zustandsautomaten gehörenden Diagramme analysieren.

- Laden Sie das Diagramm, das Sie analysieren wollen, in den Zeichenbereich.
- Wählen Sie **Diagram** → **Analyze Diagram**, um das aktuelle Diagramm zu analysieren.  
Das Diagramm wird auf syntaktische Fehler geprüft. Fehler werden im Monitorfenster (siehe Kapitel 2.4 „Das ASCET-Monitorfenster“) gemeldet.

#### **Hinweis**

---

*ESDL-Code in Bedingungen oder Aktionen wird **nicht** analysiert.*

- Klicken Sie auf eine Fehlermeldung im Monitorfenster, Register „Build“, um den Fehler im Zustandsautomaten-Editor automatisch hervorheben zu lassen.

Das Einrichten der Experimentierumgebung erfolgt ebenfalls für alle Arten von Komponenten auf dieselbe Weise und ist in „Die Experimentierumgebung“ auf Seite 603 beschrieben. Der Zustandsautomaten-Editor bietet die zusätzliche Funktion der Animation des Zustandsautomaten.

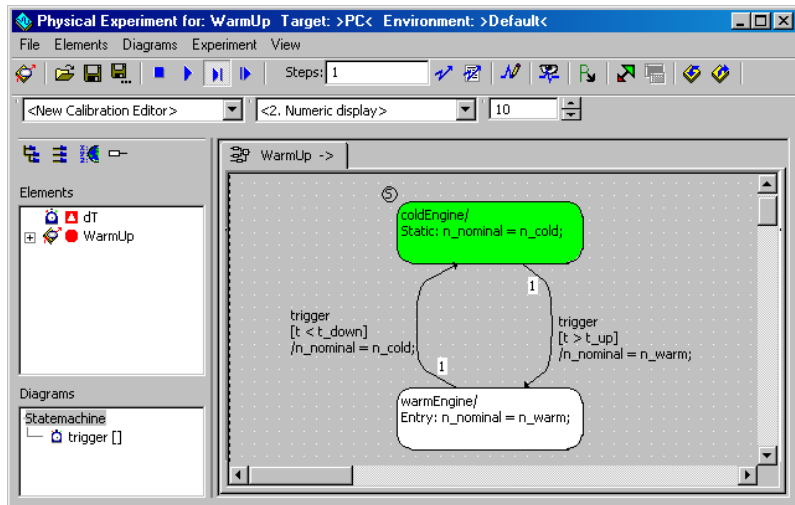
### **Verwenden der Animationsfunktion des Zustandsautomaten:**

---

- Starten Sie das Offline-Experiment für den Zustandsautomaten.
- Richten Sie das Experiment entsprechend den Erfordernissen ein.
- Schalten Sie zum Zustandsdiagramm um.
- Klicken Sie mit der rechten Maustaste auf einen Zustand.

- Wählen Sie aus dem Kontextmenü **Animate States**.
- Beginnen Sie mit der Simulation.

Die Simulation läuft wie gewöhnlich ab. Im Zustandsdiagramm wird der aktuelle Zustand durch die Animationsfarbe hervorgehoben.



### Ändern der Animationsfarbe:

Wenn Sie die Animationsfarbe ändern wollen, gehen Sie wie folgt vor.

- Wählen Sie im Komponentenmanager **Tools** → **Options**.
- Wählen Sie im Fenster „Options“ das Register „Fonts & Graphics“.
- Wählen Sie im Kombikästchen „Color of Animated States“ die gewünschte Farbe aus.
- Klicken Sie **OK**.

Der aktive Zustand wird in der gewählten Farbe dargestellt. Sie können die Animationsfarbe auch bei laufendem Experiment ändern.

Sie können bei laufendem Experiment Informationen über einzelne Zustände oder Übergänge anzeigen lassen.

## Informationen über Zustände oder Übergänge anzeigen:

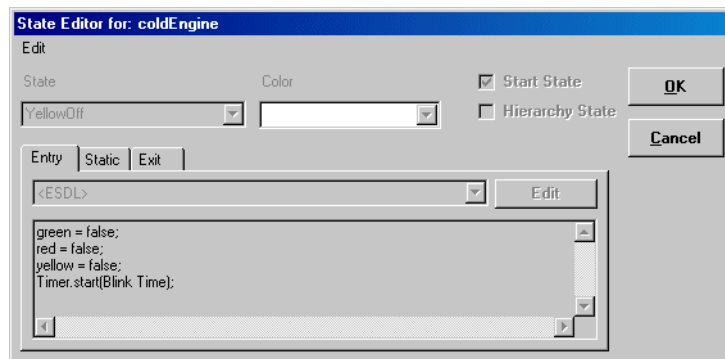
---

- Klicken Sie mit der rechten Maustaste auf einen Zustand und wählen Sie aus dem Kontextmenü **State Info**.

Oder

- Klicken Sie mit der rechten Maustaste auf einen Übergang und wählen Sie aus dem Kontextmenü **Transition Info**.

Der Zustandseditor bzw. Übergangseditor wird geöffnet; sämtliche Bearbeitungsmöglichkeiten sind deaktiviert.



Sie können die in den Registern zugewiesenen oder in ESDL spezifizierten Aktionen und Bedingungen anschauen.

- Schließen Sie das Fenster mit **OK** oder **Cancel**.

Sie können während des Experimentierens den Zustandsautomaten in seinen Anfangszustand zurücksetzen.

## Zustandsautomaten zurücksetzen:

---

- Klicken Sie mit der rechten Maustaste auf einen beliebigen Zustand.
- Wählen Sie aus dem Kontextmenü **Reset State Machine** aus.

Der Zustandsautomat wird in den Startzustand zurückgesetzt.



Dies hat *keinen* Einfluss auf die Werte irgendwelcher Variablen im des Zustandsautomaten. Wenn das Experiment nicht angehalten wurde, läuft es von dort aus normal weiter.

## 4.3 Der C-Code-Editor

---

Der C-Code-Editor wird zum Spezifizieren von Klassen, Modulen oder zeitkontinuierlichen Blöcken in der Programmiersprache C verwendet. Komponenten können in C-Code spezifiziert werden, indem der Code für eine Klasse eingegeben oder ein bestehendes Programm eingebunden und nach Bedarf verändert wird.

Dieser Abschnitt beschreibt, wie der C-Code-Editor zum Spezifizieren von Klassen und Modulen zu verwenden ist. Der C-Code-Editor folgt den gleichen grundlegenden Prinzipien wie der Blockdiagrammeditor. Nähere Informationen dazu siehe „Der Blockdiagrammeditor“ auf Seite 186.

### Erzeugen von Komponenten in C-Code:

---

- Wählen Sie im Komponentenmanager den Ordner aus, der die Komponente enthalten soll.
- Wählen Sie **Insert** → **Class** → **C-Code** oder **Module** → **C-Code**.

Oder



- benutzen Sie die Schaltflächen **Insert Class - <Type>** bzw. **Insert Module - <Type>**.
  - Wählen Sie mit einer der Pfeil-Schaltflächen den Typ **C Code**.
  - Klicken Sie auf **Insert Class - C Code** oder **Insert Module - C Code**.

Eine neue Komponente wird erzeugt.

- Geben Sie den Namen für die Komponente ein und drücken Sie <ENTER>.
- Wählen Sie in der Menüzeile **Component** → **Edit Item**

oder

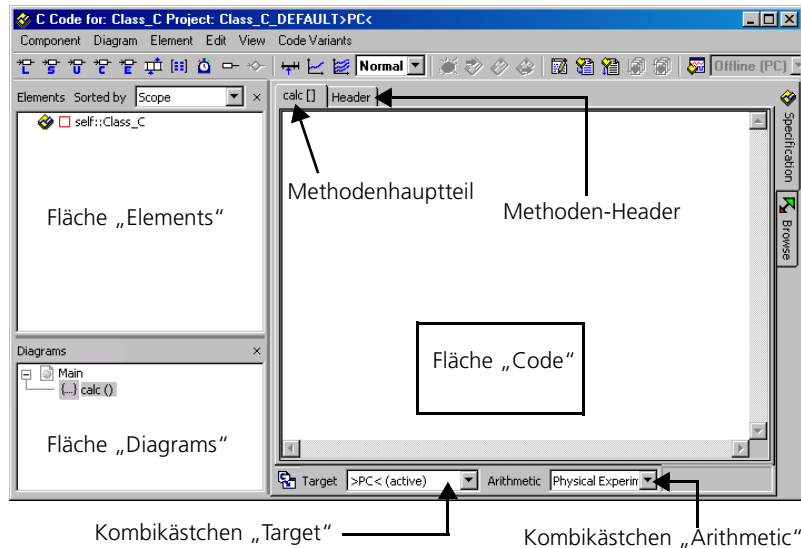
- drücken Sie <ENTER>

oder

- doppelklicken Sie in der Liste „1 Database“ auf den Namen der Komponente.

Der C-Code-Editor für die neue Komponente wird geöffnet.

Das folgende Bild zeigt den C-Code-Editor, wobei die wichtigsten Teile beschriftet sind.



Die Flächen „Elements“ und „Diagrams“ arbeiten wie im Blockdiagrammeditor. Die Diagramm-Funktion ist vorhanden, dient aber nur zum Strukturieren der Methoden der Komponente in öffentliche und private Methoden. Der Code für einen Methoden- oder Prozesshauptteil einer Komponente wird in der Code-Fläche gezeigt, wenn die Methode oder der Prozess in der Fläche „Diagrams“ angewählt ist. Die Titelleiste zeigt den Namen der Komponente, das aktuelle Projekt und das aktuelle Target.

Das Kombikästchen „Target“ zeigt das aktuelle Target. Eine C-Code-Komponente kann Code für mehrere Targets enthalten, aber der Code muss für jedes Target individuell spezifiziert werden. Das liegt daran, dass C-Code plattformabhängig ist; für eine Plattform entwickelter C-Code wird ohne Veränderungen nicht unbedingt auf einer anderen Plattform laufen.

Für jedes Target sind beim Experimentieren verschiedene Arithmetiken möglich, z.B. Gleit- oder Festkomma-Arithmetik; auch dafür ist C-Code separat zu spezifizieren. Man kann den Code für jede Target/Arithmetik-Kombination zu einer anderen kopieren, so dass der Code nur einmal eingegeben werden muss und dann angepasst werden kann.

In der Fläche „Code“ spezifizieren Sie den Hauptteil einer Methode oder eines Prozesses. Eine Methode korrespondiert mit einer Funktion in C, d.h. sie kann mehrere Argumente erhalten, aber nur einen Wert zurückgeben. Ein Prozess hat weder Argumente noch Rückgabewerte. Die Deklaration der Funktion wird von ASCET generiert; nur der Funktionsinhalt ist vom Benutzer zu definieren:

```
calc (a,b)
{
    <vom Nutzer definierter Inhalt in C-Code>
}
```

Die Argumente und Rückgabewerte der Funktion werden wie im Blockdiagrammeditor angegeben.

Falls Sie Systembibliotheken einbinden wollen, können Sie dies über eine gewöhnliche Include-Anweisung im Header der Klasse oder des Modules erreichen. Den Header kann man durch Anklicken des Registers „Header“ in der Fläche „Code“ sehen. Lokale Variablen einer Methode werden mit normalen C-Anweisungen deklariert. Der Geltungsbereich einer Variablen ist die Methode oder der Prozess, wo sie erklärt ist, außerhalb kann sie nicht referenziert werden.

Basiselemente werden wie im Blockdiagrammeditor durch Anklicken einer **Element**-Schaltfläche erzeugt. Elemente arbeiten wie Elemente in Komponenten, die durch Blockdiagramme beschrieben sind, d.h. sie können in allen Methoden oder Prozessen der Komponente verwendet werden und als global erklärt werden. Nachdem ein Element erzeugt ist, können Sie darauf Bezug nehmen, indem Sie seinen Namen in der Code-Fläche eingeben.

In ähnlicher Weise werden Argumente und Rückgabewerte für Klassen sowie Nachrichten für Module wie in einer Blockdiagrammkomponente definiert. Falls eine Komponente andere referenziert, gibt es einen syntaktischen Unterschied; somit sollten Sie frühzeitig entscheiden, ob eine Komponente andere referenziert.

### *Beschreibung der Menüfunktionen*

---

- **Component**
  - *Clean code generation directory*  
Löscht alle Dateien im Codegenerierungsverzeichnis.

- *Touch*  
Erzwingen einer Neugenerierung bei der nächsten Codegenerierung  
(*Flat* → die editierte Komponente,  
*Recursive* → auch die referenzierten Komponenten).
- *Generate Code* (<CTRL> + <F7>)  
Der Code für eine Komponente wird generiert.
- *Compile*  
Kompiliert den generierten Code.
- *Open Experiment*  
Ein Experiment wird gestartet.
- *Default Project*  
Erlaubt die Bearbeitung des Standardprojektes, das beim Experimentieren mit Komponenten verwendet wird  
(*Edit* → öffnet einen Editor für das Standardprojekt,  
*Resolve Globals* → erzeugt für jedes importierte Element der Komponente, für das kein exportiertes Element vorhanden ist, automatisch ein globales Element,  
*Delete Unused Globals* → löscht nicht genutzte globale Elemente).
- *Edit Layout*  
Der Layout-Editor wird geöffnet.
- *Edit Data*  
Der Dateneditor für eine Komponente wird geöffnet. Das Durchsuchen der Daten einer Komponente ist möglich.
- *Edit Implementation*  
Der Implementierungseditor wird geöffnet. Das Durchsuchen der Implementierungen einer Komponente ist möglich.
- *Edit Notes*  
Öffnet den Notizeneditor – Hier können Notizen zur Komponente gemacht werden.
- *Check Dependency*  
Es wird überprüft ob die Zuweisung der formalen Parameter zu den Modellparametern korrekt ist.
- *Export Data*  
Exportieren des Datensatzes einer Komponente.

- *Show Path*  
Zeigt den Pfad einer aufgenommenen Komponente.
- *Copy Path to Clipboard*  
Kopiert den Pfad einer aufgenommenen Komponente in die Zwischenablage  
(*Model Path* → Pfad im Modell,  
*Model asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente im Modell als Hyperlink ermöglicht,  
*Database Path* → Pfad in der Datenbank,  
*Dabase asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente in der Datenbank als Hyperlink ermöglicht).
- *File Out Generated Code*  
Speichern des generierten Code im Dateisystem  
(*Flat* → die editierte Komponente,  
*Recursive* → auch die referenzierten Komponenten).
- *View Generated Code*  
Generiert den Code für die Komponente und zeigt diesen in einem Texteditor an.  
Der Texteditor kann im ASCET-Optionsfenster, Knoten „ASCII Editor“ (s. Seite 61), gewählt werden.
- *File Out Generic Code For External Make*  
Speichern des generierten Codes in ein beliebiges Verzeichnis zur späteren Weiterverarbeitung durch separate Werkzeuge, z.B. in einem externen Make/Build - Prozess.
- *Exit*  
Beendet den C-Code-Editor.

- **Diagram**

### Hinweis

Dieses Menü steht in der Fläche „Diagrams“ auch als Kontextmenü zur Verfügung.

- *Analyze Diagram*  
Analysieren des aktuellen Diagramms (im C-Code-Editor deaktiviert).

- *Add Diagram*  
(*Public* → enthält nur öffentliche Methoden,  
*Private* → enthält nur private Methoden)
- *Rename Diagram*  
Umbenennen eines Diagramms.
- *Delete Diagram*  
Löschen eines Diagramms aus der Komponente.
- *Add Method*  
Erzeugen einer Methode. Verfügbar für Klassen und Module.
- *Add Process*  
Erzeugen eines Prozesses.
- *Add Trigger / Action / Condition*  
Deaktiviert; nur im Zustandsautomaten-Editor verfügbar.
- *Edit*  
Bearbeiten der Schnittstelle einer Methode / eines Prozesses.
- *Rename*  
Umbenennen einer Methode / eines Prozesses.
- *Delete*  
Löschen einer Methode / eines Prozesses.
- *Move Up*  
Bewegen eines Diagramms oder einer Methode / eines Prozesses  
(nach oben).
- *Move Down*  
Bewegen eines Diagramms oder einer Methode / eines Prozesses  
(nach unten).
- *Move*  
Bewegen von Methoden/Prozessen zwischen Diagrammen.
- *Edit Implementation*  
Spezifizieren der Implementierung einer Methode / eines Prozesses.

- **Element**

### Hinweis

Die Funktionen **Add Item**, **Rename**, **Delete** und **Edit** stehen nur zur Verfügung, wenn ASCET-MD installiert ist.

Dieses Menü steht in der Fläche „Elements“ auch als Kontextmenü zur Verfügung.

- *View* → *Collapse all* klappt die Baumstruktur in der Fläche „Elements“ zu, sodass nur noch die Komponente selbst zu sehen ist.
- *View* → *Expand all* klappt die Baumstruktur vollständig auf, sodass der gesamte Inhalt der Komponente angezeigt wird.
- *Add Item*  
Aufnehmen einer Komponente als komplexes Element.
- *Rename* (<F2>)  
Umbenennen eines Diagrammelements.
- *Delete* (<DEL>)  
Löschen eines Diagrammelements.
- *Show Occurences*  
Zeigt alle Vorkommen eines Elements an (nur für Blockdiagramme).
- *Edit*  
Bearbeiten der Konfiguration eines Elements.
- *Edit Data*  
Bearbeiten der Daten eines Elements.
- *Edit Implementation*  
Bearbeiten der Implementierung eines Elements.
- *Set Cache Locking*  
Diese Menüfunktion ist nur für die Arbeit mit ASCET-RP von Bedeutung. Sie ist im ASCET-RP-Handbuch beschrieben.
- *Edit Component*  
Öffnen des Spezifikationseditors für eine aufgenommen Komponente.
- *Replace Component*  
Ersetzen einer Komponente durch eine andere Komponente. Der Name der alten Komponente bleibt erhalten.

- *Show Path*  
Zeigt den Pfad einer aufgenommenen Komponente.
- *Copy Path to Clipboard*  
Kopiert den Pfad einer aufgenommenen Komponente in die Zwischenablage  
(*Model Path* → Pfad im Modell,  
*Model asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente im Modell als Hyperlink ermöglicht,  
*Database Path* → Pfad in der Datenbank,  
*Dabase asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente in der Datenbank als Hyperlink ermöglicht).
- *Notes*  
Öffnet den Notizeneditor für eine aufgenommene Komponente.
- *Edit Distribution*  
Öffnet einen Editor für die Distribution (nur bei Gruppentabellen).
- *Edit Max Size*  
Öffnet einen Editor, in dem die maximale Größe eines Arrays einer beliebigen Tabelle oder Matrix vorgegeben werden kann.
- *Copy Elements* (<CTRL> + <C>)  
Kopieren eines Elements aus der Liste „Elements“.
- *Paste Elements* (<CTRL> + <V>)  
Einfügen eines kopierten Elements in die Liste „Elements“.
- *File Out Data*  
Schreiben der Daten eines Array oder einer Tabelle in eine Datei.
- *File In Data*  
Lesen der Daten eines Array oder einer Tabelle aus einer Datei.
- *Export Data*  
Exportieren des Datensatzes einer aufgenommenen Komponente.
- **Edit**
  - *Element / Data / Implementation*  
Entspricht **Edit** / **Edit Data** / **Edit Implementation** aus dem Kontextmenü der Elementliste.



- *Cut* (<CTRL> + <X>)
 

Schneidet den markierten Text aus.
- *Copy* (<CTRL> + <C>)
 

Kopiert den markierten Text.
- *Paste* (<CTRL> + <V>)
 

Fügt Code aus der Zwischenablage ein.
- *Find / Replace* (<CTRL> + <F>)
 

Öffnet den Editor zum Suchen und Ersetzen von C-Code.
- *Select All* (<CTRL> + <A>)
 

Markiert den gesamten Code der aktuellen Methode oder des aktuellen Prozess.
- *Load from File*

Importieren / Einfügen aus einer Datei.
- *Store to File*

Code wird in eine externe Datei geschrieben.
- *Print*

Druckt den aktuellen Code aus.
- *Printer Setup*

Öffnet das Auswahlfenster „Printer Selection“, in dem der Drucker ausgewählt und eingerichtet wird.
- *Save* (<CTRL> + <S>)
 

Speichern einer Methode oder eines Prozesses.
- **View**
  - *Show/Hide*

Zeigt/verbirgt verschiedene Teile des Editors oder der Komponente.  
(*Element List* → Liste „Elements“,  
*Diagram List* → Liste „Diagrams“).
- **Code Variants**
  - *Copy C-Code To*

Der Code wird (ohne Veränderung) in die ausgewählte Target/Arithmetik-Kombination kopiert.
  - *Copy C-Code From*

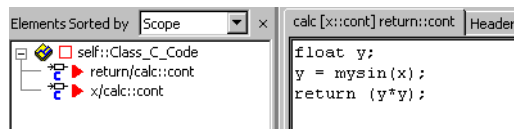
Kopieren einer Target/Arithmetik Kombination.

### 4.3.1 Spezifizieren von Komponenten in C-Code

Eine Methode mit dem Namen `calc` wird standardmäßig angelegt. Das Erzeugen von Methoden und Prozessen und die Spezifikation der Schnittstellen geht wie im Blockdiagrammeditor und ist in „Definieren der Schnittstelle einer Komponente“ auf Seite 201 beschrieben. Das Erzeugen von Basiselementen (s. Seite 218) und Enumerationen (s. Seite 219), die Aufnahme von Komponenten als komplexe Elemente (s. Seite 229) und das Bearbeiten von Notizen (s. Seite 231) funktionieren ebenfalls wie im Blockdiagrammeditor.

#### **Spezifizieren einer Methode oder eines Prozesses:**

- Wählen Sie in der Fläche „Diagrams“ die Methode oder den Prozess, die/der spezifiziert werden soll.  
Der Name erscheint in der Lasche des linken Registers.
- Wählen Sie aus dem Kombikästchen „Target“ ein Target aus.  
Alle in Ihrem ASCET-System installierten Targets sind in der Liste vorhanden.
- Wählen Sie aus dem Kombikästchen „Arithmetic“ die gewünschte Arithmetik für den Code.
- Klicken Sie auf das Register „Header“.
- Geben Sie Header-Informationen für die Klasse oder Methode ein, z.B. lokale Variablen, Include-Anweisungen oder Makros.
- Klicken Sie auf das Register „<Methodenname>“.
- Geben Sie den Code für die Methode ein.



In der Fläche „Elements“ definierte Basiselemente können durch Eingeben ihrer Namen referenziert werden. Dies ist jedoch nur der

Fall, wenn keine referenzierten Komponenten innerhalb der Komponente verwendet werden.

- Wiederholen Sie den Vorgang für alle Methoden und Prozesse.

Da C-Code immer plattformabhängig ist, müssen Sie den Code einer C-Code-Komponente für jede Plattform und jede Arithmetik individuell spezifizieren. Ein für eine Target/Arithmetik-Kombination entwickelter Code kann jedoch leicht kopiert und für eine andere Kombination benutzt werden.

### **C-Code für einzelne Module oder Klassen kopieren:**

---

Um existierenden C-Code einer einzelnen Klasse oder eines Moduls auf ein anderes Target zu kopieren, verwenden Sie eine der beiden Möglichkeiten, die im folgenden beschrieben sind.

1. Verwenden Sie **Code Variants** → **Copy C-Code To**.

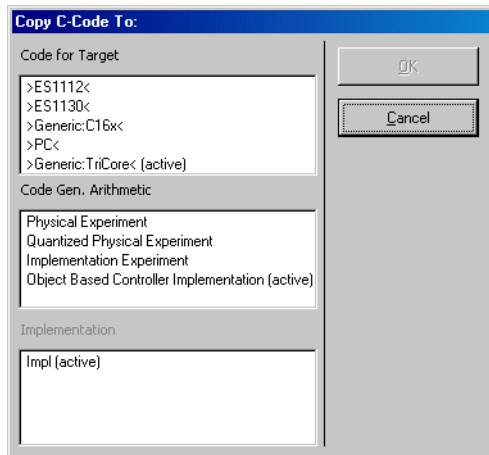
Target >PCx (active) ▼

Arithmetic Object Based ▼

Implementation Impl (act) ▼

- Öffnen Sie das Modul/die Klasse im C-Code-Editor.
- Wählen Sie im Kombikästchen „Target“ das Target aus, für das der C-Code erstellt wurde.
- Wählen Sie im Kombikästchen „Arithmetik“ den Experimenttyp aus, für den der C-Code erstellt wurde.
- Wählen Sie im Kombikästchen „Implementation“ die Implementierung aus, für die der C-Code erstellt wurde.
- Wählen Sie jetzt **Code Variants** → **Copy C-Code To**.

- Das Dialogfenster „Copy C-Code to:“ wird geöffnet.



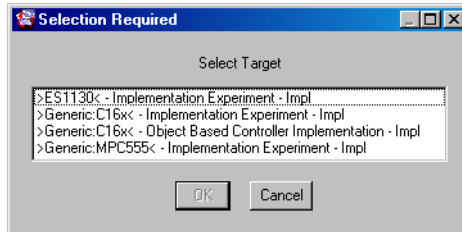
- Wählen Sie in der Fläche „Code for Target“ ein Target.
- Wählen Sie im Feld „Code Gen. Arithmetic“ den geeigneten Experimenttyp.  
Alle Experimente können eine Implementierung auswerten. Alle definierten Implementierungen werden in der Fläche „Implementation“ gezeigt.
- Wählen Sie eine Implementierung aus.  
Sobald Sie die Auswahl abgeschlossen haben, wird die Schaltfläche **OK** aktiviert.
- Klicken Sie auf **OK**.  
Der C-Code wird auf das ausgewählte Target kopiert.

2. Verwenden Sie **Code Variants** → **Copy C-Code From**.

- Benutzen Sie im C-Code-Editor die Kombikästchen „Target“, „Arithmetic“ und „Implementation“, um das von Ihnen verwendete Target mit passendem Experimenttyp und Implementierung einzurichten.

- Wählen Sie jetzt **Code Variants** → **Copy C-Code From**.

Das Fenster „Selection Required“ wird geöffnet.



- Wählen Sie das Target, den Experimenttyp und die Implementierung aus, von dem Sie den Code kopieren wollen, und klicken Sie **OK**.

Der C-Code wird auf das gewählte Target kopiert.

### **Bearbeiten des Codes einer Methode oder eines Prozesses:**

- Markieren Sie den zu editierenden Codeabschnitt

*oder*

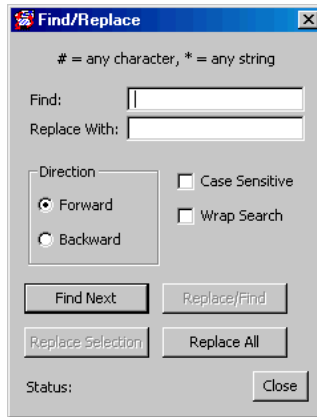
- wählen Sie **Edit** → **Select All**, um den gesamten Code der aktuellen Methode oder des aktuellen Prozesses zu markieren.
- Wählen Sie **Edit** → **Cut**, um den markierten Text auszuschneiden und in die Zwischenablage zu speichern.
- Wählen Sie **Edit** → **Copy**, um den markierten Text in die Zwischenablage zu kopieren.
- Wählen Sie **Edit** → **Paste**, um den Code aus der Zwischenablage einzufügen.

Der Text, den Sie ausschneiden oder kopieren, wird in der Zwischenablage von Windows gespeichert und kann mit anderen Anwendungen (z.B. C-Entwicklungsumgebungen) ausgetauscht werden. Sie können auch Code zwischen Methoden, Prozessen oder Komponenten austauschen.

## Suchen und Ersetzen von C-Code:

---

- Wählen Sie **Edit** → **Find/Replace**.  
Das Dialogfenster „Find/Replace“ wird geöffnet.



- Geben Sie den zu suchenden Text in das Feld „Find“ ein.

Es können Platzhalter benutzt werden: # für ein beliebiges Zeichen, \* für eine Zeichenkette.

- Geben Sie den Text, der eingesetzt werden soll, in das Feld „Replace With“ ein.
- Klicken Sie auf die Schaltfläche **Find Next**, um das nächste Auftreten des gesuchten Textes zu finden.

Der gefundene Suchtext wird hervorgehoben. Wird der Suchtext nicht gefunden, erscheint die Nachricht `String Not Found` in der Statuszeile am unteren Rand des Dialogfensters.

- Klicken Sie auf die Schaltfläche **Replace Selection**, um den hervorgehobenen Text durch jenen aus dem Feld „Replace With“ zu ersetzen.

Diese Schaltfläche ist nur vorhanden, wenn der Text vorher gefunden wurde.

- Klicken Sie auf die Schaltfläche **Replace/Find**, um den hervorgehobenen Text zu ersetzen und das nächste Auftreten zu suchen.
- Klicken Sie auf die Schaltfläche **Replace All**, um alle Vorkommen zu ersetzen.
- Klicken Sie auf die Schaltfläche **Close**, um das Dialogfenster „Find/Replace“ zu schließen.

### **Anpassen der Suchkriterien:**

---

- Klicken Sie auf die Option **Forward**, um von der Cursorposition an vorwärts zu suchen.

*oder*

- Klicken Sie auf die Option **Backward**, um ab dem Cursor rückwärts zu suchen.
- Durch Aktivsetzen des Kästchens **Case Sensitive** wird Groß- und Kleinschreibung unterschieden.
- Durch Aktivsetzen des Kästchens **Wrap Search** wird am Ende bzw. Anfang des Textes die Suche am Anfang bzw. Ende fortgesetzt.

Ist diese Option gewählt, beginnt die Suche an der Cursorposition in angegebener Richtung und setzt sich am Anfang oder Ende des Textes bis zur Einfügeposition fort.

### **Speichern einer Methode oder eines Prozesses:**

---

- Wählen Sie **Edit** → **Save**

*oder*

- drücken Sie <CTRL> + <S>, um die aktuelle Methode oder den aktuellen Prozess zu speichern.

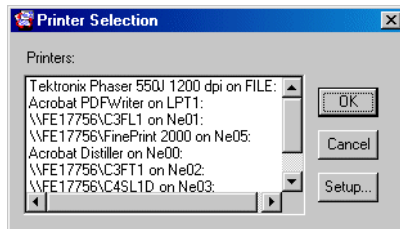
Dieser Befehl ist nur möglich, wenn die Methode oder der Prozess seit dem letzten Speichern verändert worden ist. Beim Umschalten zwischen Methoden ohne vorheriges Speichern werden Sie gefragt, ob die aktuelle Methode gespeichert werden soll.

### Ausdrucken des C-Codes:

---

- Wählen Sie **Edit** → **Print**, um den Code auf Ihrem Standarddrucker auszudrucken.
- Wählen Sie **Edit** → **Printer Setup**, um einen Drucker auszuwählen.

Das Fenster „Printer Selection“ wird geöffnet.



- Wählen Sie im Feld „Printers“ einen Drucker.
- Klicken Sie auf **Setup**, um die Druckereinrichtung zu ändern.
- Passen Sie im Optionsfenster des Druckers die Einstellungen an.



- Klicken Sie **OK**, um die Auswahl zu übernehmen.

Das Fenster „Printer Font Selection“ öffnet sich.



- Stellen Sie die Schrift ein, die Sie verwenden wollen.
- Klicken Sie **OK**, um die Auswahl zu übernehmen.

### 4.3.2 Externer Editor

Der vorhandene Editor zum Erzeugen eines C-Codes verfügt nur über einen geringen Leistungsumfang und einfache Bedienungsmöglichkeiten. Eine andere Variante zur Codeerzeugung besteht darin, einen benutzerdefinierten externen Editor (z.B. Notepad, Codewright, etc.) zu wählen. Da der Link zum externen Editor über ein File System realisiert ist, muss eine Verknüpfung der Dateien mit den Endungen „\*.c“ und „\*.h“ zum Editor in der Windows-Konfiguration erfolgen. Ohne diese Verknüpfung kann der externe Editor aus der ASCET-Umgebung nicht geöffnet werden, allerdings weist eine Fehlermeldung auf die fehlende Verknüpfung hin.

Nach Betätigung der Schaltfläche für den externen Editor wird die Ansicht vom C-Code-Editor geändert, unterteilt in ein oberes Feld mit der Auflistung aller vorhandenen Methoden bzw. Prozesse einer Komponente und in ein unteres Feld zur Darstellung des C-Codes.

Der Code einer Methode bzw. eines Prozesses wird beim Öffnen des externen Editors in eine oder mehrere Dateien geschrieben, die in einem temporären Verzeichnis abgelegt werden. Nach Beendigung der Arbeit müssen die Dateien erst im externen Editor abgespeichert werden, bevor sie in die ASCET-Umgebung zurückübertragen werden können. Ein Beenden des externen Editors aus der ASCET-Umgebung schreibt die abgespeicherten Codeänderungen aus dem externen Editor in die ASCET-Umgebung, dabei wird das temporäre Verzeichnis gelöscht. Codeänderungen, die noch in der offenen externen Editor-

umgebung durchgeführt werden, lassen sich nicht mehr in die ASCET Umgebung übertragen. Soll der Code erneut bearbeitet werden, muss ein neuer externer Editor aus der ASCET-Umgebung aufgerufen werden.

### Hinweis

*Um eine im externen Editor durchgeführte Codeänderungen in die ASCET-Umgebung zu transportieren, muss der Code im externen Editor zuvor abgespeichert werden.*

### Öffnen eines externen Editors:

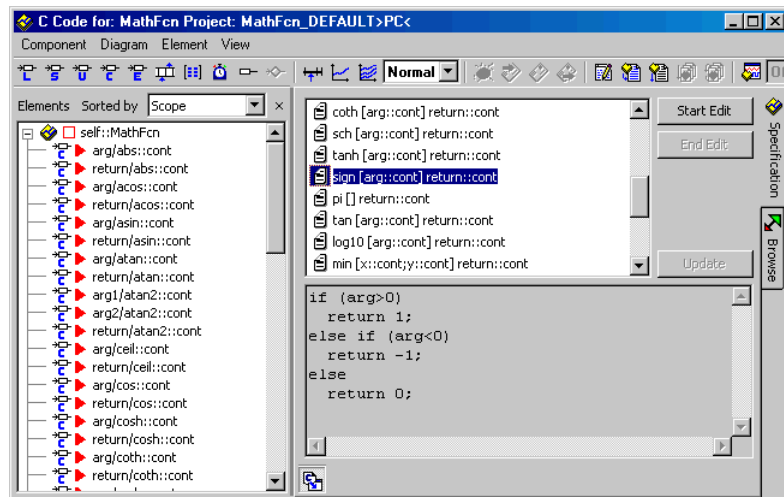


- Klicken Sie auf die Schaltfläche **Activate External Editor** am unteren Rand des Editorfensters.

Die Ansicht des C-Code-Editors ändert sich. Das Feld „Diagrams“ wird ausgeblendet, das Feld „Code“ wird durch ein Auswahlfeld und ein Textfeld ersetzt. Im Auswahlfeld werden alle definierten Methoden/Prozesse angezeigt.

- Wählen Sie aus dem Auswahlfeld die Methode/den Prozess aus, deren Code geändert werden soll.

Der gültige Code der ausgesuchten Methode bzw. Prozess erscheint zur Kontrolle im Textfeld.



- Klicken Sie auf die Schaltfläche **Start Edit**, um den externen Editor mit dem Code der markierten Methode bzw. Prozess zu öffnen.

 `sign [arg::cont] return::cont`

Nach dem Öffnen des externen Editors wird im oberen Dialogfeld die Methode bzw. Prozess durch einen roten Punkt im Symbol markiert.

- Bearbeiten Sie den Code im externen Editor.

### Schließen eines externen Editors:

---

- Speichern Sie im externen Editor den neu erstellten oder geänderten Code ab.
- Klicken Sie im C-Code-Editor auf die Schaltfläche **Update**.

Der aktuelle Stand der Codeänderung im externen Editor wird in die ASCET-Umgebung zu übertragen. Die Codeänderungen werden im Textfeld angezeigt.

Die Verbindung zum externen Editor wird dabei nicht getrennt. Dadurch ist es möglich, weiter im externen Editor zu arbeiten, Änderungen vorzunehmen und nach Abspeichern der Änderungen diese erneut in der ASCET-Umgebung zu aktualisieren.

- Klicken Sie auf die Schaltfläche **End Edit**, um die Arbeit mit dem externen Editor zu beenden.

Die Verbindung zum externen Editor wird getrennt, der rote Punkt im Symbol der Methode bzw. Prozess gelöscht. Anschließend Änderungen im geöffneten externen Editor werden *nicht mehr* berücksichtigt.

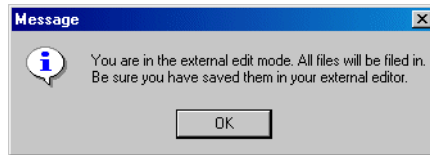
- Schließen Sie den externen Editor.

## Externen Editormodus beenden:

---

- Klicken Sie erneut auf die Schaltfläche **Activate External Editor**.

Wenn Sie vorher nicht auf **End Edit** geklickt haben, wird eine Warnung ausgegeben. Sie werden darauf hingewiesen, dass der Inhalt des externen Editors in die ASCET-Umgebung übernommen wird.



- Sichern Sie bei Bedarf die Datei im externen Editor, damit aktuelle Änderungen übernommen werden.
- Bestätigen Sie die Warnung.

Der externe Editormodus wird beendet und die ursprüngliche Darstellung des C-Code-Editors wieder hergestellt.

### 4.3.3 Verwendung des Editors für externe Quellen

---

Der Editor für externe Quellen ermöglicht die Integration bereits bestehender C-Programme als Teil einer Spezifikation einer C-Code-Komponente. Sie können C-Quellen oder Objekt-Code einbinden. Alle Funktionen in einer externen Quelldatei (für Quell- und Objektmodule) müssen in einer Header-Datei erklärt werden, die ebenfalls Teil der Komponente ist.

Externe Module müssen auch separat für jedes Target und jede Arithmetik spezifiziert werden. Auswählen von Targets und Arithmetik läuft, wie weiter oben in diesem Kapitel beschrieben, ab.

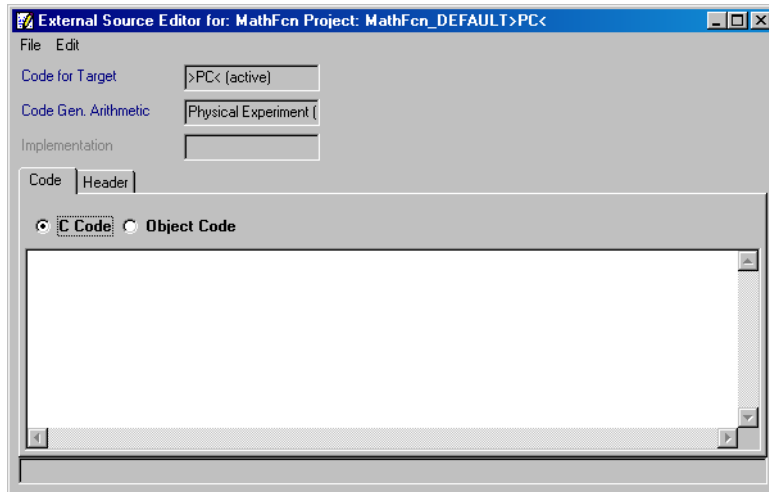
## Öffnen des Editors für externe Quellen:

---



- Klicken Sie auf die Schaltfläche **External Source Editor**.

Der Editor für externe Quellen erscheint.



## Einbinden einer externen Quelldatei:

---

- Klicken Sie auf das Register „Code“.  
Sie können nur eine externe Quelldatei in eine Komponente einbinden; dies kann entweder ein C-Code-Modul oder ein Objektcode-Modul sein.
- Klicken Sie auf die Option **C Code** oder **Object Code**.
- Geben Sie die Funktionen des externen Moduls ein (nur Option **C Code**)

oder

- wählen Sie **Edit** → **Load from File**.  
Das Dateiauswahlfenster erscheint.
- Wählen Sie das externe C- oder Objekt-Modul aus.

- Klicken Sie auf **Öffnen**.  
Der Inhalt der ausgewählten Datei wird in die Textfläche geschrieben; die ursprüngliche Datei wird dabei nicht beeinflusst.
- Wählen Sie **Edit** → **Store to File**.  
Das Dateiauswahlfenster erscheint. Der Fensterinhalt wird in die ausgewählte Datei geschrieben.
- Wählen Sie **Edit** → **Save** zum Speichern des externen Moduls.

Im Editor für externen Quellen sind die gleichen Bearbeitungsbefehle verfügbar wie im C-Code-Editor.

Wenn eine externe Klasse ein externes Quellmodul enthält, muss auch eine Header-Datei existieren, in der die Funktionen des Quellmoduls erklärt sind.

#### **Hinzufügen einer Header-Datei zum Quellmodul:**

---

- Klicken Sie auf das Register „Header“.
- Geben Sie den Text der Header-Datei ein oder laden Sie ihn, wie oben beschrieben.
- Aktivieren Sie das Kästchen **Use header global**, um die gleiche Header-Datei für alle Ihre C-Codes zu verwenden.

So markierte Header-Dateien werden in jede generierte \* .c-Datei eingebunden.

#### **Prüfen des externen Moduls:**

---

- Wählen Sie **File** → **Compile C-Code**.  
Dieser Befehl ruft den aktuell ausgewählten Compiler auf und übersetzt die .c- und .h-Dateien des externen Moduls. Dies endet nicht in ausführbarem Code, sondern dient nur zum Überprüfen der syntaktischen Korrektheit. Wenn die Quellen nicht korrekt sind, zeigt der Compiler entsprechende Fehlermeldungen an.

Der ESDL-Editor wird zum Spezifizieren von Komponenten in ESDL-Code verwendet. ESDL-Komponenten können Klassen, Module oder zeitkontinuierliche Blöcke sein. Eine Klasse in ESDL-Code arbeitet genauso wie eine als Blockdiagramm spezifizierte Klasse. Wenn sie von einer anderen Komponente verwendet wird, sieht sie genauso aus und verhält sich so wie eine Klasse, die als Blockdiagramm vorgegeben wurde.

Dieses Kapitel erläutert, wie Klassen und Module in ESDL-Code erstellt werden. Bevor Sie dieses Kapitel lesen, sollten Sie mit der Erstellung von Klassen als Blockdiagramme, wie in Kapitel „Der Blockdiagrammeditor“ auf Seite 186 erläutert, vertraut sein. Die ESDL-Sprache selbst ist im Kapitel „Spezifikation in ESDL“ des ASCET Referenzhandbuchs beschrieben.

#### Erstellen einer Klasse/eines Moduls in ESDL-Code:

- Wählen Sie im Komponentenmanager den Ordner aus, in den Sie die Komponente platzieren wollen.
- Wählen Sie **Insert** → **Class** → **ESDL**.

*Oder*



- benutzen Sie die Schaltflächen **Insert Class – <Type>** bzw. **Insert Module – <Type>**.
  - Wählen Sie mit einer der Pfeil-Schaltflächen den Typ **ESDL**.
  - Klicken Sie auf **Insert Class – ESDL** oder **Insert Module – ESDL**.

Eine neue Komponente wird erzeugt.

- Geben Sie einen Namen für die Komponente ein und drücken Sie <ENTER>.
- Wählen Sie in der Menüzeile **Component** → **Edit Item**

*oder*

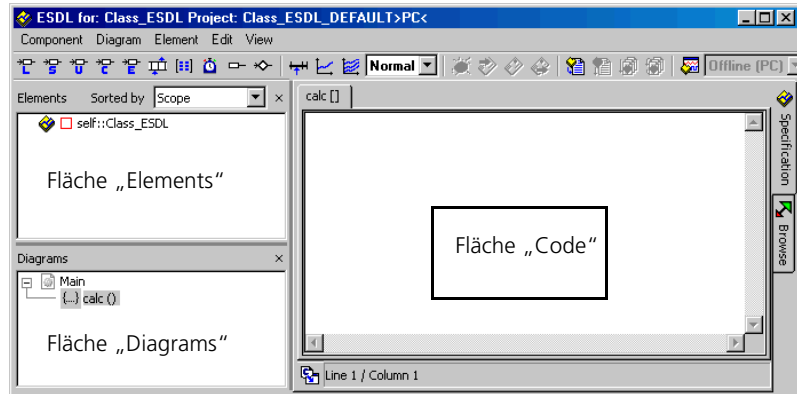
- drücken Sie <ENTER>

*oder*

- doppelklicken Sie in der Liste „Elements“ auf den Namen der Komponente.

Der ESDL-Editor wird geöffnet.

Wie im Blockdiagrammeditor haben in ESDL-Code spezifizierte Komponenten Methoden und Prozesse, die ihrerseits Elemente enthalten können. Es sind die gleichen Elementarten verfügbar; Schnittstelle und Methoden werden auf gleiche Weise definiert. Die Diagramm-Funktion ist ebenfalls vorhanden, dient aber nur zum Strukturieren der Methoden der Komponenten in öffentliche und private Methoden. Der ESDL-Code der Komponente wird für eine Methode oder einen Prozess in der Fläche „Code“ des ESDL-Editors angezeigt.



Die in der Menüleiste vorhandenen Funktionalitäten entsprechen den Funktionalitäten des C-Code-Editors (siehe Seite 331), mit zwei Ausnahmen: Die Menüfunktion **Diagram** → **Analyze Diagram** ist im ESDL-Editor verfügbar, ebenso **View** → **Show/Hide** → **Impl. Casts in Element List**.

#### 4.4.1 Spezifizieren von Klassen in ESDL

##### **Erzeugen von ESDL-Code:**

- Fügen Sie die nötigen Methoden hinzu, wie in „Definieren der Schnittstelle einer Komponente“ auf Seite 201 beschrieben.
- Wählen Sie in der Fläche „Diagrams“ eine Methode.
- Erstellen Sie die gewünschten Elemente durch Klicken auf die jeweiligen Schaltflächen und Eingeben der Elementnamen.



- Geben Sie den Code für die Methode in der Fläche „Code“ ein.

Wie im C-Code-Editor können Sie auch einen externen Editor verwenden (siehe Kapitel 4.3.2).

- Wählen Sie **Edit** → **Save**

oder

- drücken Sie <CTRL> + <S>, um den Code der aktuellen Methode zu speichern.

Sie werden bei jedem Wechsel zu einer anderen Methode gefragt, ob Sie speichern wollen.

In einer textuellen Spezifikation gibt es keine Sequenzaufrufe; die Reihenfolge, in der Anweisungen abgearbeitet werden, ist durch die Reihenfolge im Quellcode bestimmt. Auf Elemente und andere Methoden wird im Code über ihre Namen verwiesen.

### **Implementation-Casts in ESDL verwenden:**

---

Anders als im Blockdiagrammeditor können Implementation-Casts im ESDL-Editor nur über die Schaltfläche eingefügt werden.



- Fügen Sie mit der Schaltfläche **Implementation cast** die gewünschte Anzahl Implementation-Casts hinzu.
- Verwenden Sie die Implementation-Casts im ESDL-Code gemäß den im Kapitel „Implementation-Casts in ESDL“ des ASCET Referenzhandbuchs dargestellten Regeln:
  - Implementation-Casts werden durch ihre Namen referenziert.
  - Sie werden immer in runde Klammern eingeschlossen.
  - Sie stehen unmittelbar vor dem Element, auf das sie sich beziehen.

- Bezieht sich ein Implementation-Cast auf das Ergebnis einer Operation, muss diese Operation in runde Klammern eingeschlossen werden. Die Operation kann Teil einer größeren Berechnung sein.

### **Hinweis**

*Wenn Sie einen Implementation-Cast auf eine logische Variable oder einen logischen Ausdruck anwenden, wird bei der Codegenerierung eine Fehlermeldung erzeugt.*

Wie im Blockdiagrammeditor können Sie die Anzeige der Implementation-Casts in der Liste „Elements“ mit **View** → **Show/Hide** → **Impl. Casts in Elements List** ein- und ausschalten (s. Seite 264).

### **Bearbeiten von ESDL-Code:**

---

Mit der Zwischenablage ist es möglich, ESDL-Code auszuschneiden, zu kopieren und in der gleichen oder einer anderen Methode einzufügen.

- Wählen Sie den zu bearbeitenden Code.

*Oder*

- Wählen Sie **Edit** → **Select All**, um den gesamten Code der aktuellen Methode auszuwählen.
- Wählen Sie **Edit** → **Cut** zum Ausschneiden des markierten Textes oder **Edit** → **Copy** zum Kopieren in die Zwischenablage.
- Klicken Sie an die Stelle, wo Sie den Text einfügen wollen.
- Wählen Sie **Edit** → **Paste** zum Einfügen des Textes aus der Zwischenablage.

### **Suchen/Ersetzen und Drucken von ESDL-Code:**

---

Eine hochentwickelte Such- und Ersetzungsfunktion ist im ESDL-Editor vorhanden. Sie arbeitet so wie die im C-Code-Editor und ist im Abschnitt „Suchen und Ersetzen von C-Code:“ auf Seite 342 genau beschrieben.

Sie können den ESDL-Code ausdrucken; das Verfahren ist dasselbe wie bei C-Code (s. „Ausdrucken des C-Codes:“ auf Seite 344).

#### 4.4.2 Spezifizieren von Modulen in ESDL

---

Module werden in ESDL-Code so spezifiziert wie Klassen, außer dass Prozesse anstelle von Methoden definiert werden. Des weiteren können in Modulen Messages definiert werden. Beim Spezifizieren von Klassen sind die **Message**-Schaltflächen im ESDL-Editor gesperrt (grau); beim Spezifizieren von Modulen sind sie aktiv.

#### 4.4.3 Analysieren von ESDL-Komponenten

---

Nach dem Erzeugen einer ESDL-Komponente können Sie mit der erzeugten Komponente auch experimentieren, um zu sehen, ob diese wie beabsichtigt funktioniert. Die Vorgehensweise ist dieselbe wie bei Blockdiagrammen - siehe „Analysieren von Komponenten“ auf Seite 274.

##### **Hinweis**

*Für ein einzelnes ASCET-Modul ohne Projektkontext kann Code nur im Physikalischen Experiment generiert und simuliert werden. Bei jedem anderen Codegenerator muss das Modul in ein Projekt integriert werden. Dafür kann zu jeder Klasse und jedem Modul auch ein sogenanntes Standardprojekt definiert werden. Nur auf diese Weise sind die benötigten Implementierungen verfügbar. Ohne Projektkontext fehlt die Konvertierungsformel sowie sämtliche Implementierungen importierter Größen.*

Wie bei Blockdiagrammen brauchen Sie nicht jedesmal den kompletten Code generieren; es gibt eine Analysefunktion.

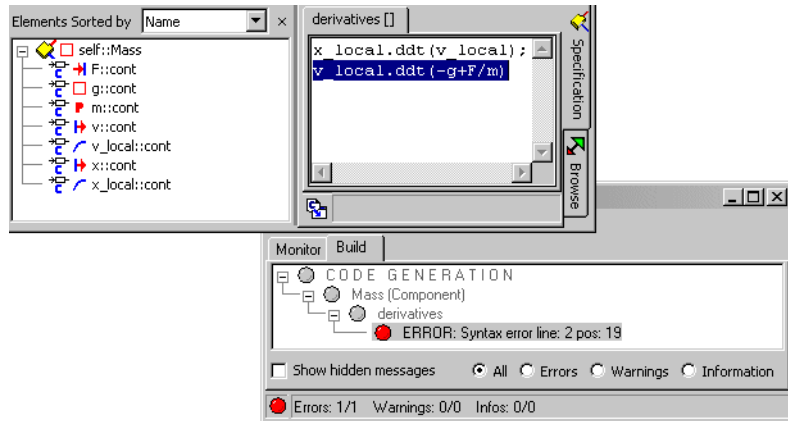
##### **Analysieren eines Diagramms:**

---

- Wählen Sie **Diagram → Analyze Diagram**, um das aktuelle ESDL-Diagramm zu analysieren.

Das Diagramm wird auf syntaktische Fehler geprüft. Entdeckte Fehler werden im ASCET Monitorfenster gemeldet.

- Klicken Sie auf eine Fehlermeldung, um den Fehler im ESDL-Editor automatisch hervorheben zu lassen.



## 4.5 Spezifizieren zeitkontinuierlicher Blöcke

Zeitkontinuierliche Blöcke (CT-Blöcke) werden zur Beschreibung von Modellen technischer Prozesse verwendet, die von ASCET Spezifikationen eingebetteter Software gesteuert werden. Auf diese Weise können ganze Regelkreise mit Hilfe von ASCET modelliert und erprobt werden. CT-Blöcke werden detailliert im Abschnitt „Zeitkontinuierliche Systeme“ des ASCET-Referenzhandbuchs erörtert.

Das Spezifizieren von CT-Blöcken geschieht wie das Spezifizieren anderer Komponenten; sie können als Blockdiagramme, C-Code oder ESDL-Code angegeben werden. Es gibt jedoch einen Unterschied in der Funktionalität zwischen durch Blockdiagramme und durch Code spezifizierten CT-Blöcken.

Mit *zeitkontinuierlichen Basisblöcken*, die durch Code angegeben sind, werden einfache physikalische Komponenten modelliert wie z.B. Räder, Bremsen oder Hydraulikleitungen. Solche Komponenten werden typischerweise durch Differentialgleichungen beschrieben, die sich leichter durch Code als durch Blockdiagramme angeben lassen. Diese werden in größeren Baugruppen zu zeitkontinuierlichen Strukturblöcken zusammengefasst und als Blockdiagramme spezifiziert.

Es ist möglich, Standard-ASCET-Klassen zu referenzieren, aber dies ist nur sinnvoll, wenn solche Klassen als Records, d.h. komplexe Variablen verwendet werden. Algorithmen, die als Standard-Klassen angegeben sind, können in CT-Blöcken nicht verwendet werden.

Dieses Kapitel zeigt, wie CT-Blöcke als Blockdiagramme und durch Code spezifiziert werden. Die verwendeten Editoren sind mit geringfügigen Abweichungen die gleichen wie für Standard-ASCET-Klassen. Es werden hier nur die Unterschiede zum Spezifizieren von Standardklassen erörtert; folglich sollten Sie mit den Kapiteln „Der Blockdiagrammeditor“ auf Seite 186, „Der C-Code-Editor“ auf Seite 329 und „Der ESDL-Editor“ auf Seite 351 vertraut sein.

#### 4.5.1 Zeitkontinuierliche Blöcke als Blockdiagramme

---

##### **Spezifizieren eines zeitkontinuierlichen Blocks:**

---

- Wählen Sie im Komponentenmanager den Ordner für den neuen Block aus.
- Wählen Sie **Insert** → **Continuous Time Block** → **Blockdiagram**.
- Geben Sie einen Namen für die Komponente ein und drücken Sie <ENTER>.
- Wählen Sie in der Menüzeile **Component** → **Edit Item**

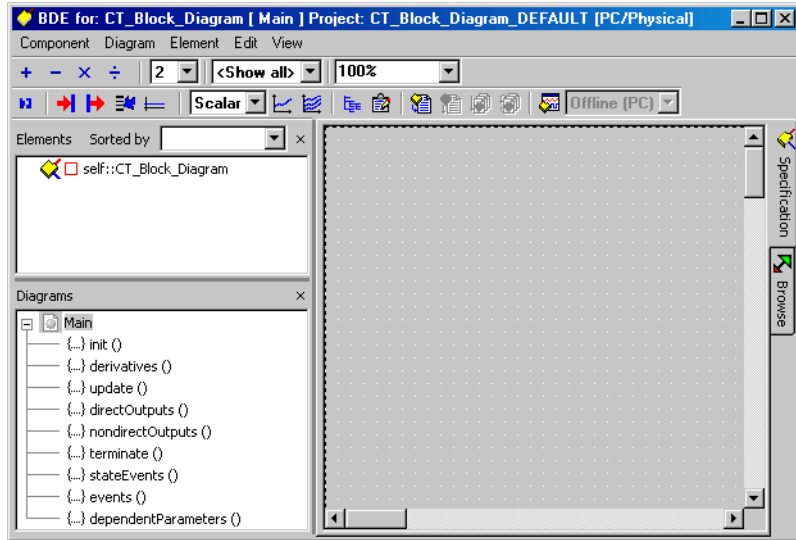
*oder*

- drücken Sie <ENTER>

*oder*

- doppelklicken Sie in der Liste „1 Database“ auf den Namen der Komponente.

Der Blockdiagrammeditor für CT-Blöcke wird geöffnet.



Wie schon an anderer Stelle erwähnt, besteht ein als Blockdiagramm spezifizierter CT-Block hauptsächlich aus Basisblöcken, die grafisch zu einer größeren Baugruppe verbunden werden. Das Spezifizieren eines Diagrammes ist dem Spezifizieren von Klassen sehr ähnlich; die wichtigsten Unterschiede sind nachstehend aufgelistet:

- Es gibt nur vier Operatoren: Addition, Subtraktion, Multiplikation und Division. Falls komplexere, nichtlineare Operatoren benötigt werden, können sie als separate Blöcke spezifiziert werden.
- Die Basiselemente, die in CT-Blöcken definiert werden können, sind verschieden; davon ausgenommen sind Kennlinien und -felder, die wie in Klassen arbeiten. Die Basiselemente, die in CT-Blöcken definiert werden können, sind im Abschnitt „Blockschnittstellen“ des ASCET-Referenzhandbuchs beschrieben.
- Es gibt nur ein Diagramm in einem CT-Block; es können keine zusätzlichen Diagramme definiert werden. Das Diagramm kann jedoch hierarchisch sein.
- Es müssen keine Schnittstellen definiert werden. Eine ganze Reihe von Methoden ist vordefiniert, wenn ein CT-Block erstellt wird. Diese können nicht verändert werden; auch können keine neuen Methoden definiert werden.

- Es können nur Klassen und andere CT-Blöcke referenziert werden. Klassen können nur verwendet werden, um Records zu definieren, nicht um Funktionalität zu spezifizieren. Wenn eine Klasse referenziert wird, werden Sie gefragt, ob sie als Eingabe oder Ausgabe verwendet werden soll.
- Es gibt keine Sequenzaufrufe in einem CT-Block. Die Auswertungsreihenfolge wird automatisch bestimmt.

Von diesen Dingen abgesehen, arbeiten alle Funktionen des Blockdiagrammeditors so wie für Klassen; alle vorhandenen Befehle haben den gleichen Effekt.

#### 4.5.2 Zeitkontinuierliche Blöcke als C-Code

---

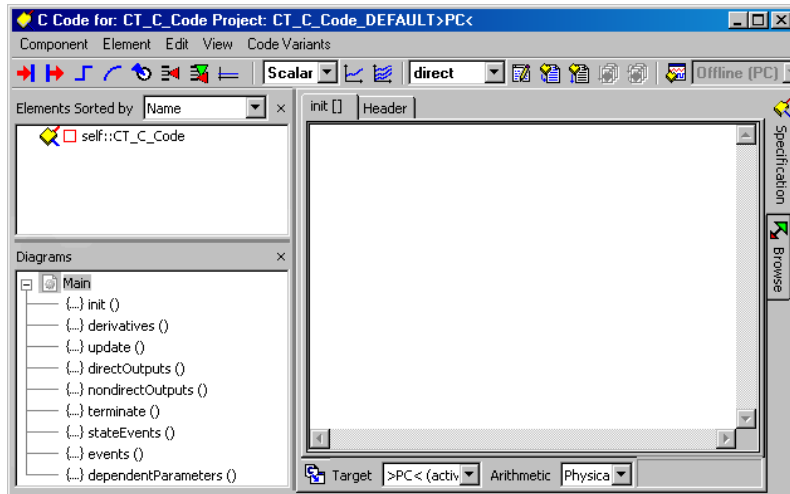
Die gleichen Betrachtungen wie bei CT-Blöcken als Blockdiagramme gelten auch für C-Code, d.h. alles, außer den oben aufgelisteten Dingen, arbeitet wie bei Klassen. Ein Unterschied zwischen CT-Blöcken als Blockdiagramme und als C-Code ist, dass sich die definierbaren Basiselemente unterscheiden. CT-Blöcke in C-Code werden im ASCET-Referenzhandbuch, Abschnitt „Modellieren in C“, behandelt.

Als C-Code spezifizierte CT-Blöcke werden hauptsächlich benutzt, um Basis-komponenten eines Streckenmodells vorzugeben. Die zum Modellieren solcher Komponenten verwendeten Differentialgleichungen können schneller und klarer durch Code als durch Blockdiagramme angegeben werden.

#### **Erstellen eines zeitkontinuierlichen Blocks:**

---

- Wählen Sie im Komponentenmanager den Ordner für den neuen Block aus.
  - Wählen Sie **Insert** → **Continuous Time Block** → **C Code**.
  - Geben Sie einen Namen für die Komponente ein und drücken Sie <ENTER>.
  - Wählen Sie **Component** → **Edit Item**
- oder*
- drücken Sie <ENTER>
- oder*
- doppelklicken Sie in der Liste „1 Database“ auf den Namen der Komponente.
- Der C-Code-Editor für CT-Blöcke wird geöffnet.



- Wählen Sie in der Schaltflächenleiste aus dem rechten Kombikästchen `direct` oder `nondirect` aus.
- Geben Sie je nach Bedarf den Code ein oder importieren Sie externe Module.

Die Benutzung des C-Code-Editors ist in Kapitel 4.3 ausführlich beschrieben.

In C-Code angegebene CT-Blöcke unterstützen entweder direkte oder indirekte Ausgaben, aber nicht beide. Die Einstellung wird im Kombikästchen „Block Behavior“ vorgenommen. Wurde `direct` ausgewählt, steht nur die Methode `directOutputs[]` zur Verfügung, bei `nondirect` nur die Methode `nondirectOutputs[]`. Genauere Angaben zu direkten und indirekten Ausgaben finden Sie im Abschnitt „Algebraische Schleifen“ im ASCET-Referenzhandbuch.

#### 4.5.3 Zeitkontinuierliche Blöcke in ESDL-Code

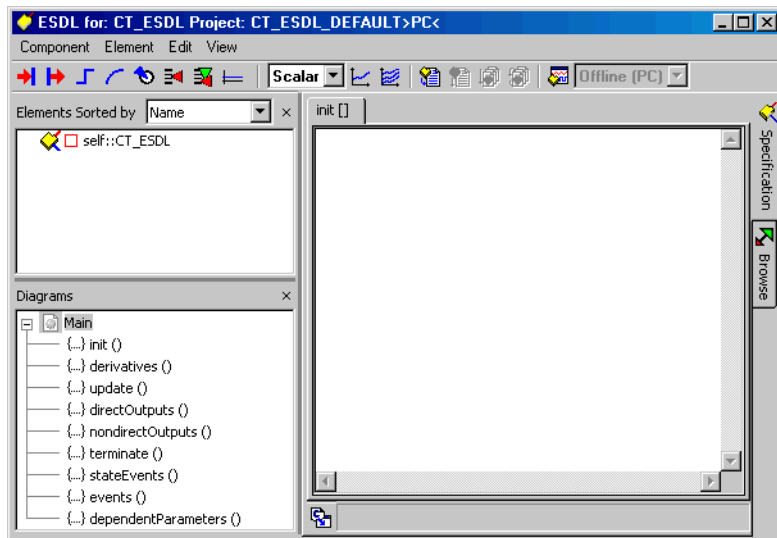
Das Spezifizieren von CT-Blöcken in ESDL ist wieder ähnlich zum Spezifizieren von Klassen, mit Ausnahme der oben ausgeführten Unterschiede. In ESDL sind die gleichen Basiselemente für CT-Blöcke wie in C verfügbar; beide Blocktypen dienen dem gleichen Zweck.

##### **Erstellen eines zeitkontinuierlichen Blocks:**

- Wählen Sie im Komponentenmanager den Ordner für den neuen Block aus.



- Wählen Sie **Insert** → **Continuous Time Block** → **ESDL**.
  - Geben Sie einen Namen für die Komponente ein und drücken Sie <ENTER>.
  - Wählen Sie **Component** → **Edit Item**  
oder  
• drücken Sie <ENTER>  
oder  
• doppelklicken Sie in der Liste „1 Database“ auf den Namen der Komponente.
- Der ESDL-Editor für CT-Blöcke wird geöffnet.



- Geben Sie den Block-Code ein.

Die Benutzung des ESDL-Editors ist ausführlich in „Der ESDL-Editor“ auf Seite 351 beschrieben.

#### 4.5.4 Experimentieren mit zeitkontinuierlichen Blöcken

Man kann Offline-Experimente mit individuellen CT-Blöcken starten. Die Blöcke können komplex sein, d.h. andere Blöcke referenzieren. Es ist jedoch nicht möglich, auf diese Weise kombinierte Experimente durchzuführen. Kombinierte Experimente bestehen aus Klassen und CT-Blöcken und werden zur

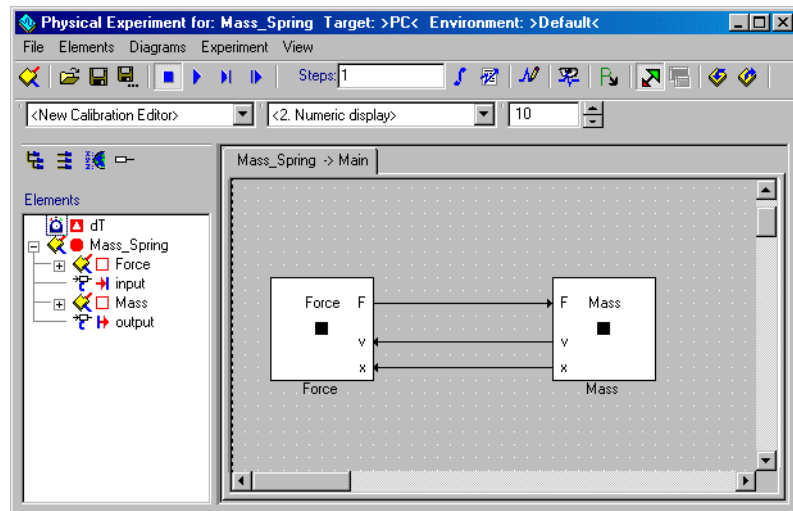
Simulation von Regelkreisen verwendet, die aus einem Controller- und einem Streckenmodell bestehen. Mit solchen Regelkreisen kann in Projekten experimentiert werden.

Offline mit individuellen CT-Blöcken zu experimentieren ist so, wie mit Klassen zu experimentieren (siehe „Analysieren von Komponenten“), außer dass es keinen Ereignisgenerator gibt. Stattdessen muss eine Integrationsmethode spezifiziert werden.

### Experimentieren mit einem zeitkontinuierlichen Block:

- Wählen Sie im Komponentenmanager den Block aus, mit dem Sie experimentieren wollen.
- Wählen Sie **Build** → **Experiment** → **Offline** zum Öffnen der Experimentierumgebung für den Block.

Sie können das Experiment auch aus dem Editor heraus starten.



- Bereiten Sie die Datengenerator-, Mess- und Verstellfenster vor und starten Sie das Experiment.

Außer bei der Konfiguration der Integrationsmethode arbeitet die Offline-Experimentierumgebung wie gewöhnlich. Die Benutzung der Experimentierumgebung ist in „Die Experimentierumgebung“ auf Seite 603 beschrieben.

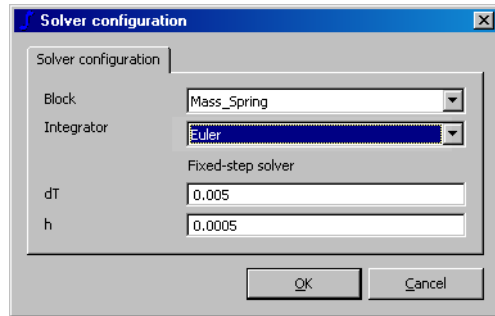
## Konfigurieren der Integrationsmethode:

- Wählen Sie **Experiment** → **Open CT Solver**.  
oder



- Klicken Sie auf die Schaltfläche **Open CT Solver**.

Das Fenster „Solver Configuration“ öffnet sich.



- Wählen Sie die Integrationsmethode aus dem Kombikästchen „Integrator“.  
Abhängig von der gewählten Methode werden verschiedene Parameterfelder angezeigt.
- Stellen Sie die Parameter  $\Delta T$  und  $h$  ein.

### Hinweis

*Beim Online-Experimentieren (s. „Online- und Offline-Experimentieren“) wird  $\Delta T$  im Register „OS“ des Projekteditors spezifiziert. Die Eingabefläche im Fenster „Solver Configuration“ ist ausgegraut; Änderungen während der Messung sind nicht möglich.*

- Klicken Sie auf **OK**.

Die Integrationsmethode wird immer für das gesamte Modell vorgegeben (globale Integration). Beim Experimentieren im Projektzusammenhang kann für jeden Block eine separate Integrationsmethode vorgegeben werden (mehrklassig). Man kann die Integrationsmethode während des Experimentes wechseln. Es sind die folgenden Integrationsmethoden vorhanden:

- Adams-Moulton 2
- Euler
- Mulstep 2
- Heun
- Runge-Kutta 4
- Variable-step Dormand/Prince RK5
- Variable-step Calvo 6(5)
- Variable-step Dormand/Prince RK8
- Variable-step implicit RK2
- Variable-step implicit RK4
- Variable-step implicit Gear 1
- Variable-step implicit Gear 2

Bei den ersten fünf können nur die Parameter  $dT$  und  $h$  eingestellt werden. Für die weiteren Methoden sind zusätzliche Parameter vorhanden. Die Integrationsmethoden sind im Kapitel „Lösen von Differentialgleichungen - Integrationsalgorithmen“ im ASCET-Referenzhandbuch beschrieben. Die Parameter haben folgende Bedeutungen:

- $dT$  ist der Kommunikationszeitrahmen, d.h. das Intervall, in dem der Block mit der Außenwelt kommuniziert.
- $h$  ist die Integrationsschrittgröße, d.h. das Intervall interner Berechnungen.
- *Initial h* ist die anfängliche Integrationsschrittgröße für die Integrationsmethoden mit variabler Schrittweite.
- *Minimum h* und *Maximum h* sind untere und obere Grenze für den Parameter  $h$ , der entsprechend der Modelldynamik bei Einsatz von Integrationsmethoden mit variabler Schrittweite berechnet wird. *Maximum h* muss kleiner oder gleich  $dT$  sein.
- *Relative error* und *Absolute error* sind der zulässige relative Fehler und der zulässige absolute Fehler für die Berechnung von  $h$ ; verfügbar bei Einsatz von Integrationsmethoden mit variabler Schrittweite.
- *Max. iterations* ist die Höchstzahl an Iterationen zur Berechnung des Parameters  $h$ . Wenn so viele Schritte wie angegeben ausgeführt wurden, wird der  $h$ -Wert von dieser Stelle verwendet.

## Die Zykluszeit überwachen

ASCET bietet über das Register „OS“ im Projekteditor die Möglichkeit, die Zykluszeit einer Timer-Task zu überwachen (s. „Überwachungsoption“ auf Seite 430).

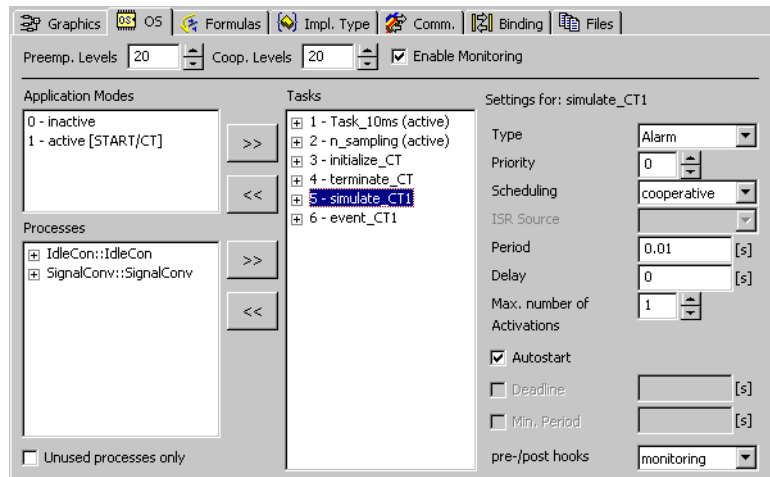
### Überwachung der Zykluszeit aktivieren:

- Öffnen Sie einen CT-Block.
- Wählen Sie **Component** → **Default Project** → **Edit**, um das Standardprojekt für den CT-Block zu öffnen.

Oder

- Öffnen Sie ein hybrides Projekt (s. Kapitel 4.8.7 „Hybride Projekte“ sowie „Projekte und hybride Projekte“ im ASCET Referenzhandbuch).
- Aktivieren Sie im Projekteditor das Register „OS“.

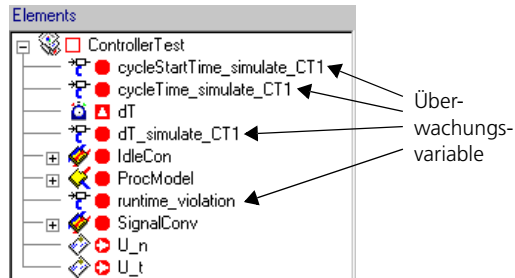
In der Liste „Tasks“ finden Sie die automatisch definierten Tasks für den CT-Block (s.a. „Kombinieren zeitkontinuierlicher Blöcke mit Modulen“ im ASCET Referenzhandbuch).



- Wählen Sie die Task `simulate_CT1`.

- Wählen Sie aus dem Kombikästchen „pre/post hooks“ die Option **Monitoring**.

Bei der nächsten Codegenerierung für ein Experiment werden vier Überwachungsvariablen erzeugt. Sie können diese in den Messfenstern anzeigen oder in den Data-Logger (nur Offline-Experiment, s. „Der Data-Logger“ auf Seite 641) schreiben.



Dabei ist für CT-Blöcke folgendes zu beachten. Die Simulation eines Projekts mit CT-Blöcken erlaubt für die gewählte Integrationsmethode das Verstellen zweier Parameter:

- die externe Kommunikationsschrittweite  $dT$  (s. „Externe Kommunikationsschrittweite  $dT$ “ im ASCET-Referenzhandbuch)
- und die Integrationsschrittweite  $h$  (s. „Integrationsschrittweite  $h$ “ im ASCET-Referenzhandbuch).

Die Zykluszeit für die Simulationstask `simulate_CTn` summiert die CPU-Zeit für alle Integrationsschritte, die während eines  $dT$ -Schritts ausgeführt wurden, und ist daher um so höher, je größer das Verhältnis  $dT/h$  ist. Ein Korrekturfaktor lässt sich jedoch nicht angeben, da außerdem noch andere Faktoren wie Größe und Art des Modells eine Rolle spielen.

Ein mit Simulink erstellter CT-Block hingegen unterscheidet nicht zwischen  $dT$  und  $h$ , beide haben immer denselben Wert. Wird ein solcher CT-Block nach ASCET importiert, gibt es keine Möglichkeit, im Experiment die Integrationsmethode oder  $h$  zu wählen. Um für CT-Blöcke in ASCET und Simulink vergleichbare Zykluszeiten zu erhalten, muss daher im ASCET Block für  $dT$  und  $h$  die Integrationsschrittweite des Simulink-Modells gewählt werden.

## 4.6 Der Boolesche Tabelleneditor

---

Eine *Boolesche Tabelle* bildet logisches Verhalten ab. Ihre Spalten beinhalten verschiedene logische Ein- und Ausgänge, deren Werte spezifiziert werden können. Die Zeilen, die sogenannten *Kombinationen*, beinhalten verschiedene Kombinationen von Eingangswerten und die zugehörigen Ausgangswerte. Die Boolesche Tabelle wird in anderen Komponenten als logische Verknüpfung verwendet. Sie können, wie an andere Komponenten, an die Boolesche Tabelle eine Notiz anhängen (s. Seite 231).

Mit dem Booleschen Tabelleneditor können Sie die Boolesche Tabelle spezifizieren. Beim Schließen des Editors wird automatisch ESDL-Code generiert, der das spezifizierte Verhalten der Ausgänge beschreibt. Die Eingänge werden als logische Variablen generiert und für die Ausgänge wird jeweils eine entsprechende Methode generiert.

### Erstellen einer Booleschen Tabelle:

---

- Wählen Sie im Komponentenmanager den Ordner für die neue Tabelle aus.
- Wählen Sie die Menüfunktion **Insert** → **Boolean Table**

oder



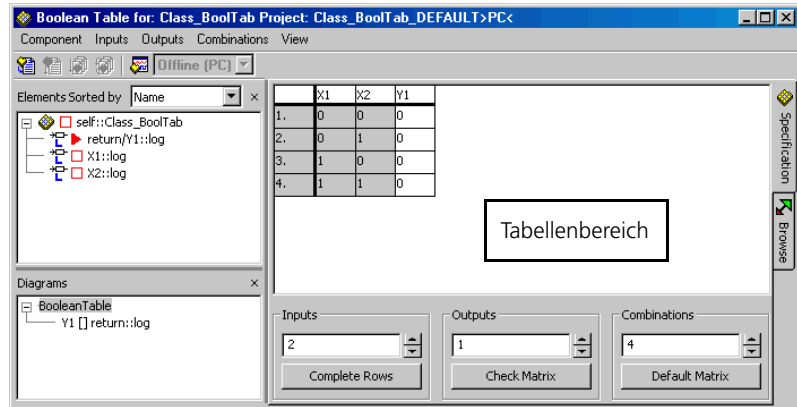
- klicken Sie auf die Schaltfläche **Insert Boolean Table**.
- Geben Sie einen Namen für die Komponente ein und drücken Sie <ENTER>. Eine neue Boolesche Tabelle wird erzeugt.
- Wählen Sie die Menüfunktion **Component** → **Edit Item**

oder

- drücken Sie <ENTER>

oder

- doppelklicken Sie auf das gewählte Element. Der Boolesche Tabelleneditor wird geöffnet.



In diesem Editor ist keine Manipulation der Schnittstellen möglich. Desweiteren können die Ein- und Ausgänge nicht von der Liste „Elements“ aus bearbeitet werden. Daher gibt es die meisten Menüpunkte aus den anderen Editoren hier nicht.

Die Hintergrundfarben der Tabelle werden im ASCET-Optionsfenster eingestellt (siehe „Optionen für Tabelleneditoren“ auf Seite 59).

### Beschreibung der Menüfunktionen

- **Component**
  - *Clean code generation directory*  
Löscht alle Dateien im Codegenerierungsverzeichnis.
  - *Touch*  
Erzwingen einer Neugenerierung bei der nächsten Codegenerierung  
(*Flat* → die editierte Komponente,  
*Recursive* → für Boolesche Tabellen deaktiviert).
  - *Generate Code*  
Der Code für eine Komponente wird generiert.
  - *Compile*  
Kompiliert den generierten Code.



- *Open Experiment*  
Ein Experiment wird gestartet.
- *Default Project*  
Erlaubt die Bearbeitung des Standardprojektes, das beim Experimentieren mit Komponenten verwendet wird  
(*Edit* → öffnet einen Editor für das Standardprojekt,  
*Resolve Globals* → erzeugt für jedes importierte Element der Komponente, für das kein exportiertes Element vorhanden ist, automatisch ein globales Element,  
*Delete Unused Globals* → löscht nicht genutzte globale Elemente).
- *Edit Layout*  
Der Layout-Editor wird geöffnet.
- *Edit Data*  
Der Dateneditor für eine Komponente wird geöffnet. Das Durchsuchen der Daten einer Komponente ist möglich.
- *Edit Implementation*  
Der Implementierungseditor wird geöffnet. Das Durchsuchen der Implementierungen einer Komponente ist möglich.
- *Edit Notes*  
Öffnet den Notizeneditor - Hier können Notizen zur Komponente gemacht werden.
- *Check Dependency*  
Es wird überprüft, ob die Zuweisung der formalen Parameter zu den Modellparametern korrekt ist.
- *Export Data*  
Exportieren des Datensatzes einer Komponente.
- *Show Path*  
Zeigt den Pfad einer aufgenommenen Komponente.
- *Copy Path to Clipboard*  
Kopiert den Pfad einer aufgenommenen Komponente in die Zwischenablage  
(*Model Path* → Pfad im Modell,  
*Model asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente im Modell als Hyperlink ermöglicht,  
*Database Path* → Pfad in der Datenbank,

*Dabase asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente in der Datenbank als Hyperlink ermöglicht).

– *Delete Unused Elements*

Löscht die Elemente aus dem Feld „Elements“, die kein Vorkommen im Diagramm haben.

– *File out Generated Code*

Speichern des generierten Codes im Dateisystem  
(*Flat* → die editierte Komponente,  
*Recursive* → auch die referenzierten Komponenten).

– *View Generated Code*

Generiert den Code für die Komponente und zeigt diesen in einem Texteditor an.

Der Texteditor kann im ASCET-Optionsfenster, Knoten „ASCII Editor“ (s. Seite 61), gewählt werden.

– *File Out Generic Code For External Make*

Speichern des generierten Codes in ein beliebiges Verzeichnis zur späteren Weiterverarbeitung durch separate Werkzeuge, z.B. in einem externen Make/Build - Prozess.

– *Exit*

Beenden des Editors für Boolesche Tabellen.

• **Inputs**

– *Add*

Fügt nach dem letzten vorhandenen Eingang einen neuen Eingang hinzu.

– *Insert*

Fügt links vom ausgewählten Eingang einen neuen Eingang hinzu.

– *Delete*

Entfernt den ausgewählten Eingang.

– *Rename*

Umbenennen des ausgewählten Eingangs.

– *Move Left*

Verschiebt den ausgewählten Eingang nach links.

– *Move Right*

Verschiebt den ausgewählten Eingang nach rechts.

- **Outputs**

Das Menü **Outputs** dient der Bearbeitung von Ausgängen. Die Menüfunktionen funktionieren auf dieselbe Weise wie im Menü **Inputs**.

- **Combinations**

- *Add*

Fügt nach der letzten Zeile der Tabelle eine neue Kombination hinzu.

- *Insert*

Fügt über der ausgewählten Kombination eine neue Kombination hinzu.

- *Delete*

Entfernt die ausgewählte Kombination.

- *Move Up*

Verschiebt die ausgewählte Kombination in der Tabelle nach oben.

- *Move Down*

Verschiebt die ausgewählte Kombination in der Tabelle nach unten.

- **View**

- *Show/Hide*

Zeigt/verbirgt verschiedene Teile des Editors oder der Komponente.  
(*Element List* → Liste „Elements“,  
*Diagram List* → Liste „Diagrams“)

### Boolesche Tabellen spezifizieren

Standardmäßig werden für eine Boolesche Tabelle zwei Eingänge  $x_1$  und  $x_2$  und ein Ausgang  $y_1$  sowie vier Kombinationen (1 - 4) für die möglichen Belegungen der Eingänge erzeugt. Die Ausgangswerte jeder Kombination liegen auf Null (*false*).

Sie können die Werte der Ein- und Ausgänge jeder Kombination bearbeiten.

#### **Spezifizieren der Ein- und Ausgänge:**

	$x_1$
1.	*
2.	1
3.	0
	*

- Klicken Sie im Tabellenbereich auf den Ein- oder Ausgangswert, den Sie ändern wollen. Ein Kombikästchen erscheint.

- Wählen Sie aus dem Kombikästchen den Wert, den Sie zuweisen wollen (0 - `false`, 1 - `true`).

#### **Hinweis**

*Eingänge können außer 0 und 1 auch den Wert \* (beliebig) haben.*

Wenn Sie weitere Ein- oder Ausgänge hinzufügen, gehen Sie wie folgt vor.

#### **Hinzufügen von Ein- oder Ausgängen:**

- Wählen Sie **Inputs** → **Add** oder **Outputs** → **Add**.

*oder*

- Klicken Sie mit der rechten Maustaste in den Tabellenbereich und wählen Sie aus dem Kontextmenü **Inputs** → **Add** oder **Outputs** → **Add**.

Ein Ein- oder Ausgang wird hinzugefügt.

*Oder*

- Stellen Sie mit den **Pfeil**-Schaltflächen im Feld „Inputs“ oder „Outputs“ unterhalb des Tabellenbereiches die gewünschte Anzahl Ein- oder Ausgänge ein.

Die entsprechenden Ein- und Ausgänge werden hinzugefügt.

Neu hinzugefügte Eingänge sind standardmäßig mit \* (beliebig) belegt, neue Ausgänge mit 0 (`false`).

Kombinationen werden mit **Combinations** → **Add** oder über das Feld "Combinations" hinzugefügt.

Sie können jetzt die Ein- und Ausgangswerte wie oben beschrieben belegen. Anstatt den Wert jedes Eingangs einzeln zu bearbeiten, können Sie eine Standardmatrix automatisch erstellen.

#### **Erstellen einer Standardmatrix**

- Erzeugen Sie die gewünschte Anzahl von Eingängen.

- Klicken Sie auf die Schaltfläche **Default Matrix**.

Es werden so viele Kombinationen erzeugt, dass sämtliche möglichen Belegungen der Eingangswerte erfasst sind. Die Anzahl der Ausgänge wird nicht geändert; die Werte sämtlicher Ausgänge werden auf 0 (`false`) gesetzt.

Wenn Sie z.B. drei Eingänge und einen Ausgang erzeugt haben, bekommen Sie mit **Default Matrix** folgende Belegung:

	X1	X2	X3	Y1
1.	0	0	0	0
2.	0	0	1	0
3.	0	1	0	0
4.	0	1	1	0
5.	1	0	0	0
6.	1	0	1	0
7.	1	1	0	0
8.	1	1	1	0

### Löschen von Ein- und Ausgängen oder Kombinationen:

---

Ein- und Ausgänge oder Kombinationen können wie folgt gelöscht werden:

- Markieren Sie den Ein- oder Ausgang oder die Kombination, den Sie löschen wollen.
- Wählen Sie **Inputs** → **Delete**, **Outputs** → **Delete** oder **Combinations** → **Delete**.

oder

- Klicken Sie mit der rechten Maustaste in den Tabellenbereich und wählen Sie aus dem Kontextmenü **Inputs** → **Delete**, **Outputs** → **Delete** oder **Combinations** → **Delete**.

Der gewählte Ein- oder Ausgang oder die Kombination wird entfernt.

*Oder*

- Stellen Sie mit den **Pfeil**-Schaltflächen im Feld „Inputs“, „Outputs“ oder „Combinations“ unterhalb des Tabellenbereiches die gewünschte Anzahl Ein- oder Ausgänge bzw. Kombinationen ein.

Die nötige Anzahl Ein- oder Ausgänge bzw. Kombinationen wird entfernt.

### **Umbenennen von Ein- und Ausgängen:**

---

Sie haben die Möglichkeit, Ein- oder Ausgänge umzubenennen. Kombinationen können nicht umbenannt werden.

- Markieren Sie im Tabellenbereich den Ein- oder Ausgang, den Sie umbenennen wollen.
- Wählen Sie **Inputs** → **Rename** oder **Outputs** → **Rename**.

*oder*

- Klicken Sie mit der rechten Maustaste auf den ausgewählten Ein-/Ausgang und wählen Sie aus dem Kontextmenü **Inputs** → **Rename** oder **Outputs** → **Rename**.

Ein Dialogfenster öffnet sich.

- Geben Sie den neuen Namen ein und bestätigen Sie mit **OK**.

### **Verschieben von Tabellenspalten und -zeilen:**

---

Sie können die Spalten der Ein- und Ausgangswerte sowie die Zeilen der Kombinationen verschieben.

- Markieren Sie im Tabellenbereich den Eingang, die Sie verschieben wollen.
- Wählen Sie **Inputs** → **Move Left** oder **Inputs** → **Move Right**.

*oder*

- Klicken Sie mit der rechten Maustaste auf den markierten Eingang und wählen Sie aus dem Kontextmenü **Inputs** → **Move Left** oder **Inputs** → **Move Right**.

Der *Inhalt* der Spalte wird nach links bzw. rechts verschoben. Die *Namen* der Spalten behalten die ursprüngliche Reihenfolge.

- Verschieben Sie die Ausgänge mit **Outputs** → **Move Left** oder **Outputs** → **Move Right**.

#### Hinweis

*Eingänge können nur mit Eingängen vertauscht werden, Ausgänge nur mit Ausgängen. Die letzte Eingangsspalte kann nicht mit der ersten Ausgangsspalte vertauscht werden.*

- Verschieben Sie die Kombinationen mit **Combinations** → **Move Up** oder **Combinations** → **Move Down**.

Wenn Sie die Werte der Ein- und Ausgänge von Hand eingestellt haben oder wenn Sie Ein- oder Ausgänge entfernt haben, ist es ratsam, die Tabelle auf Inkonsistenzen zu prüfen.

## Überprüfen der Tabelle:

---

- Wenn Sie die Tabelle nach Ihren Wünschen eingerichtet haben, klicken Sie auf **Check Matrix**.

Sind Kombinationen mit identischen Eingangswerten unterschiedliche Ausgabewerte zugewiesen, werden die betreffenden Zeilen rot umrandet.

	X1	X2	X3	Y1
1.	0	0	0	0
2.	0	0	1	0
3.	0	1	0	1
4.	0	1	1	0
5.	1	0	0	0
6.	1	0	1	0
7.	0	1	0	0
8.	1	1	0	0
9.	1	1	1	0

- Löschen Sie die überzähligen Kombinationen mit **Combinations** → **Delete**.

Wenn Sie die Tabelle schließen, ohne die fehlerhaften Kombinationen zu korrigieren, wird eine Warnung angezeigt:

Your table has bad entries. Continue closing anyway?

- Bestätigen Sie die Warnung mit **Cancel**, um in den Editor zurückzukehren und den Fehler zu korrigieren.
- Klicken Sie auf **OK**, um die fehlerhafte Tabelle zu schließen.

Wenn Sie mit der Booleschen Tabelle experimentieren wollen, können Sie das Experiment wie bei anderen Komponenten mit der Schaltfläche **Open Experiment for selected Experiment Target** starten.



## 4.7 Der Editor für Bedingunstabellen

Eine *Bedingunstabelle* bildet (ähnlich der Booleschen Tabelle) logisches Verhalten ab. Ihre Spalten und Zeilen erzeugen eine logische Matrix mit verschiedenen Bedingungen und Aktionen.

In einer Bedingunstabelle können Basiselemente verwendet werden. Desweiteren ist der Zugriff auf öffentliche Methoden aufgenommener Klassen möglich. Die Funktionalität einer Bedingunstabelle ist äquivalent zu einer `If...Then...ElseIf...Then`-Konstruktion. Jede Zeile entspricht dabei einer `If`- oder `Elseif`-Abfrage. Abb. 4-1 zeigt ein Beispiel für eine Bedingunstabelle. Die ersten drei Spalten („cont“, „log“ und „enum\_1“) enthalten die durch UND verknüpften Bedingungen, die folgenden Spalten („out“, „disc“ und „methods“) enthalten die zugehörigen Aktionen.

	cont	log	enum_1	out	disc	methods
1.	> -1.0 < 0.0	== true	== Speed	= 100.0	= 5	ClassXYZ.doThis(cont,out)
2.	== 0.0	== true	== Speed	= 0.0	= 10	ClassABC.doThat()
3.	> 0.0 < 1.0	== false	== Distance	= -100.0	= ClassX.GetA()	ClassABC.doThat() ClassXYZ.doThis(out,cont)
default	*	*	*	= 3.14159	= 0	*

**Abb. 4-1** Bedingunstabelle – Beispiel

Das Beispiel in Abb. 4-1 entspricht dem folgenden Code:

```
if ((cont > -1.0) && (cont < 0.0) && (log == true)
    && (enum_1 == Speed)){
    out = 100.0;
    disc = 5;
    ClassXYZ.doThis(cont, out);
} else
if ((cont == 0.0) && (log == true)
    && (enum_1 == Speed)){
    out = 0.0;
    disc = 10;
    ClassABC.doThat();
} else
if ((cont > 0.0) && (cont < 1.0) && (log == false)
    && (enum_1 == Distance)){
    out = -100.0;
```

```

    disc = ClassX.GetA();
    ClassABC.doThat();
    ClassXYZ.doThis(out,cont);
} else {
    out = 3.14159;
    disc = 0;
}

```

Eine Bedingungstabelle wird durch den Aufruf einer Triggermethode aktiviert. Bei Aktivierung wird die passende Zeile ermittelt und die darin enthaltenen Aktionen und Berechnungen ausgeführt.

Bedingungstabellen sind ESDL-Klassen, die in einem speziellen *Editor für Bedingungstabellen* spezifiziert werden. Bedingungen und Aktionen werden in zwei getrennten Bereichen der Tabelle, Bedingungs- und Aktionsbereich, eingegeben. Jeder Bereich hat eine einstellbare Anzahl von Spalten und eine eigene, frei wählbare Hintergrundfarbe (siehe „Optionen für Tabelleneditoren“ auf Seite 59).

### Erstellen einer Bedingungstabelle:

---

- Wählen Sie im Komponentenmanager den Ordner für die neue Tabelle aus.

- Wählen Sie **Insert** → **Conditional Table**

oder



- klicken Sie auf die Schaltfläche **Insert Conditional Table**.

- Geben Sie einen Namen für die Komponente ein und drücken Sie <ENTER>.

Eine neue Bedingungstabelle wird erzeugt.

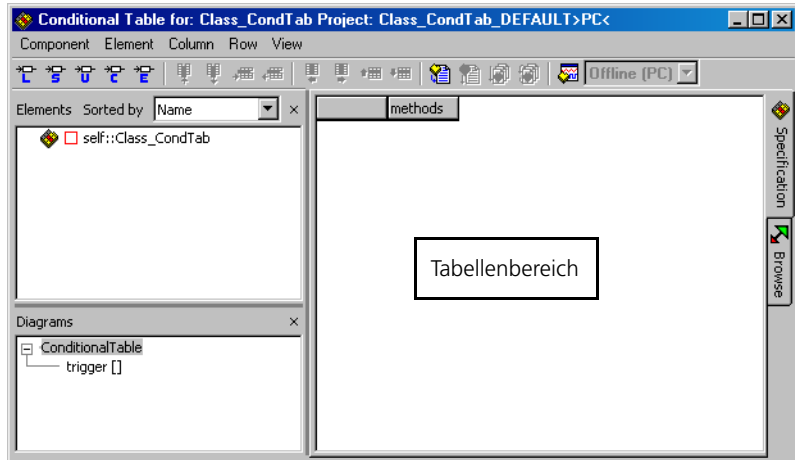
- Wählen Sie **Component** → **Edit Item**

oder

- drücken Sie <ENTER>

oder

- doppelklicken Sie auf das gewählte Element.  
Der Editor für Bedingungstabellen wird geöffnet.



Die öffentliche Triggermethode `trigger` wird in jedem Fall automatisch angelegt. Sie ist zwingend erforderlich und kann nicht gelöscht oder umbenannt werden. Sie können keine weiteren Methoden oder Diagramme hinzufügen.

Die Spalte „methods“ wird ebenfalls automatisch angelegt. Sie steht immer am Ende des Aktionsbereichs und kann nicht gelöscht oder verschoben werden. Bis auf die Spalte „methods“ ist die Tabelle leer.

Kapitel 4.7.1 beschreibt die Einrichtung einer Bedingungstabelle, Kapitel 4.7.2 beschreibt die Spezifikation der Bedingungen und Aktionen und Kapitel 4.7.3 beschreibt das Experimentieren mit Bedingungstabellen.

### *Beschreibung der Menüfunktionen*

- **Component**
  - *Clean code generation directory*  
Löscht alle Dateien im Codegenerierungsverzeichnis.
  - *Touch*  
Erzwingen einer Neugenerierung bei der nächsten Codegenerierung  
(*Flat* → die editierte Komponente,  
*Recursive* → auch die referenzierten Komponenten).

- *Generate Code*  
Der Code für eine Komponente wird generiert.
- *Compile*  
Kompiliert den generierten Code.
- *Open Experiment*  
Ein Experiment wird gestartet.
- *Default Project*  
Erlaubt die Bearbeitung des Standardprojektes, das beim Experimentieren mit Komponenten verwendet wird  
(*Edit* → öffnet einen Editor für das Standardprojekt,  
*Resolve Globals* → erzeugt für jedes importierte Element der Komponente, für das kein exportiertes Element vorhanden ist, automatisch ein globales Element  
*Delete Unused Globals* → löscht nicht genutzte globale Elemente).
- *Edit Layout*  
Der Layout-Editor wird geöffnet.
- *Edit Data*  
Der Dateneditor für eine Komponente wird geöffnet. Das Durchsuchen der Daten einer Komponente ist möglich.
- *Edit Implementation*  
Der Implementierungseditor wird geöffnet. Das Durchsuchen der Implementierungen einer Komponente ist möglich.
- *Edit Notes*  
Öffnet den Notizeneditor - Hier können Notizen zur Komponente gemacht werden.
- *Check Dependency*  
Es wird überprüft, ob die Zuweisung der formalen Parameter zu den Modellparametern korrekt ist.
- *Export Data*  
Exportieren des Datensatzes einer Komponente.
- *Show Path*  
Zeigt den Pfad einer aufgenommenen Komponente.

- *Copy Path to Clipboard*  
Kopiert den Pfad einer aufgenommenen Komponente in die Zwischenablage  
(*Model Path* → Pfad im Modell,  
*Model asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente im Modell als Hyperlink ermöglicht,  
*Database Path* → Pfad in der Datenbank,  
*Database asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente in der Datenbank als Hyperlink ermöglicht).
- *File Out Generated Code*  
Speichern des generierten Codes im Dateisystem  
(*Flat* → die editierte Komponente,  
*Recursive* → auch die referenzierten Komponenten).
- *View Generated Code*  
Generiert den Code für die Komponente und zeigt diesen in einem Texteditor an.  
Der Texteditor kann im ASCET-Optionsfenster, Knoten „ASCII Editor“ (s. Seite 61), gewählt werden.
- *File Out Generic Code For External Make*  
Speichern des generierten Codes in ein beliebiges Verzeichnis zur späteren Weiterverarbeitung durch separate Werkzeuge, z.B. in einem externen Make/Build-Prozess.
- *Exit*  
Beenden des Editors für Bedingungstabellen.
- **Element**  
(steht in der Fläche „Elements“ auch als Kontextmenü zur Verfügung)

### Hinweis

Die Funktionen **Add Item**, **Rename**, **Delete** und **Edit** stehen nur zur Verfügung, wenn ASCET-MD installiert ist.

- *View* → *Collapse all* klappt die Baumstruktur in der Fläche „Elements“ zu, sodass nur noch die Komponente selbst zu sehen ist.
- *View* → *Expand all* klappt die Baumstruktur vollständig auf, sodass der gesamte Inhalt der Komponente angezeigt wird.

- *Add Item*  
Aufnehmen einer Komponente als komplexes Element.
- *Rename (<F2>)*  
Umbenennen eines ausgewählten Elements.
- *Delete (<DEL>)*  
Löschen eines ausgewählten Elements.
- *Show Occurrences*  
Anzeigen aller Vorkommen eines Elements (nur im Blockdiagramm-  
editor).
- *Edit*  
Bearbeiten der Konfiguration eines ausgewählten Elements.
- *Edit Data*  
Bearbeiten der Daten eines ausgewählten Elements.
- *Edit Implementation*  
Bearbeiten der Implementierung eines ausgewählten Elements.
- *Set Cache Locking*  
Diese Menüfunktion ist nur für die Arbeit mit ASCET-RP von Bedeu-  
ung. Sie ist im ASCET-RP-Handbuch beschrieben.
- *Edit Component*  
Öffnen des Spezifikationseditors für eine ausgewählte aufgenom-  
mene Komponente.
- *Replace Component*  
Ersetzen einer Komponente durch eine andere Komponente. Der  
Name der alten Komponente bleibt erhalten.
- *Show Path*  
Zeigt den Pfad einer ausgewählten aufgenommenen Komponente.  
Zeigt den Pfad einer aufgenommenen Komponente.
- *Copy Path to Clipboard*  
Kopiert den Pfad einer aufgenommenen Komponente in die Zwi-  
schenablage  
(*Model Path* → Pfad im Modell,  
*Model asd:// link* → Pfad im ASCET-Protokollformat, der das Öff-  
nen/Referenzieren einer ASCET-Komponente im Modell als Hyper-  
link ermöglicht,  
*Database Path* → Pfad in der Datenbank,

*Dabase asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente in der Datenbank als Hyperlink ermöglicht).

- *Notes*

Öffnet den Notizeneditor für eine ausgewählte aufgenommene Komponente.

- *Edit Distribution / Edit Max Size*

Da Bedingungstabellen nur skalare Basiselemente enthalten können, sind diese Menüfunktionen deaktiviert.

- *Copy Elements* (<CTRL> + <C>)

Kopieren eines oder mehrerer ausgewählter Elemente aus der Liste „Elements“.

- *Paste Elements* (<CTRL> + <V>)

Einfügen eines oder mehrerer kopierter Elemente in die Liste „Elements“.

- *File Out Data / File In Data*

Da Bedingungstabellen nur skalare Basiselemente enthalten können, sind diese Menüfunktionen deaktiviert.

- *Export Data*

Exportieren des Datensatzes einer aufgenommenen Komponente.

- **Column**

(steht im Tabellenbereich auch als Kontextmenü zur Verfügung)

- *Add*

Fügt eine neue Spalte hinzu  
(*Condition* → im Bedingungsbereich,  
*Instruction* → im Aktionsbereich).

- *Delete*

Entfernt die ausgewählte Spalte (ohne das zugehörige Element zu löschen).

- *Move Left*

Verschiebt die ausgewählte Spalte nach links.

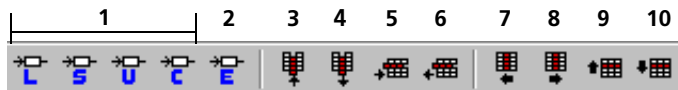
- *Move Right*

Verschiebt die ausgewählte Spalte nach rechts.

- **Row**  
(steht im Tabellenbereich auch als Kontextmenü zur Verfügung)
  - *Add*  
Fügt eine neue Zeile hinzu.
  - *Delete*  
Löscht die ausgewählte Zeile.
  - *Move Up*  
Verschiebt die ausgewählte Zeile nach oben.
  - *Move Down*  
Verschiebt die ausgewählte Zeile nach unten.
- **View**
  - *Show/Hide*  
Zeigt/verbirgt verschiedene Teile des Editors.  
(*Element List* → Liste „Elements“,  
*Diagram List* → Liste „Diagrams“)

### Beschreibung der Schaltflächen

---



1. Variablen (Logic, Signed Discrete, Unsigned Discrete, Continuous)
2. Enumeration
3. Add a Column (Spalte einfügen)
4. Delete a Column (Spalte entfernen)
5. Add a Row (Zeile einfügen)
6. Delete a Row (Zeile entfernen)
7. Move Column Left (Spalte nach links verschieben)
8. Move Column Right (Spalte nach rechts verschieben)
9. Move Row Up (Zeile nach oben verschieben)
10. Move Row Down (Zeile nach unten verschieben)





11. Generate Code (Code erzeugen)

12. Open Experiment for selected Experiment Target (generiert Code und öffnet das Offline-Experiment)

Nicht nummerierte Schaltflächen sind im Editor für Bedingungstabellen immer deaktiviert.

#### 4.7.1 Bedingungstabelle einrichten

Standardmäßig enthält die Bedingungstabelle keine Elemente. Bis auf die Spalte „methods“ ist die Tabelle leer. Sie müssen alle benötigten Elemente, Spalten und Zeilen hinzufügen. Erst wenn Sie mindestens ein Element angelegt haben, können Sie Spalten hinzufügen; Reihen können Sie erst hinzufügen, wenn im Bedingungsbereich mindestens eine Spalte existiert.

##### **Elemente hinzufügen:**

Bedingungstabellen können skalare Basiselemente, Enumerationen und Klassen als Elemente enthalten. Die Methode `trigger` kann Argumente, jedoch keinen Rückgabewert enthalten.

1. Basiselemente, Enumerationen



- Klicken Sie auf eine der Variablen-Schaltflächen oder auf **Enumeration**, um ein Basiselement oder eine Enumeration zu erzeugen.

In der Liste „Elements“ wird ein entsprechendes Element hinzugefügt.

- Geben Sie einen Namen für das Element ein und drücken Sie <ENTER>.

2. komplexe Elemente

- Fügen Sie eine Klasse wie auf Seite 229 beschrieben hinzu.

##### **Hinweis**

*Sie können keine speziellen Klassen wie Zustandsautomaten, CT-Blöcke, Boolesche Tabellen oder Bedingungstabellen hinzufügen.*

3. Argumente und methodenlokale Variablen der Methode `trigger`

- Wählen Sie in der Liste „Diagrams“ die Methode `trigger` aus.

- Wählen Sie aus dem Kontextmenü **Edit** oder

- doppelklicken Sie auf den Namen der Methode.  
Der Schnittstelleneditor wird geöffnet. Da `trigger` keinen Rückgabewert haben kann, fehlt das Register „Return“.
- Fügen Sie die gewünschten Argumente wie auf Seite 202 beschrieben hinzu.
- Fügen Sie die gewünschten methodenlokalen Variablen wie auf Seite 204 beschrieben hinzu.  
Methodenlokale Variablen sind nur innerhalb der Bedingungstabelle verfügbar.

Sie können die Daten und Implementierungen der Elemente wie gewohnt bearbeiten; näheres dazu finden Sie in den Kapiteln 4.11 „Bearbeiten von Daten“ und 4.12 „Bearbeiten von Implementierungen“.

Um die Elemente außerhalb der Bedingungstabelle zugänglich zu machen, müssen Sie die Get- oder Set-Ports der Elemente (sofern vorhanden) aktivieren oder die Elemente als global deklarieren. Andere Zugriffsmöglichkeiten auf die Elemente in einer Bedingungstabelle gibt es nicht.

#### **Elemente für externe Kommunikation zugänglich machen:**

---

- Wählen Sie in der Liste „Elements“ ein Element aus.
- Wählen Sie **Element** → **Edit**  
*oder*
- wählen Sie aus dem Kontextmenü **Edit**, um den Elementeditor zu öffnen.  
Abhängig von dem gewählten Element kann der Elementeditor unterschiedlich aussehen (siehe Kapitel 4.10.1). Das Feld „Scope“ und die Optionen **Set()** und **Get()** sind aber immer vorhanden.

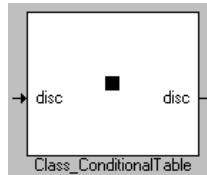
##### 1. Get-/Set-Ports verwenden

- Aktivieren Sie die Option **Set()**, um einen Eingang für das Element hinzuzufügen.  
Das Element kann nun von außerhalb der Bedingungstabelle beschrieben werden.

- Aktivieren Sie die Option **Get()**, um einen Ausgang für das Element hinzuzufügen.

Das Element kann nun von außerhalb der Bedingungstabelle gelesen werden.

Die so erzeugten Ein- und Ausgänge sind als Anschlüsse am Layout der Bedingungstabelle zu sehen.



## 2. globale Elemente verwenden

- Wählen Sie im Feld „Scope“ die Option **Imported**, wenn das Element in einer anderen Komponente/einem anderen Projekt definiert wurde und in der Bedingungstabelle verwendet werden soll.
- Wählen Sie **Exported**, wenn Sie das Element in der Bedingungstabelle definieren und in anderen Komponenten/Projekten verwenden wollen.

Wenn Sie ein Element hinzugefügt haben, werden die Menüfunktionen zum Erzeugen von Spalten für Bedingungen und Aktionen aktiviert.

### Hinweis

Die Schaltfläche **Add a Column** kann grundsätzlich Spalten im Bedingungsbereich und im Aktionsbereich anlegen. Wählen Sie eine existierende Spalte, um so den Bereich der neuen Spalte festzulegen; dadurch wird die Schaltfläche aktiviert.

Jeder Spalte wird bei der Erzeugung eine Variable, eine Enumeration oder ein Argument zugewiesen. Komplexe Elemente können den Spalten nicht zugewiesen werden, sie werden nur bei der Spezifikation von Bedingungen oder Aktionen verwendet.

## Spalten im Bedingungsbereich anlegen:

---

Sie können im Bedingungsbereich bis zu 100 Spalten anlegen.

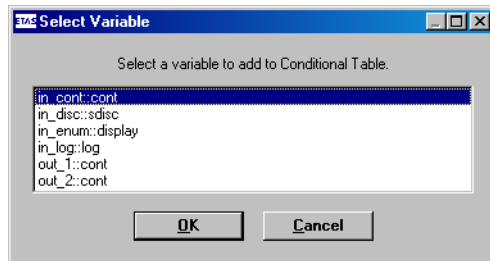
- Wählen Sie **Column** → **Add** → **Condition**.  
Diese Funktion ist über die Menüzeile und das Kontextmenü des Tabellenbereichs erreichbar.

Oder

- Markieren Sie die Spalte, neben der Sie die neue Spalte einfügen wollen.
- Klicken Sie auf **Insert a Column**.



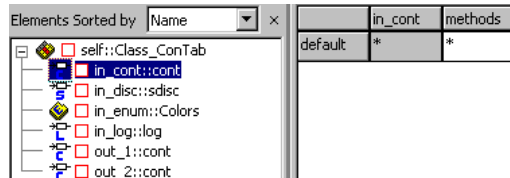
Das Fenster „Select Variable“ öffnet sich. Es enthält eine Liste aller vorhandenen Variablen, die noch keiner Spalte des Bedingungsbereichs zugewiesen wurden.



- Wählen Sie eine Variable aus der Liste.  
Sie können jede Variable genau einmal einer Spalte des Bedingungsbereichs zuweisen.

- Klicken Sie auf **OK**, um das Fenster zu schließen.

Im Bedingungsbereich wird eine Spalte mit dem Namen der ausgewählten Variablen erzeugt. Wenn die erzeugte Spalte die erste der gesamten Tabelle ist, wird gleichzeitig die Zeile „Default“ erzeugt.



Die neue Spalte wird am rechten Rand des Bedingungsbereichs angelegt, wenn sie mit der Menüfunktion **Column → Add → Condition** erzeugt wurde. Wenn Sie die Schaltfläche **Insert a Column** verwendet haben, wird die neue Spalte rechts neben der ausgewählten Spalte erzeugt.

### Spalten im Aktionsbereich anlegen:

Sie können im Aktionsbereich bis zu 100 Spalten anlegen.

- Wählen Sie aus der Menüzeile oder dem Kontextmenü **Column → Add → Instruction**.

Oder



- Markieren Sie die Spalte, neben der Sie die neue Spalte einfügen wollen.
- Klicken Sie auf **Insert a Column**.

Das Fenster „Select Variable“ öffnet sich. Es enthält eine Liste aller vorhandenen Variablen, die noch keiner Spalte des Aktionsbereichs zugewiesen wurden.

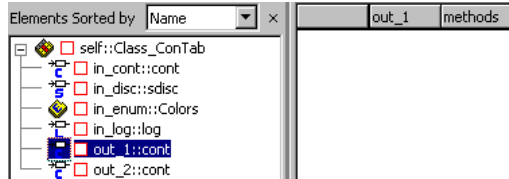
- Wählen Sie eine Variable aus der Liste.  
Sie können jede Variable genau einmal einer Spalte des Aktionsbereichs zuweisen.

### Hinweis

Argumente der Methode `trigger` können im Aktionsbereich **nicht** zugewiesen werden.

- Klicken Sie auf **OK**, um das Fenster zu schließen.

Im Aktionsbereich wird eine Spalte mit dem Namen der ausgewählten Variablen erzeugt. Selbst wenn die erzeugte Spalte die erste der gesamten Tabelle ist, wird *keine* Zeile erzeugt.



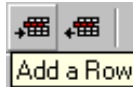
Die neue Spalte wird links neben der Spalte „methods“ angelegt, wenn sie mit der Menüfunktion **Column → Add → Instruction** erzeugt wurde. Wenn Sie die Schaltfläche **Insert a Column** verwendet haben, wird die neue Spalte rechts neben der ausgewählten Spalte erzeugt.

### Zeilen anlegen:

Wenn mindestens eine Spalte im Bedingungsbereich angelegt wurde, können Sie Zeilen hinzufügen. Eine Bedingungstabelle kann bis zu 100 Zeilen enthalten; da die Zeile „default“ vorgegeben ist, können Sie 99 Zeilen hinzufügen.

- Wählen Sie die Zeile aus, nach der Sie die neue Zeile einfügen wollen.
- Wählen Sie aus der Menüzeile oder dem Kontextmenü **Row → Add**

oder



- klicken Sie auf **Add a Row**.

Die neue Zeile wird nach der ausgewählten eingefügt. Wenn Sie keine Zeile oder „default“ ausgewählt haben, wird die neue Zeile unmittelbar vor der Zeile „default“ eingefügt.

Die Elemente der Zeile enthalten zunächst alle ein Sternchen. Wie Sie die Bedingungen und Aktionen spezifizieren, ist in Kapitel 4.7.2 beschrieben.

	in_cont	in_disc	out_1	methods
1.	*	*	*	*
default	*	*	*	*

### Spalten verschieben:

---

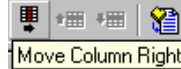
Sie können die Spalten verschieben, jedoch nur innerhalb des Bereichs.

- Markieren Sie die Spalte, die Sie verschieben wollen.
- Wählen Sie aus der Menüzeile oder dem Kontextmenü **Column** → **Move Left** oder **Column** → **Move Right**



oder

- klicken Sie auf **Move Column Left** oder **Move Column Right**.



Die Spalte wird um eine Position nach links oder rechts verschoben.

Die erste Spalte eines Bereichs kann nicht nach links verschoben werden, die letzte Spalte eines Bereichs kann nicht nach rechts verschoben werden. Die Spalte „methods“ kann nicht verschoben werden; sie ist immer die letzte Spalte des Aktionsbereichs.

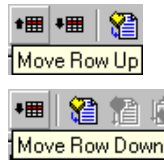
### Zeilen verschieben:

---

Je weiter oben in der Tabelle eine Zeile steht, desto höher ist ihre Priorität. Sie können die Zeilen verschieben und so die Priorität der darin spezifizierten Bedingung beeinflussen.

- Markieren Sie die Zeile, die Sie verschieben wollen.

- Wählen Sie aus der Menüzeile oder dem Kontextmenü **Row** → **Move Up** oder **Row** → **Move Down**



oder

- klicken Sie auf **Move Row Up** oder **Move Row Down**.

Die Zeile wird um eine Position nach oben oder unten verschoben.

Die erste Zeile kann nicht nach oben verschoben werden, die letzte nummerierte Zeile kann nicht nach unten verschoben werden.

Die Zeile „default“ kann nicht verschoben werden; sie ist immer letzte Zeile der Tabelle.

Sie können Spalten, Zeilen und Elemente wieder aus der Tabelle entfernen.

### Spalten entfernen:

---

- Wählen Sie die Spalte, die Sie löschen wollen.
- Wählen Sie aus der Menüzeile oder dem Kontextmenü **Column** → **Delete**

oder

- klicken Sie auf **Delete a Column**.

Die ausgewählte Spalte wird entfernt. Das zugewiesene Element wird jedoch nicht gelöscht, es ist weiterhin in der Liste „Elements“ vorhanden und kann neu zugewiesen werden.



### Zeilen löschen:

---

- Wählen Sie die Zeile, die Sie löschen wollen.
- Wählen Sie aus der Menüzeile oder dem Kontextmenü **Row** → **Delete**

oder

- klicken Sie auf **Delete a Row**.

Die ausgewählte Zeile wird entfernt.



### Elemente löschen:

---

- Markieren Sie in der Liste „Elements“ das Element, das Sie löschen wollen.



- Wählen Sie **Element** → **Delete**  
oder
- wählen Sie aus dem Kontextmenü **Delete**  
oder
- drücken Sie die Taste <DEL>.  
Ein Bestätigungsfenster öffnet sich.
- Bestätigen Sie den Löschvorgang mit **OK**.  
Das Element wird aus der Liste entfernt. War es einer Spalte zugewiesen, wird die Spalte ebenfalls gelöscht.

#### Hinweis

Wenn das gelöschte Element Teil der Spezifikation einer Bedingung oder Aktion ist, wird diese Spezifikation **nicht** automatisch gelöscht. Sie müssen notwendige Änderungen manuell vornehmen.

### 4.7.2 Bedingungstabelle spezifizieren

Die Zellen der neu angelegten Spalten und Zeilen enthalten zunächst alle ein Sternchen (\*). Sie müssen die tatsächlichen Bedingungen und Aktionen manuell eingeben.

**Bedingungen:** Die Zellen im Bedingungsbereich können Bedingungen der folgenden Form enthalten:

`== b, <= b, >= b, < b, > b, != b, *`

- `b` ist entweder ein Wert oder ein Element oder der Name einer öffentlichen Methode einer referenzierten Klasse. Auch daraus zusammengesetzte komplexe Ausdrücke in ESDL-Syntax sind möglich.
- Der Stern bedeutet, dass ein beliebiger Wert möglich ist; er wird dann verwendet, wenn das Element der Spalte für die aktuelle Zeile keine Bedeutung hat.
- In einer Zelle können mehrere durch Zeilenumbruch getrennte Bedingungen angegeben werden. Diese werden als UND-verknüpft interpretiert.

	cont
1.	> -1.0 < 0.0

#### Hinweis

Die Zellen im Bedingungsbereich der Zeile „defaults“ können **nicht** bearbeitet werden. Sie enthalten immer den Stern.

**Aktionen:** Die Zellen im Aktionsbereich können Zuweisungen der folgenden Form enthalten:

- = `b, Klasse.Methode(<Größen>), *`
  - `b` ist ein Wert, ein Element oder der Name einer öffentlichen Methode einer referenzierten Klasse. Auch daraus zusammengesetzte komplexe Ausdrücke in ESDL-Syntax sind möglich.
  - `Klasse.Methode(<Größen>)` steht für einen Methodenaufwurf; diese Form der Aktion ist ausschließlich in der Spalte „methods“ erlaubt. Alle öffentlichen Methoden der referenzierten Klassen können aufgerufen werden.
  - Der Stern bedeutet, dass in der betreffenden Spalte keine Aktion (Wertzuweisung oder Methodenaufwurf) definiert ist.

Sie können für die Spezifikation von Bedingungen und Aktionen nur Elemente oder Klassen verwenden, die in der Liste „Elements“ enthalten sind. Andernfalls wird bei der Codegenerierung eine Fehlermeldung erzeugt.

### Hinweis

*Es gibt während der Spezifikation von Bedingungen und Aktionen keine Überprüfung der Syntax und Semantik. Achten Sie daher unbedingt auf korrekte Eingaben, insbesondere von Methodennamen. Fehlermeldungen werden erst zum Zeitpunkt der Codegenerierung erzeugt.*

### Bedingung oder Aktion eingeben:

	in_cont	in_disc
1.	<input type="text"/>	*
default	*	*

- Doppelklicken Sie in die Zelle, die Sie bearbeiten wollen.

Die Zelle wird zum Eingabefeld.

- Geben Sie die Bedingung oder die Aktion ein. Bedingungen und Aktionen werden als reine Zeichenketten eingegeben.

### Hinweis

*Es wird nicht überprüft, ob verschiedene Zeilen im Bedingungsbereich identische Bedingungen enthalten.*

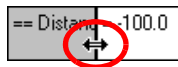
Eine Zelle im Aktionsbereich kann ebenfalls mehrere Zeilen enthalten. Jede Zeile wird als eigene Zuweisung (siehe Abbildung) bzw. Methodenaufruf (Spalte „methods“) betrachtet.

out	methods
= -100.0	ClassABC.doThat()
= out + cont	ClassXYZ.doThis(out,cont)
= MathFcn.cos(out)	

- Klicken Sie irgendwo in den Tabellenbereich, um die Eingabe zu übernehmen.

### Spaltenbreite verändern:

Wenn die Breite einer Spalte zu klein für die in den Zellen enthaltenen Aktionen oder Bedingungen ist, können Sie sie verbreitern.



- Bewegen Sie den Mauszeiger auf die rechte senkrechte Begrenzung einer Spalte. Der Mauszeiger verwandelt sich in einen Doppelpfeil.
- Ziehen Sie die Spaltenbegrenzung nach links oder rechts, um die Spaltenbreite zu ändern. Wenn Sie den Editor verlassen, wird die Spaltenbreite auf den Standardwert zurückgesetzt.

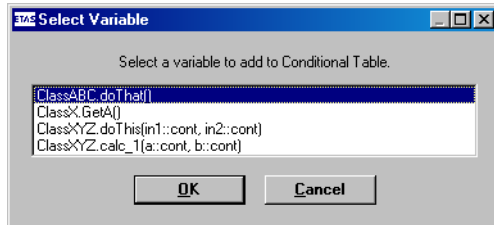
### Methoden in Bedingungen/Aktionen verwenden:

Zur Vermeidung von Fehlern bei der Eingabe von Methodennamen gibt es ein Hilfsmittel für die Eingabe.

- Markieren Sie die Zelle, in der Sie eine Methode verwenden wollen.

- Klicken Sie mit der rechten Maustaste in den Tabellenbereich und wählen Sie aus dem Kontextmenü **Insert Method**.

Das Fenster „Select Variable“ öffnet sich. Es enthält eine Liste aller öffentlichen Methoden aller referenzierten Klassen.



- Wählen Sie die Methode aus, die Sie verwenden wollen.
- Klicken Sie auf **OK**.

Der Name der Methode wird in einer eigenen Zeile am Ende der Zelle eingefügt. Wenn die Methode Argumente hat, werden die in der Klasse definierten Namen eingetragen.

3.	> 0.0 < 1.0 ClassX.GetA()	== false	= -100.0	*	ClassX.GetA()	ClassABC.doThat() ClassXYZ.doThis(in1::cont, in2::cont)
----	---------------------------------	----------	----------	---	---------------	--

- Sie müssen den eingefügten Methodenaufruf nachbearbeiten.
  - Tragen Sie anstelle der vorgegebenen Argumentnamen existierende Variablenamen aus Ihrer Bedingungsstabelle ein.
  - Fügen Sie im Bedingungsereich einen Vergleichsoperator (s. Seite 393) ein.

- Fügen Sie im Aktionsbereich das Zuweisungszeichen = ein (ausgenommen Spalte „methods“) und entfernen Sie ggf. den Stern.

3.	> 0,0 < 1,0 != ClassX.GetA()	== false	= -100.0	= ClassX.GetA()	ClassABC.doThat() ClassXYZ.doThis(out,cont)
----	------------------------------------	----------	----------	-----------------	--

### 4.7.3 Experimentieren mit Bedingunstabellen

Wenn Sie die Tabelle vollständig spezifiziert haben, können Sie wie für andere Komponenten Code generieren und Experimente durchführen (siehe Kapitel 4.1.10 „Analysieren von Komponenten“). Lediglich die für Blockdiagramme und ESDL-Komponenten verfügbare Funktion **Diagram → Analyze Diagram** ist für Bedingunstabellen nicht verfügbar.

Folgende Problemfälle können bei der Codegenerierung auftreten:

- **Fehlerhafte Eingaben** bei der Spezifikation der Bedingungen und Aktionen führen zu Fehlermeldungen.
- Wenn Sie in verschiedenen Zeilen des Bedingungsereichs **identische Bedingungen** spezifiziert haben, werden bei der Codegenerierung zwar alle entsprechenden `If...Then...`-Zeilen generiert, aber nur die erste wird bei der Auswertung der Tabelle auch berücksichtigt. Es wird *keine Warnung* angezeigt, da keine inhaltliche Überprüfung der Tabelle stattfindet.
- Wenn Sie in einer Enumeration, die von der Bedingunstabelle verwendet wird, **Enumeratoren löschen, hinzufügen oder verschieben**, kann das zu Inkonsistenzen in der Bedingunstabelle führen. Bei der Codegenerierung wird ggf. eine entsprechende Fehlermeldung angezeigt.
- Wenn Sie eine **Enumeration**, die von der Bedingunstabelle verwendet wird, aus der **Datenbank löschen**, wird die Enumeration bei der Codegenerierung als undefiniertes Element behandelt; eine entsprechende Fehlermeldung wird angezeigt.

Die Bedingunstabelle kann nur über die Methode `trigger` aufgerufen werden. Bei jedem Aufruf der Tabelle werden folgende Schritte durchgeführt:

1. Die Methodenargumente der Methode `trigger` werden prozessiert und die Werte der Elemente zugewiesen.

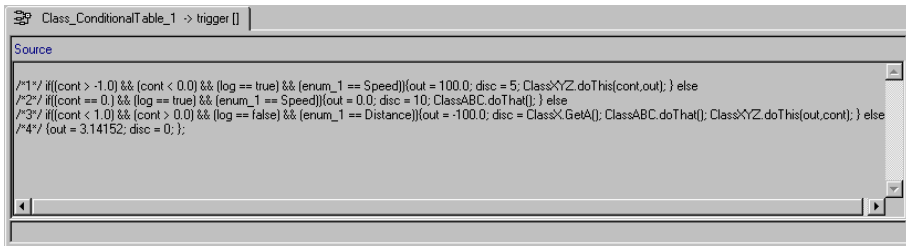
2. Eine passende Bedingung wird gesucht. Sobald eine passende Bedingung gefunden ist, wird die Suche abgebrochen, auch wenn weiter unten andere ebenfalls passende Bedingungen spezifiziert sind.

Wenn keine Bedingung passt, wird die Standardbedingung (Zeile „default“) ausgewählt.

3. Die zur gefundenen Bedingung gehörenden Aktionen werden ausgeführt.

Da `trigger` keinen Rückgabewert haben kann, müssen die Ergebnisse der Aktionen über Get-Ports (siehe Seite 386) weitergegeben werden. Um die Datenkonsistenz zu jedem Zeitpunkt sicherzustellen, werden globale Variable zu diesem Zweck nicht empfohlen.

Beim Offline-Experiment wird wie bei allen ESDL-Komponenten im Fenster „Physical Experiment“ der Code angezeigt. Das ist übrigens die einzige Stelle, an der der Anwender den ESDL-Code sieht.



```
Class_ConditionalTable_1 -> trigger []
Source
/*1*/ if((cont > -1.0) && (cont < 0.0) && (log == true) && (enum_1 == Speed)){out = 100.0; disc = 5; ClassXYZ.doThis(cont,out); } else
/*2*/ if((cont == 0.) && (log == true) && (enum_1 == Speed)){out = 0.0; disc = 10; ClassABC.doThat(); } else
/*3*/ if((cont < 1.0) && (cont > 0.0) && (log == false) && (enum_1 == Distance)){out = -100.0; disc = ClassX.GetA(); ClassABC.doThat(); ClassXYZ.doThis(out,cont); } else
/*4*/ {out = 3.14152; disc = 0; ;}
```

Sie können das Experiment wie in Kapitel 6.1 „Die Experimentierumgebung“ beschrieben einrichten und durchführen.

## 4.8 Der Projekteditor

Ein Projekt in ASCET definiert die Funktionalität eines eingebetteten Softwaresystems und dient als Grundlage für die Generierung des Codes für das eingebettete System. Eine solche Spezifikation kann auf verschiedenen experimentellen Targets durchgeführt werden und ist echtzeitfähig. Der Festkomma-Code für Mikrocontroller kann generiert werden, sobald eine Implementierungstransformation für das eingebettete Softwaresystem definiert wurde. Dieser Festkomma-Code kann dann auf den Targets des Mikrocontrollers in Fullpass- oder Bypass-Experimenten laufen.

Alternativ kann ein Projekt zum Modellieren eines Steuerkreises verwendet werden, der aus einer Kombination von CT-Blöcken und Modulen besteht. In diesem Fall werden die CT-Blöcke verwendet, um das Streckenmodell zu

beschreiben, und die Module dienen der Beschreibung des Controllers. Mit einem solchen Modell kann dann auf einem experimentellen Target experimentiert werden.

Ein Projekt wird in Modulen entwickelt, wobei zuerst die einzelnen Komponenten des Systems entwickelt und getestet und diese danach in einem Projekt kombiniert werden. Das Entwickeln eines Projekts verläuft in folgenden Schritten:

- Auswählen der Module, aus denen sich das System zusammensetzt. Die Definitionen der Komponenten sind referenziert, d.h. wenn die Definition einer Komponente verändert wird, wirkt sich diese Änderung direkt auf alle Projekte aus, die diese Komponente verwenden.
- Definieren des gesamten Steuerflusses des eingebetteten Softwaresystems durch Konfigurieren des Echtzeit-Betriebssystems.
- Festlegen der Implementierungstransformation zur Generierung des Codes mit Festkomma-Arithmetik. Um diese Implementierung zu definieren, muss die Transformationsformel in das Projekt einbezogen werden.

Wie Komponenten können Projekte mehrere Datensätze und Implementierungen enthalten. Deshalb ist beim Generieren des ausführbaren Codes aus der formalen Definition die geeignete Variante auszuwählen. Die aktuelle Variante eines Projekts wird durch Auswählen des Datensatzes (dies ist auch möglich, wenn nur mit Komponenten experimentiert wird) sowie durch Auswählen der Implementierungstransformation festgelegt.

Das Definieren einer Implementierung ist nur für die Generierung des Codes mit Festkomma-Arithmetik erforderlich. Wird nur eine Gleitkomma-Arithmetik benötigt (z.B. in einer Bypass-Umgebung), muss die Implementierungstransformation nicht definiert werden.

Die Codegenerierung kann für ein spezifisches Projekt durch Auswählen der Codevariante und der Zielplattform, für die der Code generiert werden soll, angepasst werden. Der Code kann für drei Arithmetiktypen generiert werden: Gleitkomma-Arithmetik, Festkomma-Arithmetik und quantisierte Gleitkomma-Arithmetik. Bei letzterer handelt es sich um eine Simulation der Festkomma-Arithmetik auf der Grundlage der Gleitkomma-Arithmetik, bei der die Auswirkungen der Quantisierung untersucht werden können, wobei die Quantisierung und Grenzen der Werte interaktiv bei der Ausführung des Codes geändert werden können.

### **Erstellen eines neuen Projekts:**

---

- Wählen Sie im Komponentenmanager den Ordner aus, in dem Sie ein neues Projekt anlegen wollen.

- Wählen Sie die Menüfunktion **Insert** → **Project**

oder



- klicken Sie auf die Schaltfläche **Insert Project**.
- Geben Sie den Namen des neuen Projekts ein und drücken Sie <ENTER>.

Ein neues Projekt wird erzeugt.

- Wählen Sie in der Menüzeile **Component** → **Edit Item**

oder

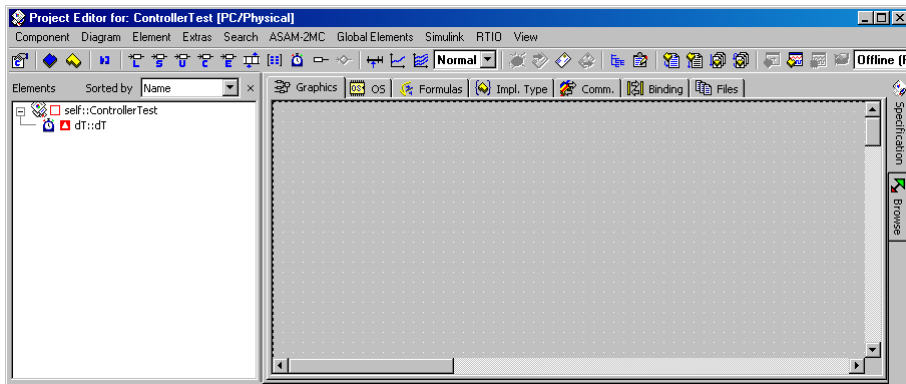
- drücken Sie <Enter>

oder

- doppelklicken Sie in der Liste „1 Database“ auf das Projekt.

Der Projekteditor wird geöffnet.

Wie Komponenten werden Projekte in der Datenbank gespeichert, mit Ausnahme der aus den Spezifikationseditoren generierten Testprojekte. Einzelne Projekte werden im Projekteditor zusammengesetzt und eingerichtet, und im Projekteditor wird mit ihnen experimentiert. Alle nachfolgend beschriebenen Operationen können über den Projekteditor durchgeführt werden.



Die Fläche „Elements“ liefert eine baumartige Übersicht über die Komponenten, aus denen das Projekt besteht. Diese Liste enthält die Module, die durch das Projekt referenziert werden, sowie die durch sie referenzierten Komponenten.



ten. Sie kann auf- und zugeklappt werden. Die Elemente eines Projekts, die zur Kommunikation zwischen den Modulen und Komponenten verwendet werden, sind im Kontext des Projekts global.

#### 4.8.1 Standardprojekt für eine Komponente

Aus jeder mit einem Spezifikationseditor erstellten Komponente kann ein eigenes Test- oder Standardprojekt erstellt werden. Allerdings erscheint das Standardprojekt nicht im Komponentenmanager, da es direkt aus dem Spezifikationseditor heraus erstellt wird. Das Einrichten des Standardprojekts findet gleichfalls im Projekteditor statt und schließt auch das Experimentieren mit dem Standardprojekt ein. Die Funktionalitäten und Operationen, die im Projekteditor durchgeführt werden können, werden in den kommenden Kapiteln und Abschnitten beschrieben.

##### **Bearbeiten eines Standardprojekts:**

---

- Wählen Sie im Spezifikationseditor **Component → Default Project → Edit**.  
Der Projekteditor für das Standardprojekt wird geöffnet.
- Wählen Sie im Spezifikationseditor **Component → Default Project → Resolve Globals**.  
Für jedes importierte Element in der Komponente, für das kein exportiertes Element vorhanden ist, wird ein globales Element erzeugt (siehe auch „Definieren der globalen Kommunikation“ auf Seite 416). Der Projekteditor wird nicht geöffnet.
- Wählen Sie im Spezifikationseditor **Component → Default Project → Delete Unused Globals**.  
Nicht genutzte globale Elemente werden aus dem Standardprojekt der Komponente entfernt.

Die Offline-Experimentierumgebung, die für Komponenten zur Verfügung steht, wird in das Offline-Experimentieren für das Standardprojekt eingebettet. Neben der Offline-Experimentierumgebung bietet ASCET eine Online-Experimentierumgebung für Projekte. In der Online-Experimentierumgebung werden die Projekte in Echtzeit mit dem im Echtzeit-Betriebssystem definierten Verhalten ausgeführt.

- **Component**
  - *Clean code generation directory*  
Löscht alle Dateien im Codegenerierungsverzeichnis.
  - *Touch*  
Erzwingen einer Neugenerierung bei der nächsten Codegenerierung  
(*Flat* → die editierte Komponente,  
*Recursive* → auch die referenzierten Komponenten).
  - *Generate Code* (<CTRL> + <F7>)  
Der Code für das Projekt wird generiert.
  - *Compile*  
Kompiliert den generierten Code.
  - *Link Only*  
Verknüpft die Objektdateien des Projekts zu einer ausführbaren HEX-Datei.
  - *Build* (<F7>)  
Generiert, speichert und baut ausführbaren Code für das gewählte Target. Die Ergebnisse werden in der Datenbank gespeichert.
  - *Build from directory* (<ALT> + <F7>)  
Generiert, speichert und baut ausführbaren Code für das gewählte Target. Die Ergebnisse werden in ein Verzeichnis geschrieben, aber nicht in der Datenbank gespeichert.
  - *Rebuild All* (<SHIFT> + <F7>)  
Führt eine komplette Neugenerierung des gesamten Projekts und aller zugehöriger Komponenten aus. Dieser Befehl hat dieselbe Wirkung wie **Component** → **Touch** → **Recursive** gefolgt von **Component** → **Build**.
  - *Transfer Project*  
Transferiert das Projekt zu dem ausgewählten Experiment.  
Die Schaltfläche ist nur aktiviert, wenn in den Build-Optionen das Target `ES1130`, `ES1135` oder `Prototyping` und im Kombikästchen „Experiment Target“ der Eintrag `INCA` oder `INTECRIO` gewählt wurde.

- *Open Experiment*  
Ein Experiment wird gestartet.
- *Flash Target*  
Der durch ASCET generierte Code wird anstatt in den RAM in den Flash-Speicher des experimentellen Targets geschrieben. Ein Startprogramm zum Hochfahren aus dem Flash-Speicher ist im Code integriert. Die Hardware des Targets führt dann das ASCET-Modell nach jedem Reset aus.
- *Reconnect to Experiment*  
Stellt die Verbindung zum auf dem ausgewählten Target laufenden Experiment wieder her.
- *Edit Data*  
Der Dateneditor für das Projekt wird geöffnet. Das Durchsuchen von Daten im Projekt beinhalteteter Komponenten ist möglich.
- *Edit Implementation*  
Der Implementierungseditor für das Projekt wird geöffnet. Das Durchsuchen von Implementierungen im Projekt beinhalteteter Komponente ist möglich.
- *Edit Notes*  
Öffnet den Notizeneditor – Hier können Notizen zum Projekt gemacht werden.
- *Edit Project Properties*  
Öffnet das Fenster „Project Properties“, in dem die Projekteinstellungen gesetzt werden (siehe „Projekteinstellungen“ auf Seite 438).
- *Check Dependency*  
Es wird überprüft ob die Zuweisung der formalen Parameter zu den Modellparametern korrekt ist.
- *Export Data*  
Exportiert den Datensatz des Projekts.
- *Show Path*  
Zeigt den Pfad einer aufgenommenen Komponente.
- *Copy Path to Clipboard*  
Kopiert den Pfad einer aufgenommenen Komponente in die Zwischenablage  
(*Model Path* → Pfad im Modell,

*Model asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente im Modell als Hyperlink ermöglicht,

*Database Path* → Pfad in der Datenbank,

*Dabase asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente in der Datenbank als Hyperlink ermöglicht).

– *File Out Generated Code*

Speichern des generierten Code im Dateisystem

(*Flat* → die editierte Komponente,

*Recursive* → auch die referenzierten Komponenten).

– *View Generated Code*

Generiert den Code für die Komponente und zeigt diesen in einem Texteditor an.

Der Texteditor kann im ASCET-Optionsfenster, Knoten „ASCII Editor“ (s. Seite 61), gewählt werden.

– *File Out Generic Code For External Make*

Speichern des generierten Code in ein beliebiges Verzeichnis zur späteren Weiterverarbeitung durch separate Werkzeuge z.B. in einem externen Make/Build - Prozess.

– *Exit*

Beenden des Projekteditors.

• **Diagram**

– *Store To Cache*

Die Spezifikation einer Komponente wird im Cache-Speicher gespeichert. (Es erfolgt keine permanente Speicherung in der Datenbank.)

– *Load Diagram*

Laden eines Diagramms.

• **Element**

– *View* → *Collapse all* klappt die Baumstruktur in der Fläche „Elements“ zu, sodass nur noch die Komponente selbst zu sehen ist.

– *View* → *Expand all* klappt die Baumstruktur vollständig auf, sodass der gesamte Inhalt der Komponente angezeigt wird.

– *Add Item*

Aufnehmen einer Komponente in das Projekt.

- *Rename* (<F2>)
 

Umbenennen einer Komponente.
- *Delete* (<DEL>)
 

Löschen einer Instanz einer Komponente.
- *Show Occurences*

Anzeigen aller Vorkommen eines Elements (nur im Blockdiagramm-editor).
- *Edit*

Bearbeiten der Konfiguration eines Elements.
- *Edit Data*

Bearbeiten der Daten eines Elements.
- *Edit Implementation*

Bearbeiten der Implementierung eines Elements.
- *Set Cache Locking*

Diese Menüfunktion ist nur für die Arbeit mit ASCET-RP von Bedeutung. Sie ist im ASCET-RP-Handbuch beschrieben.
- *Edit Component*

Öffnen des Spezifikationseditors für eine aufgenommenen Komponente.
- *Replace Component*

Ersetzen einer Komponente durch eine andere Komponente. Der Name der alten Komponente bleibt erhalten.
- *Show Path*

Zeigt den Pfad einer aufgenommenen Komponente.
- *Copy Path to Clipboard*

Kopiert den Pfad einer aufgenommenen Komponente in die Zwischenablage  
 (*Model Path* → Pfad im Modell,  
*Model asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente im Modell als Hyperlink ermöglicht,  
*Database Path* → Pfad in der Datenbank,  
*Dabase asd:// link* → Pfad im ASCET-Protokollformat, der das Öffnen/Referenzieren einer ASCET-Komponente in der Datenbank als Hyperlink ermöglicht).

- *Notes*  
Öffnet den Notizeneditor für eine aufgenommene Komponente.
- *Edit Distribution*  
Öffnet einen Editor für die Distribution (nur bei Gruppentabellen).
- *Edit Max Size*  
Öffnet einen Editor, in dem die maximale Größe eines Arrays einer beliebigen Tabelle oder Matrix vorgegeben werden kann.
- *Copy Elements* (<CTRL> + <C>)  
Kopieren eines Elements aus der Liste „Elements“.
- *Paste Elements* (<CTRL> + <V>)  
Einfügen eines kopierten Elements in die Liste „Elements“.
- *File Out Data*  
Schreiben der Daten eines Array oder einer Tabelle in eine Datei.
- *File In Data*  
Lesen der Daten eines Array oder einer Tabelle aus einer Datei.
- *Export Data*  
Exportieren des Datensatz einer aufgenommenen Komponente
- **Extras**
  - *Copy C-Code From*  
Kopiert C-Code von einem anderen Target.
  - *Global Replace Formula*  
Ersetzt eine Formel rekursiv.
  - *Update Implementations*  
Aktualisiert alle Implementierungen des Projekts.
  - *Select Hardware*  
Diese Menüfunktion ist nur verfügbar, wenn Sie ASCET-RP installiert haben.
- **Search**  
Abgesehen von der eingeschränkten Reichweite der Suche sind die Suchkriterien dieselben wie in „Durchsuchen einer Datenbank“ auf Seite 139.
  - *Components*  
Durchsucht das Projekt nach einer Komponente.

- *References to component*  
Durchsucht das Projekt nach Referenzen auf die angegebene Komponente.
- *Declarations of method*  
Durchsucht das Projekt nach der definierenden Komponente der angegebenen Methode/des Prozesses.
- *References to method*  
Durchsucht das Projekt nach einer Methode oder einem Prozess.
- *Declarations of element*  
Durchsucht das Projekt nach der definierenden Komponente des angegebenen Elements.
- *References to element*  
Durchsucht das Projekt nach Komponenten, die das angegebene Element verwenden.
- *Senders of message*  
Durchsucht das Projekt nach Modulen, die die angegebene Message senden.
- *Receivers of message*  
Durchsucht das Projekt nach Modulen, die die angegebene Message empfangen.
- *Find/Replace ESDL or C code*  
Suchen/Ersetzen einer Zeichenkette in aufgenommenen C-Code- oder ESDL-Komponenten.
- **ASAM-2MC**
  - *Write*  
Erzeugen von Applikationsdateien
  - *Read Hex. File*  
Einlesen von Applikationsdateien
- **Global Elements**
  - *Resolve Globals*  
Dieser Mechanismus erzeugt für jedes importierte Element in einem Modul, für das kein exportiertes Element vorhanden ist, automatisch ein globales Element.

- *Delete Unused Globals*

Löschen von nicht genutzten globalen Elementen.

Folgende Menüpunkte sind nur bei der Wahl einzelner Register verfügbar.

*Register „Graphics“*

---

- **View**

- *Show/Hide* → *Element List*

Zeigt/verbirgt die Liste „Elements“

- *Redraw*

Zeichnet das Diagramm neu.

- *Rebuild Connections*

Glättet fragmentierte Verbindungen, die bei der automatischen Konvertierung von Operatorimplementierungen (s. Seite 549) entstanden sind.

- *Undo* (<CTRL> + <Z>)

Rückgängigmachen der letzten Aktion.

- *Redo* (<CTRL> + <Y>)

Erneute Ausführung einer rückgängig gemachten Aktion.

- *Show Hierarchy Path*

Die oberste Hierarchie des Pfades wird angezeigt.

- *Parent Component*

Öffnet einen Editor für die aufnehmende Komponente. (Dieser Punkt ist nur verfügbar, wenn eine aufgenommene Komponente editiert wird.)

- *Parent Hierarchy Level*

Der Hierarchiepfad wird angezeigt.

- *Page frame Portrait*

Darstellung des Diagramms in Hochformat.

- *Page frame Landscape*

Darstellung des Diagramms in Querformat.

- *Grid*

Einstellung des Rasters im Zeichenbereich.

- *Print Diagram*

Der Inhalt des Registers „Graphics“ wird gedruckt (siehe Seite 283).



- *Save as Postscript/ Bitmap/ RTF/ GIF*  
Speichert das Diagramm im angegebenen Format.

### Register „OS“

---

- **Operating System**

- *Copy From Target*  
Konvertierung des Projektes von einem ausgewählten Target auf das aktuelle Target.
- *Copy To Target*  
Konvertierung des Projektes auf ein ausgewähltes Target.

- **Application Mode**

#### Hinweis

*Die Menüfunktionen sind auch als Kontextmenü im Feld „Application Modes“ verfügbar.*

- *Add*  
Anlegen eines neuen Betriebsmodus.
- *Rename*  
Umbenennen des Betriebsmodusnamen.
- *Delete*  
Löschen eines Betriebsmodus.
- *As Start Mode*  
Betriebsmodus als Startmodus definieren.
- *As Default CT-Mode*  
Setzt den ausgewählten Betriebsmodus als Standardmodus für CT-Blöcke.
- *Assign*  
Zuweisen eines Betriebsmodus zu einer Task.
- *Show in Tasks*  
Wählt alle Tasks, denen der gewählte Betriebsmodus zugewiesen ist.
- *Show Init Tasks*  
Wählt alle Initialisierungs-Tasks, denen der gewählte Betriebsmodus zugewiesen ist.

- **Process**

**Hinweis**

*Die Menüfunktionen sind auch als Kontextmenü im Feld „Processes“ verfügbar.*

- *Assign*  
Zuweisen eines Prozesses zu einer Task.
- *Show in Tasks*  
Anzeigen aller Vorkommen eines Prozesses in der Fläche „Tasks“.

- **Task**

**Hinweis**

*Die Menüfunktionen sind auch als Kontextmenü im Feld „Tasks“ verfügbar.*

- *Add*  
Erstellen einer Task.
- *Rename*  
Umbenennen einer Task.
- *Delete*  
Löschen einer Task.
- *Move Up*  
Bewegen des Prozesses innerhalb der Task nach oben.
- *Move Down*  
Bewegen des Prozesses innerhalb der Task nach unten.
- *Deassign Application Modes*  
Entfernen eines Betriebsmodus aus der aktuellen Task.
- *Deassign Processes*  
Entfernen eines Prozesses aus der aktuellen Task.
- *Delete Undefined Processes*  
Entfernen nicht definierter Prozesses aus der aktuellen Task.
- *Open Module*  
Öffnet das Modul, das den ausgewählten Prozess enthält.

- *Show in Processes*  
Zeigt alle Vorkommen eines Prozesses in der "Processes" Fläche an.
- *Show in Application Modes*  
Zeigt alle einer Task zugewiesenen Betriebsmodi in der „Application Modes“ Fläche an.
- *Set Cache Locking*  
Diese Menüfunktion ist nur für die Arbeit mit ASCET-RP von Bedeutung. Sie ist im ASCET-RP-Handbuch beschrieben.

### Register „Formulas“

---

- **Global Formulas**

#### Hinweis

*Die Menüfunktionen sind auch als Kontextmenü im Register verfügbar.*

- *Add*  
Hinzufügen einer Formel.
- *Add missing*  
Findet undefinierte Formeln in der aktiven Implementierung und fügt für jeden gefundenen Formelnamen eine lineare Formel mit dem Verhalten der Identität hinzu.
- *Rename*  
Ändern des Formelnamen.
- *Delete*  
Löschen der Formel.
- *Edit*  
Öffnen des Formeleditors.
- *File Out*  
Auslagern von Formeln in eine Datei  
(*All* → alle Formeln,  
*Selected* → ausgewählte Formeln).
- *File In*  
Einfügen von Formeln aus einer Datei  
(*All* → alle in der Datei enthaltene Formeln,  
*Select* → in der Datei ausgewählte Formeln).

- *Select all* (<CTRL> + <A>)  
Wählt alle Formeln aus.

Register „Impl. Type“

---

- **Global Implementations**

**Hinweis**

*Die Menüfunktionen sind auch als Kontextmenü im Register verfügbar.*

- *Add*  
Hinzufügen eines Implementierungstypen  
(*cont* → mit Modelltyp *cont*,  
*sdisc* → mit Modelltyp *sdisc*,  
*udisc* → mit Modelltyp *udisc*)
- *Rename*  
Ändert des Namens des Implementierungstypen.
- *Delete*  
Löscht die ausgewählten Implementierungstypen.
- *Edit*  
Öffnet den Editor für Implementierungstypen.
- *Copy implementation*  
Kopiert die aktuellen Einstellungen des ausgewählten Implementierungstyps in die Zwischenablage der Datenbank.
- *Paste implementation*  
Fügt die Einstellungen aus der Zwischenablage der Datenbank in den ausgewählten Implementierungstyp ein.
- *File Out*  
Auslagern von Implementierungstypen in eine XML-Datei  
(*All* → alle Implementierungstypen,  
*Selected* → ausgewählte Implementierungstypen).
- *File In*  
Einfügen von Implementierungstypen aus einer XML-Datei  
(*All* → alle in der Datei enthaltene Implementierungstypen,  
*Select* → in der Datei ausgewählte Implementierungstypen).
- *Select all* (<CTRL> + <A>)  
Wählt alle Implementierungstypen aus.

- **Project Files**

**Hinweis**

*Die Menüfunktionen sind auch als Kontextmenü im Register verfügbar.*

- *View*  
Öffnen der Projektdatei mit der zugehörigen Anwendung.
- *Edit*  
Bearbeiten der Projektdatei mit der zugehörigen Anwendung.
- *Add*  
Hinzufügen einer externen Datei.
- *Delete (<DEL>)*  
Löschen einer Datei.
- *Update*  
Aktualisiert die ausgewählten Projektdateien.
- *Write Selected File(s)*  
Schreibt ausgewählte Projektdateien in das Standardverzeichnis.
- *Write Selected File(s) to*  
Schreibt ausgewählte Projektdateien im ausgewählten Verzeichnis.
- *Write All Files*  
Schreibt alle zum Projekt gehörigen Projektdateien in das Standardverzeichnis.
- *Replace With Backup Contents*  
Ersetzt die Projektdatei(en) durch das Backup aus der Datenbank.
- *Show Default Path*  
Zeigt den Pfad des Standardverzeichnisses an.
- *Explore Default Path*  
Öffnet den Explorer mit dem Pfad des Standardverzeichnisses.
- *Copy From Target*  
Konvertierung des Projektes von einem ausgewählten Target auf das aktuelle Target.
- *Copy To Target*  
Konvertierung des Projektes auf ein ausgewähltes Target.

- *Select all* (<CTRL> + <A>)

Wählt alle Projektdateien aus.

### 4.8.3 Spezifizieren eines Projekts

---

Module und CT-Blöcke werden einbezogen, indem sie zur Fläche „Elements“ des Projekteditors wie in anderen Spezifikationseditoren hinzugefügt werden.

#### **Hinzufügen einer Komponente zu einem Projekt:**

---

- Wählen Sie im Projekteditor **Element** → **Add Item**.  
Das Dialogfenster „Select Item...“ öffnet sich.
- Wählen Sie im Fenster „Select Item...“ aus der Liste „1 Database“ die gewünschte Komponente aus.
- Klicken Sie auf **OK**.  
Die Komponente wird zur Fläche „Elements“ des Projekts hinzugefügt.

Jedes Modul, das in einem Projekt verwendet wird, hat einen Instanznamen. Wie in den Spezifikationseditoren wird jeder Komponente ein Standardname gegeben, wenn sie in ein Projekt einbezogen wird. Dieser Name gilt nur für die Instanz des Moduls im Projekt und hat keinen Einfluss auf das ursprüngliche Modul.

#### **Umbenennen einer Komponente:**

---

- Wählen Sie in der Fläche „Elements“ das Modul aus, das Sie umbenennen wollen.
- Wählen Sie **Element** → **Rename**.
- Geben Sie den neuen Namen ein und drücken Sie <ENTER>.

#### **Löschen einer Instanz einer Komponente:**

---

- Wählen Sie in der Fläche „Elements“ die Komponente, die Sie entfernen wollen.
- Wählen Sie **Element** → **Delete**.  
Die Instanz des Moduls wird gelöscht. Dieser Vorgang hat keine Auswirkung auf die Komponente in der Datenbank.

Es ist möglich, den entsprechenden Editor zu öffnen und jede Komponente, die Teil eines Projekts ist, aus dem Projekteditor heraus zu bearbeiten. Das Bearbeiten einer Komponente aus einem Projekt heraus hat den gleichen Effekt wie deren Bearbeitung aus dem Komponentenmanager, d.h. die funktionale Beschreibung wird verändert, was sich auf alle Instanzen der Komponente auswirkt.

Beim Öffnen eines Spezifikationseditors aus dem Projekteditor heraus werden die aktuellen Projekteinstellungen im Spezifikationseditor aktiviert. Wird zum Beispiel für das aktuelle Projekt die Festkomma-Codegenerierung eingegeben, stellt sich der Spezifikationseditor ebenfalls auf Festkomma-Codegenerierung um. Wird der Spezifikationseditor aus dem Komponentenmanager gestartet, ist nur die Gleitkomma-Codegenerierung verfügbar.

### **Bearbeiten einer Komponente in einem Projekt:**

---

- Wählen Sie in der Fläche „Elements“ des Projekteditors die Komponente aus, die Sie bearbeiten wollen.
- Doppelklicken Sie auf den Namen der Komponente

*oder*

- wählen Sie **Element** → **Edit Component**.

*oder*

- drücken Sie <Enter>.

Der Editor öffnet sich für die zu bearbeitende Komponente.

Es ist möglich, die Daten und Implementierung von Elementen innerhalb eines Projekts zu bearbeiten sowie Daten zu und aus ihnen zu lesen und zu schreiben. Alle Befehle funktionieren wie im Blockdiagrammeditor.

Sie können an das Projekt oder an die darin enthaltenen Komponenten Notizen anhängen. Bei der automatischen Erzeugung einer Dokumentation werden Notizen mit in die Dokumentation eingefügt.

### **Bearbeiten der Notizen eines Projekts:**

---

- Wenn Sie die Notiz des Projekts bearbeiten wollen, wählen Sie **Component** → **Notes**.

*Oder*

- Markieren Sie in der Liste „Elements“ oder im Register „Graphics“ die aufgenommene Komponente, deren Notiz Sie bearbeiten wollen.

- Wählen Sie **Element** → **Notes**.

Der Notizeneditor für den gewählten Eintrag wird geöffnet. Einzelheiten dazu siehe „Notizen“ auf Seite 747.

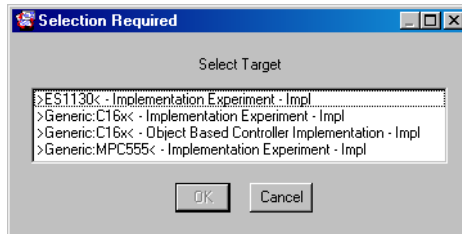
Wenn Sie ein Projekt bearbeiten, das für ein anderes Target geschrieben wurde, können die Einstellungen für den C-Code und das Betriebssystem (s. Seite 421) kopiert werden.

### **C-Code eines gesamten Projektes kopieren:**

---

Um den C-Code aller Klassen und Module eines Projekts von einem anderen Target, einem anderen Experimenttyp oder einer anderen Implementierung zu kopieren, gehen Sie wie folgt vor:

- Wählen Sie im Projekteditor die passenden Optionen für Ihr Target, wie in Kapitel 4.8.8 beschrieben.
- Wählen Sie **Extras** → **Copy C-Code From**.  
Das Fenster „Selection Required“ wird geöffnet.



- Wählen Sie das Target, von dem Sie den Code kopieren wollen, und klicken Sie **OK**.  
Der Targetcode wird auf Ihr Target kopiert.

#### 4.8.4 Definieren der globalen Kommunikation

---

In Komponenten wird der Datenfluss über die Schnittstellenelemente der Komponenten organisiert, d.h. durch Definieren von Eingaben und Argumenten sowie Leseausgaben und Rückgabewerten. In einem Projekt wird die Kommunikation über importierte und exportierte Elemente definiert, die wie globale Variablen funktionieren. Diese Elemente werden entsprechend ihrer Namen gegenseitig aufeinander abgebildet, d.h. die Namen für zwei abzubil-



dende Elemente müssen identisch sein. Aus diesem Grund ist es erforderlich, die Namen in den Modulen so zuzuordnen, dass diese im Projekt zueinander passen.

Ein Element kann entweder durch ein Modul oder durch das Projekt selbst exportiert werden. Jedes exportierte Element kann mehrmals von anderen Modulen importiert werden, aber ein Element kann nur einmal exportiert werden. Enthalten zwei Module innerhalb eines Projekts ein exportiertes Element mit dem gleichen Namen, erscheint eine Fehlermeldung.

Elemente können auch im Projekt erzeugt werden. Sie sind dann mit Elementen gleichen Namens in den Modulen verknüpft. Die Verknüpfung erfolgt immer automatisch, d.h. der Benutzer kann die Verknüpfung von importierten und exportierten Elementen nur durch Zuordnung passender Namen zu diesen Elementen beeinflussen.

### Ansehen der Verknüpfungen von Variablen in einem Projekt:

- Klicken Sie im Projekteditor auf das Register „Binding“.

Label	Imported by	Exported by
dT::dT	...	MyProject
	PI_Module	...
	PT1_Module	...
memory::cont	PI_Module	PT1_Module
n_nominal::msg[cont]	PT1_Module	PI_Module
PI_Module::PI_Module	...	MyProject
PT1_Module::PT1_Module	...	MyProject
temp::msg[cont]	PI_Module	PT1_Module

Unbound only

Die Verknüpfungen der Variablen für das Projekt werden angezeigt.

- Sie können das Feld **Unbound only** unten im Dialogregister aktiv setzen, um die Listenansicht zu filtern.

Im vorstehenden Beispiel gibt es zwei aufgenommene Komponenten und vier globale Elemente:

- Die vom Projekt `MyProject` exportierte Variable `dT`. Diese Variable wird immer standardmäßig im Projekt angelegt und von allen referenzierten Modulen importiert.
- Die beiden aufgenommenen Module `PI_Module` und `PT1_Module`.

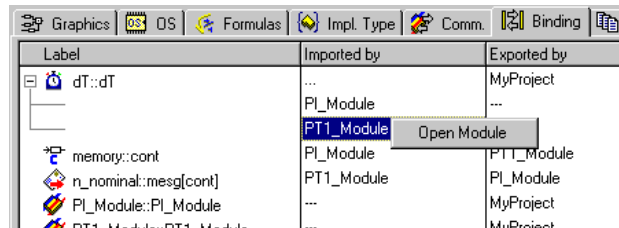
- Die Message `n_nominal`, die vom Modul `PI_Module` exportiert wird, wobei eine Send-Message mit diesem Namen definiert wird. Diese wird vom Modul `PT1_Module` importiert, wo eine Receive-Message mit dem gleichen Namen definiert wird.
- Die Message `temp`, die von `PT1_Module` exportiert und von `PI_Module` importiert wird.
- Die Variable `memory`, die von `PT1_Module` exportiert und von `PI_Module` importiert wird.

Send-Messages sowie Send&Receive-Messages werden standardmäßig als exportierte Elemente definiert, wohingegen Receive-Messages als importierte Elemente definiert werden. Es ist möglich, diese Zuordnungen zu ändern. Variablen müssen immer explizit als importierte oder als exportierte Elemente erklärt werden. Wird ein Element durch mehr als ein Modul importiert, wird dies wie bei der Variablen `dT` im vorstehenden Beispiel durch eine Hierarchieanzeige angegeben.

Wenn Sie feststellen, dass ein Element irrtümlich mehrfach exportiert wird, können Sie die exportierende Komponente aus dem Register „Binding“ heraus öffnen.

#### Öffnen einer Komponente aus dem Register „Binding“:

- Klicken Sie in der Spalte „Imported by“ oder „Exported by“ mit der rechten Maustaste auf eine Komponente und wählen Sie **Open Module** aus dem Kontextmenü.



#### Oder

- Doppelklicken Sie auf die Komponente, die Sie bearbeiten wollen.  
Der Editor für die gewählte Komponente wird geöffnet.

Durch Anklicken des Registers „Comm.“ erhalten Sie eine andere Ansicht der Elementabbildungen, bei der nur die innerhalb des Projekts definierten Messages angezeigt werden. Hier wird der implizierte Datenfluss durch Send- und Receive-Messages dargestellt.

Label	Sender of	Receiver of
n_nominal::msg[cont]	PI_Module	PT1_Module
temp::msg[cont]	PT1_Module	PI_Module

Without Send/Receive Messages

Diese Ansicht zeigt nur die Messages und in welchen Modulen sie als Send- oder Receive-Messages definiert sind. Durch Aktivierung des Feldes unten im Dialogregister kann auch diese Listenansicht gefiltert werden.

Für jedes importierte Element in einem Modul, für das kein exportiertes Element vorhanden ist, wird automatisch ein globales Element erzeugt. Diese beiden Elemente sind dann miteinander verknüpft.

#### Definieren von globalen Elementen in einem Projekt:

- Wählen Sie **Global Elements** → **Resolve Globals**, um die globalen Elemente aufzulösen.

#### Hinweis

*Bevor ein Experiment mit dem Projekt gestartet werden kann, muss dieser Befehl durchgeführt werden.*

Erzeugte globale Elemente, die nicht im Projekt verwendet werden, können durch den Befehl **Delete Unused Globals** entfernt werden.

#### Löschen von nicht genutzten globalen Elementen:

- Wählen Sie **Global Elements** → **Delete Unused Globals**, um alle nicht genutzten Globalelemente zu löschen.

### 4.8.5 Definieren des Scheduling im Betriebssystemeditor

Bei dem Scheduling von Prozessen handelt es sich um die zweite Hauptoperation, die im Projekteditor ausgeführt wird. Das Betriebssystem aktiviert nicht direkt die einzelnen Prozesse. Statt dessen werden durch den Aktivierungsmodus

aus Prozesse in Sequenzen gruppiert, die Tasks genannt werden. Diese Tasks werden durch das Betriebssystem aktiviert, und bei Aktivierung wird die Sequenz von Prozessen, die der Task zugeordnet sind, in der entsprechenden Reihenfolge ausgeführt.

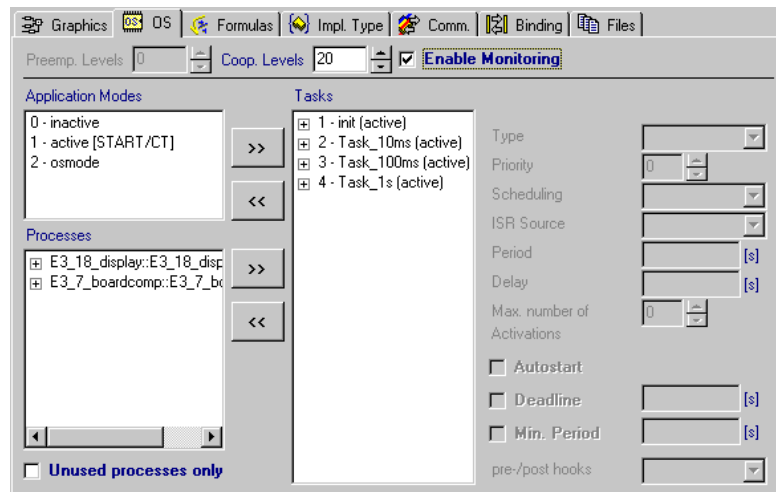
### **Hinweis**

Alle Einstellungen im Betriebssystemeditor müssen gesondert für jedes Target festgelegt werden, wobei jedoch die Einstellungen zwischen den verschiedenen Targets kopiert werden können.

### *Einstellungen des Betriebssystems*

#### **Einrichten des Betriebssystems:**

- Klicken Sie auf das Register „OS“ im Projekteditor, um den Betriebssystemeditor zu öffnen.



- Geben Sie in den Feldern „Preemp. Levels“ bzw. „Coop. Levels“ die Anzahl von kooperativen und präemptiven Ebenen für das Projekt ein.

Die verfügbare Anzahl ist von der Implementierung des Echtzeit-Betriebssystems auf dem aktuellen Target abhängig. Ist das Target ein PC, stehen zum Beispiel 0 präemptive Ebenen

zur Verfügung, da präemptives Multitasking in der PC-Implementierung des Betriebssystems nicht möglich ist.

- Wenn Sie die Debugging-Möglichkeit nutzen wollen, aktivieren Sie die Option **Enable Monitoring**.

Die Einstellungen des Betriebssystems für das aktuelle Target können auf ein anderes Zielgerät kopiert werden bzw. es lassen sich Einstellungen von einem anderen Target auf das aktuelle Target kopieren. Das Umschalten zwischen Targets wird im Abschnitt „Projekteinstellungen“ auf Seite 438 beschrieben.

### **Kopieren von Einstellungen des Betriebssystems:**

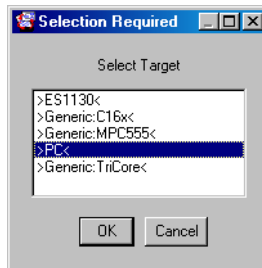
---

- Wählen Sie **Operating System** → **Copy To Target**

oder

- wählen Sie **Operating System** → **Copy From Target**.

Das Dialogfenster „Selection Required“ wird geöffnet.



- Wählen Sie ein Target aus und klicken Sie auf **OK**.

Die Einstellungen werden je nach dem gewählten Befehl entweder vom aktuellen auf das ausgewählte Target oder umgekehrt kopiert.

### *Tasks und Prozesse*

---

Dieser Abschnitt beschreibt, wie mit dem Betriebssystemeditor ein Betriebssystem-Schedule angelegt wird.

## Erstellen einer Task:

---

- Wählen Sie **Task** → **Add**.  
Eine neue Task wird angelegt.
- Geben Sie im Feld „Tasks“ den Namen für die neue Task ein und drücken Sie <ENTER>.

Sobald Sie die erforderlichen Tasks erstellt haben, können Sie zu jeder Task Prozesse hinzufügen.

## Zuordnen eines Prozesses zu einer Task:

---

- Wählen Sie in der Fläche „Tasks“ die Task aus, der Sie einen Prozess zuordnen wollen.
- Wählen Sie in der Fläche „Processes“ den Prozess aus, den Sie der ausgewählten Task zuordnen wollen.

- Wählen Sie **Process** → **Assign**

oder

- klicken Sie auf die Schaltfläche >>

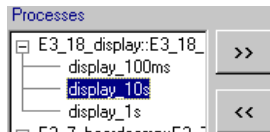
oder

- wählen Sie das Kontextmenü **Assign**

oder

- ziehen Sie den Prozess aus der Fläche „Processes“ mit der Maus auf die gewünschte Task.

Der Prozess wird der Task zugeordnet und wird immer ausgeführt, wenn die Task eingeplant ist



## Entfernen eines Prozesses aus einer Task:

---

Um einen Prozess aus einer Task zu entfernen, gehen Sie wie folgt vor:

- Wählen Sie in der Fläche „Tasks“ den Prozess, den Sie entfernen wollen.

- Wählen Sie **Task** → **Deassign Processes**

oder

- klicken Sie auf die Schaltfläche << (siehe obige Abbildung)

oder

- wählen Sie das Kontextmenü **Deassign Processes**

oder

- ziehen Sie den Prozess aus der Fläche „Tasks“ mit der Maus in die Fläche „Processes“.

Der Prozess wird aus der aktuellen Task entfernt.

Die Reihenfolge der Prozesse innerhalb einer Task bestimmt die Reihenfolge, in der die Prozesse ausgeführt werden.

### **Verschieben eines Prozesses in einer Task:**

---

Um einen Prozess in einer Task zu verschieben, gehen Sie wie folgt vor:

- Wählen Sie einen Prozess in der Fläche „Tasks“ aus.
- Wählen Sie **Task** → **Move Up** oder **Task** → **Move Down**, um die Position eines Prozesses in der Task zu bestimmen.

### **Öffnen einer Komponente aus dem Feld „Tasks“:**

---

Wenn Sie einen Prozess noch einmal bearbeiten wollen, können Sie die Komponente, zu der er gehört, aus dem Feld „Tasks“ heraus öffnen.

- Wählen Sie in der Fläche „Tasks“ den Prozess aus, den Sie bearbeiten wollen
- Wählen Sie aus der Menüleiste **Tasks** → **Open Module**

oder

- wählen Sie aus dem Kontextmenü **Open Module**

oder

- doppelklicken Sie auf den Prozess.  
Der Editor für die zugehörige Komponente wird geöffnet.

### **Zuordnungen von Betriebsmodi, Tasks und Prozessen prüfen:**

---

1. Um den Ursprung eines einer Task zugewiesenen Prozesses festzustellen, gehen Sie wie folgt vor:
  - Wählen Sie in der Fläche „Tasks“ einen Prozess.

- Klicken Sie mit der rechten Maustaste auf den Prozess und wählen Sie aus dem Kontextmenü **Show in Processes**

*oder*

- wählen Sie **Tasks** → **Show in Processes**.  
In der Fläche „Processes“ wird der gewählte Prozess hervorgehoben.

2. Um festzustellen, welcher Task ein Prozess zugewiesen wurde, gehen Sie wie folgt vor:

- Wählen Sie in der Fläche „Processes“ einen Prozess.
- Wählen Sie aus dem Kontextmenü **Show in Tasks**

*oder*

- wählen Sie **Process** → **Show in Tasks**.  
In der Fläche „Tasks“ wird der gewählte Prozess hervorgehoben.

3. Um festzustellen, welcher Betriebsmodus einer Task zugewiesen wurde, gehen Sie wie folgt vor:

- Wählen Sie in der Fläche „Tasks“ eine Task.
- Wählen Sie aus dem Kontextmenü **Show in Application Modes**

*oder*

- wählen Sie **Tasks** → **Show in Application Modes**.  
In der Fläche „Application Modes“ wird der zugewiesene Betriebsmodus hervorgehoben.

4. Um festzustellen, welche Tasks zu einem Betriebsmodus zugewiesen wurden, gehen Sie wie folgt vor:

- Wählen Sie in der Fläche „Application Modes“ einen Betriebsmodus.
- Wählen Sie aus dem Kontextmenü **Show in Tasks**

*oder*



- wählen Sie **Application Mode** → **Show in Tasks**.

In der Fläche „Tasks“ werden alle zugehörigen Tasks hervorgehoben.

### **Grundeinstellungen einer Task:**

Die Konfiguration einer Task bestimmt ihre Priorität unter anderen Tasks, ihren Triggermodus und andere Einzelheiten hinsichtlich der Aktivierung und Ausführung einer Task.

### **Einrichten einer Task (Grundeinstellungen):**

---

#### 1. Triggermodus

Folgende Triggermodi stehen zur Verfügung:

- *Alarm*-Tasks werden gleich zu Beginn jeder im Feld „Period“ definierten Periode ausgelöst.
- *Init*-Tasks werden nur einmal ausgelöst – beim Starten des Betriebsmodus (*Application Mode*), dem sie zugeordnet sind.
- *Software*-Tasks werden durch Befehle des Betriebssystems ausgelöst.
- *Interrupt*-Tasks werden durch Hardwareereignisse ausgelöst. Die verfügbaren Hardwareereignisse sind vom Target abhängig. Ist das Target zum Beispiel ein Transputerboard, könnte eine *Interrupt*-Task durch die am Channel 0 ankommenden Daten ausgelöst werden.
- *Timetable*-Tasks (nur Mikrocontroller-Targets mit ERCOS<sup>EK</sup>-Unterstützung) sind Alarm-Tasks, die in eine Zeitstauertabelle geschrieben werden. Auf diese Weise kann Laufzeit eingespart werden (allerdings auf Kosten höheren Speicherbedarfs).

- Wählen Sie einen Triggermodus aus dem Kombikästchen „Type“ aus.

## 2. Scheduling

- Weisen Sie im Kombikästchen „Scheduling“ der Task einen von drei möglichen Scheduling-Modi zu.

Diese Option ist für *Init-* und *Interrupt-Tasks* nicht verfügbar.

<code>cooperative</code>	Wenn Sie <i>cooperative</i> wählen, unterbricht die Task eine laufende Task mit geringerer Priorität erst, wenn der laufende Prozess abgeschlossen ist.
<code>preemptive</code>	Wenn Sie <i>preemptive</i> wählen, unterbricht die Task eine laufende Task mit niedrigerer Priorität sofort. Der laufende Prozess wird unterbrochen und wieder aufgenommen, nachdem die unterbrechende Task beendet ist.
<code>non-preemptable</code>	Weder der aktuelle Prozess noch die laufende <i>nicht-unterbrechbare</i> Task werden unterbrochen, wenn eine Task mit höherer Priorität aktiviert wird.

Da in ERCOS<sup>EK</sup> alle präemptiven Tasks eine höhere Priorität haben als kooperative Tasks, ergibt sich folgendes Verhalten bei taskunterbrechung:

- Kooperative Tasks unterbrechen niemals präemptive Tasks.
  - Kooperative Tasks unterbrechen einander erst, wenn der laufende Prozess (oder die gesamte Task, falls diese nur einen Prozess hat) abgeschlossen ist.
  - Präemptive Tasks unterbrechen immer sofort.
- Details zu den Scheduling-Modi finden Sie in Kapitel 1.1.1 „Scheduling“ im ASCET-Referenzhandbuch.

### Hinweis

Bei einzelnen Mikrocontrollern (z. B. MPC555) kann die Verwendung nicht-unterbrechbarer Tasks an bestimmte Bedingungen geknüpft sein (siehe das jeweilige ASCET-SE-Referenzhandbuch). Bei Verletzung dieser Bedingungen werden von der Codegenerierung entsprechende Fehlermeldungen erzeugt. ASCET-SE für RTA-OSEK bietet eine andere Auswahl von Scheduling-Modi; Details siehe ASCET-SE user's guide.

### 3. Priorität

Die Auslösung von Tasks wird durch ihre Priorität bestimmt, wenn gleichzeitig mehrere Tasks zur Ausführung anstehen. Wurde zum Beispiel geplant, dass mehrere *Alarm*-Tasks gleichzeitig ausgelöst werden, so wird die Task mit der höchsten Priorität zuerst ausgelöst. Tasks werden unterbrochen, wenn eine Task mit einer höheren Priorität als die aktuell laufende Task aktiviert wird. Das generelle Prioritätsschema finden Sie in Kapitel 1.1.2 „Tasks“ im ASCET-referenzhandbuch.

- Ordnen Sie der Task eine Priorität zu, indem Sie im Feld „Priority“ eine Zahl eingeben.

#### Hinweis

*Nicht-unterbrechbare Tasks und präemptive Tasks haben einen gemeinsamen Prioritätenbereich. Kooperative Tasks haben einen eigenen Prioritätenbereich.*

Diese Option ist für *Init*-Tasks nicht verfügbar.

### 4. Trigger

Diese Einstellung bestimmt, welches Ereignis als Trigger für eine *Interrupt*-Task fungieren soll. Für alle anderen Tasks ist das Kombikästchen nicht verfügbar.

Die im Kombikästchen aufgelisteten Optionen sind vom aktuellen Target abhängig.

- Wählen Sie den Trigger aus dem Kombikästchen „ISR Source“ aus.

### 5. Andere

Der Periodenwert bestimmt, wie oft die Task aktiviert wird. Beträgt der Periodenwert zum Beispiel 0.01, wird die Task nach jeder hundertstel Sekunde aktiviert.

- Geben Sie einen Periodenwert in Sekunden im Feld „Period“ ein.

Diese Einstellung ist nur für *Alarm*- und *Time-table*-Tasks verfügbar.

Wird eine Verzögerung (*Delay*) festgelegt, wird die entsprechende Task zum ersten Mal nach der angegebenen Zeit aktiviert. Ist der Verzögerungswert 0, wird die Task aktiviert, sobald das Programm startet, und danach zu Beginn jeder Periode.

- Geben Sie im Feld „Delay“ einen Verzögerungswert in Sekunden ein.

Diese Einstellung ist nur für *Alarm-* und *Timetable-*Tasks verfügbar.

Der Wert *Max. number of Activations* bestimmt, wie oft eine Task parallel aktiviert werden kann. Wird eine Task erneut aktiviert, bevor die Ausführung nach ihrer vorherigen Aktivierung beendet ist, ist die Task dann doppelt aktiviert. Zum Einsparen von Systemressourcen kann die maximale Anzahl von gleichzeitigen Aktivierungen begrenzt werden.

- Passen Sie den Wert im Feld „Max. number of Activations“ an (*Alarm-*, *Timetable-* und *Software-*Tasks).

Diese Einstellung ist für *Alarm-*, *Timetable-* und *Software-*Tasks verfügbar.

### Hook-Routinen einstellen:

#### Hinweis

*Dies gilt nicht für ASCET-SE für RTA-OSEK.*

Sie haben die Möglichkeit, für jede Task die Generierung von Hook-Routinen für Debug-Zwecke einzustellen.

### Hook-Routinen für eine Task einstellen:

- Stellen Sie sicher, dass die Option **Enable Monitoring** aktiviert ist.
- Wählen Sie aus dem Kombikästchen „pre/post hooks“ eine der verfügbaren Optionen.

none                      keine Debug-Funktion

monitoring      vollständige Debug-Funktion

Entsprechend den Einstellungen werden bei der Codegenerierung geeignete Datenstrukturen erzeugt und die passenden ERCOS<sup>EK</sup> Bibliotheken eingebunden. Zu diesem Zweck erzeugt ASCET die Make-Variable `E_HOOKS`. Die folgende Tabelle gibt die Werte von `E_HOOKS` für die verschiedenen Einstellungen im OS-Editor an:

<b>Enable Monitoring</b>	<b>pre-/post hooks</b>	<b>generierter Wert für <code>E_HOOKS</code></b>	<b>Verfügbarkeit von <code>dT</code> im Code</b>
ja	„monitoring“ für mindestens eine Task	MONITORING	ja, zusätzlich Debuginformation
nein	„monitoring“ für mindestens eine Task	DTONLY	ja
beliebig	„none“ für alle Tasks	DTONLY	nein

Für Simulationen mit Experimentaltargets ist nur die Option `monitoring` von Bedeutung, da diese die Erzeugung der Überwachungsvariablen (s. Seite 430) steuert. In vollem Umfang kommen die Optionen in Verbindung mit Mikrocontroller-Targets zum Einsatz, sie sind daher im ASCET-SE-Benutzerhandbuch beschrieben.

#### **OSEK-spezifische Einstellungen:**

Zusätzlich können Sie OSEK-spezifische Einstellungen vornehmen. Da diese Optionen nur in Verbindung mit Mikrocontroller-Targets von Bedeutung sind, entnehmen Sie weitergehende Informationen bitte dem ASCET-SE-Benutzerhandbuch.

#### **Deadline- und Min. Period-Optionen setzen:**

- Aktivieren Sie die Option **Deadline**, wenn Sie sicherstellen wollen, dass zwei aufeinanderfolgende Taskaktivierungen eine bestimmte Zeitdifferenz nicht überschreiten.  
Das zugehörige Eingabefeld wird aktiviert.
- Geben Sie den maximalen Abstand in Sekunden zwischen zwei Aktivierungen ein.  
Die Option **Deadline** ist für *Alarm-*, *Time-table-* und *Software-*Tasks verfügbar.
- Aktivieren Sie die Option **Min. Period**, wenn Sie sicherstellen wollen, dass zwei aufeinanderfolgende Taskaktivierungen eine bestimmte Zeitdifferenz nicht unterschreiten.  
Das zugehörige Eingabefeld wird aktiviert.

- Geben Sie den minimalen Abstand in Sekunden zwischen zwei Aktivierungen ein.

Die Option **Min. Period** ist nur für *Interrupt-Tasks* verfügbar.

### Autostart-Option setzen:

---

Die Option **Autostart** ist für *Software-Tasks* und *Alarm-Tasks* verfügbar. Sie hat für jeden der beiden Typen eine unterschiedliche Bedeutung.

- Wählen Sie in der Liste „Tasks“ eine *Software-Task*.
- Aktivieren Sie die Option **Autostart**.  
Damit wird die Softwaretask bei der Initialisierung der zugehörigen Betriebsmodi (Application Modes) einmalig ausgeführt.

*Oder*

- Wählen Sie in der Liste „Task“ eine *Alarm-Task*.
- Aktivieren Sie die Option **Autostart**.  
Damit ist festgelegt, dass der Timer für die ausgewählte Task bei der Initialisierung der zugehörigen Betriebsmodi gestartet wird.

### Überwachungsoption

---

Die Überwachungsoption bietet drei Variable für jede Task, die zu Überwachungs- und Testzwecken während dem Experimentieren mit Projekten verwendet werden können, sowie eine weitere Variable, die zur Überwachung aller Tasks dient. Die Variablen werden automatisch erzeugt, wenn die Option **Enable Monitoring** aktiviert ist und wenn für die Task im Kombikästchen „pre-/post hooks“ der Eintrag `monitoring` ausgewählt wird. Dies muss für jede Task einzeln erfolgen, für die die Überwachungsvariablen erzeugt werden sollen. Es werden die folgenden drei taskindividuellen Überwachungsvariablen erzeugt:

<code>cycleStartTime_&lt;taskname&gt;</code>	Gibt den Zeitpunkt der letzten Aktivierung der Task in Sekunden an.
<code>cycleTime_&lt;taskname&gt;</code>	Gibt die Zeit in Sekunden an, die für die Ausführung der Task erforderlich ist.
<code>dT_&lt;taskname&gt;</code>	Gibt den Zeitunterschied zwischen der letzten und der aktuellen Aktivierung der Task in Sekunden an.

Darüber hinaus wird die folgende Variable einmal für alle Tasks angelegt:

<code>runtime_violation</code>	Gibt an, wie oft der Schedule der Task nicht erfüllt wurde.
--------------------------------	---

### Überwachungsoption aktivieren:

---

- Wählen Sie in der Liste „Tasks“ eine oder mehrere Tasks.
- Wählen Sie im Kombikästchen „pre-/post hooks“ den Eintrag `monitoring`.

Bei der nächsten Codegenerierung werden für jede Task die Überwachungsvariablen erzeugt. Sie können diese in den Messfenstern anzeigen oder im Data-Logger (nur Offline-Experiment, s. „Der Data-Logger“ auf Seite 641) aufzeichnen.

### *Betriebsmodi*

---

Das Betriebssystem kann verschiedene Betriebsmodi (*Application Modes*) umfassen, wobei jeder Betriebsmodus über seinen eigenen Satz Tasks verfügt, der ihm zugeordnet wurde. Auf diese Weise lassen sich verschiedene Betriebszustände des gesteuerten Systems modellieren, d.h. ein Motor kann verschiedene Betriebsmodi für die Zeit des Warmlaufens und für den normalen Betrieb haben.

### Erstellen eines Betriebsmodus:

---

- Wählen Sie **Application Mode** → **Add**.  
Es wird ein neuer Betriebsmodus angelegt und im Feld „Application Modes“ angezeigt.
- Geben Sie einen Namen für den Betriebsmodus ein und drücken Sie <ENTER>.
- Soll der Betriebsmodus der Startmodus sein, wählen Sie **Application Mode** → **As Start Mode** aus.

Der Betriebsmodus, der zum Startmodus erklärt wird, ist der Modus, in dem sich das System beim Starten befindet. Das Umschalten zwischen den Betriebsmodi erfolgt mit Befehlen des Betriebssystems.

## Zuordnen von Betriebsmodi zu einer Task:

---

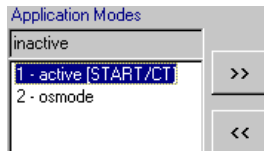
- Wählen Sie einen oder mehrere Betriebsmodi aus der Fläche „Application Modes“ aus.  
Sie können mehrere Betriebsmodi auswählen, indem Sie darauf klicken und die Taste <CTRL> gedrückt halten.
  - Wählen Sie die Task aus, die Sie dem Betriebsmodus bzw. den Betriebsmodi in der Fläche „Tasks“ zuordnen wollen.
  - Wählen Sie **Application Mode** → **Assign**.
- oder

- wählen Sie die gewünschten Betriebsmodi aus und ziehen sie mit der Maus auf die gewünschte Task.

Eine Task kann verschiedene Betriebsmodi haben; sie wird immer dann entsprechend ihren Taskeinstellungen geplant, wenn ein ihr zugeordneter Betriebsmodus aktiviert ist.

- Wählen Sie **Task** → **Deassign Application Modes**, um die Zuweisung eines Betriebsmodus zu einer Task aufzuheben.

Anstelle der Menüfunktionen können Sie auch die nebenstehend abgebildeten Schaltflächen verwenden.



### Hinweis

Die Umschaltung von einen in den anderen Betriebsmodus erfolgt durch Aufruf eines Betriebssystemdienstes. Näheres findet sich in der API-Beschreibung des ERCOS<sup>EK</sup> Handbuchs.

## 4.8.6 Verwaltung externer Projektdateien

---

Zu einem Projekt gehören in der Regel weitere Projektdateien. Dies können zum Beispiel Protokolle, Dokumente für den Schriftverkehr oder auch C-Dateien sein. Damit die Zuordnung der externen Projektdateien eindeutig dem Projekt zugeordnet werden können, werden diese Dateien im Projekteditor unter dem Register „Files“ verwaltet.



Das Register „Files“ enthält eine Tabelle mit Name, Größe (in Byte) und Erzeugungsdatum der Dateien. Die Pfadangaben dieser Dateien werden als absolute Pfade gespeichert; nur wenn die Datei in einem von sechs vorgegebenen Verzeichnissen liegt, wird ein relativer Pfad gesetzt. In dem Fall wird in der Spalte „Abstract“ eines der folgenden Schlüsselwörter angezeigt.

Schlüsselwort	Pfad
%MAIN%	ETAS\Ascet5.2 (ASCET-Installationsverzeichnis)
%CGEN%	ETAS\Ascet5.2\cgen (Zielverzeichnis d. Codegenerierung)
%MAINTARGET%	ETAS\Ascet5.2\target (Target-Hauptverzeichnis)
%TARGET%	ETAS\Ascet5.2\target\<<target name> (Targetverzeichnis für das aktuelle Target; ist z. B. das Target PC gewählt, steht TARGET für ETAS\Ascet5.2\target\PC.)
%DATA%	ETASData\Ascet5.2 (Produktdatenverzeichnis)
%SPECIFIC%	benutzerdefinierter Pfad, einstellbar im Optionsfenster (siehe Seite 50)

### Hinzufügen einer externen Projektdatei:

- Klicken Sie im Projekteditor auf das Register „Files“.

Name	Size	Date	Abstract
a_basdef.h	1732 Bytes	09.08.2001 07:45:14	%MAINTARGET%
bla.c	5839 Bytes	21.11.2002 15:56:52	---
docu.ini	9621 Bytes	24.10.2002 08:09:22	%MAIN%
mem_lay.a2l	2 Bytes	30.11.2000 18:17:32	%TARGET%
ProcModel.exp	87040 Bytes	11.12.2001 10:05:08	%DATA%\export\
PROJCT.c	4253 Bytes	22.11.2002 13:36:47	%CGEN%
project.a2l	0 Bytes	22.11.2002 13:39:11	%DATA%
test_asam.a2l	0 Bytes	22.11.2002 13:28:04	%SPECIFIC%

Alle eingebundenen Projektdateien werden angezeigt.

- Wählen Sie **Project Files** → **Add**.  
Ein Dateiauswahlfenster wird geöffnet.
- Wählen Sie eine Datei und drücken Sie **<ENTER>**.

Die ausgewählte Datei wird in den Projekteditor eingebunden.

## Ansehen einer Projektdatei:

---

Sie können eine Projektdatei aus dem Register „Files“ heraus ansehen.

- Wählen Sie im Register „Files“ die Projektdatei, die Sie ansehen wollen.
- Wählen Sie aus der Menüzeile **Project Files** → **View**

oder

- wählen Sie aus dem Kontextmenü **View**.  
Die Projektdatei wird mit der zugehörigen Anwendung geöffnet.  
Ist mit dem Dateityp keine Anwendung verknüpft, bleibt der Befehl wirkungslos.

## Löschen einer Projektdatei im Projekteditor:

---

- Wählen Sie die zu löschende Datei im Projekteditor aus.
- Wählen Sie **Project Files** → **Delete**.  
Die entsprechende Datei wird aus dem Projekt (aber nicht von der Festplatte) gelöscht.

## Schreiben von Projektdateien:

---

- Wählen Sie eine oder mehrere Dateien im Projekteditor aus.
- Wählen Sie **Project Files** → **Write Selected File(s)**, um die Dateien in ihre Standardverzeichnisse zu schreiben.

Oder

- Wählen Sie **Project Files** → **Write Selected File(s) to**, um die Dateien in ein beliebiges Verzeichnis zu schreiben.

Oder

- Wählen Sie **Project Files** → **Write All Files**, um alle Dateien in ihre Standardverzeichnisse zu schreiben.
- Wenn Sie wissen wollen, wo das Standardverzeichnis einer Projektdatei liegt, wählen Sie **Project Files** → **Show Default Path**, um den Pfad anzuzeigen.

- Wählen Sie **Project Files** → **Explore Default Path**, um das Standardverzeichnis der ausgewählten Datei im Explorer zu öffnen.

### Hinweis

**Project Files** → **Show Default Path** und **Project Files** → **Explore Default Path** sind nicht verfügbar, wenn mehrere Projektdateien ausgewählt wurden.

### Aktualisieren der Projektdatei:

Wenn Sie auf der Festplatte gespeicherte Projektdateien geändert haben, können Sie diese im Projekt wie folgt aktualisieren.

- Wählen Sie eine oder mehrere Dateien im Projekteditor aus.
- Wählen Sie **Project Files** → **Update**.  
Die in der Datenbank gespeicherte Versionen der gewählten Projektdateien werden durch die auf der Festplatte gespeicherten Versionen ersetzt.

In der Datenbank werden immer ältere Versionen der Projektdateien als Backup gehalten. Wenn Sie die Backup-Version brauchen, gehen Sie wie folgt vor.

- Wählen Sie eine oder mehrere Dateien aus.
- Wählen Sie **Project Files** → **Replace With Backup Contents**.  
Die aktuellen Versionen der Projektdateien werden durch die Backup-Versionen ersetzt.

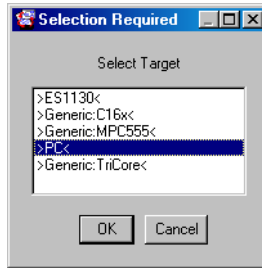
### Kopieren der Projektdatei:

- Wählen Sie **Project Files** → **Copy From Target**.

Oder

- Wählen Sie **Project Files** → **Copy To Target**.

- Das Dialogfenster „Selection Required“ wird geöffnet.



- Wählen Sie ein Target aus und klicken Sie auf **OK**.

Die Datei wird je nach dem gewählten Befehl entweder vom ausgewählten auf das aktuelle Target oder umgekehrt kopiert.

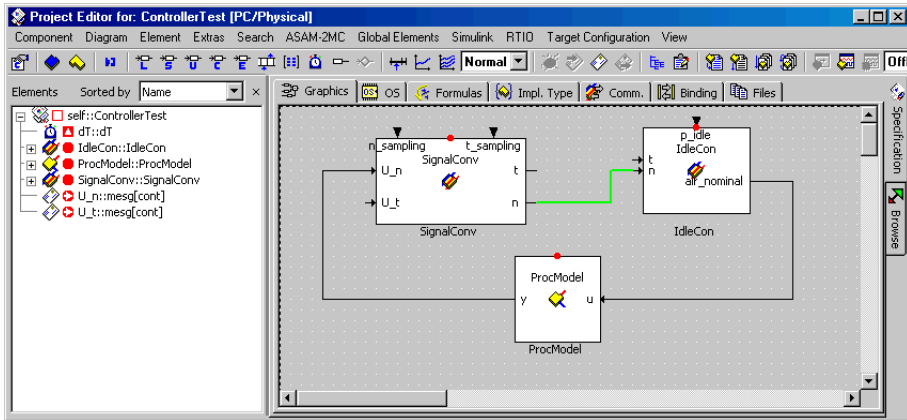
#### 4.8.7 Hybride Projekte

Hybride Projekte sind Projekte, die sowohl CT-Blöcke als auch Module enthalten. CT-Blöcke implementieren Streckenmodelle, d.h. mathematische Modelle der Prozesse, die durch Embedded Control Systeme gesteuert werden.

##### **Einrichten eines hybriden Projekts:**

- Richten Sie die Module des Projekts wie weiter vorn in diesem Kapitel beschrieben ein.
- Fügen Sie die gewünschten CT-Blöcke ein.
- Klicken Sie auf das Register „Graphics“ des Projekteditors.
- Ziehen Sie die CT-Blöcke und Module, die Sie im hybriden Projekt verbinden wollen, in den Zeichenbereich.

- Verbinden Sie die CT-Blöcke und Module wie Elemente in Blockdiagrammen.



Sie können das Layout der einzelnen Komponenten wie im Abschnitt „Layout von aufgenommenen Komponenten“ beschrieben anpassen.

Die Kommunikation zwischen CT-Blöcken sowie zwischen CT-Blöcken und Modulen funktioniert über die grafischen Verbindungen wie bei Blockdiagrammen. In der aktuellen Version von ASCET können Sie auch Module auf diese Weise verbinden. Die Verbindungslinien zwischen Modulen haben jedoch keinen Einfluss auf die Kommunikation zwischen den Modulen (die immer auf einem auf Namen beruhenden Verknüpfungsmechanismus beruht, wie vorstehend beschrieben). Die Verbindungslinien zwischen den Modulen werden als „Kommentarlinien“ interpretiert und können zur Veranschaulichung verwendet werden. Die Farbe der Kommentarlinien wird im ASCET Optionsfenster festgelegt (siehe „Knoten „Colors““ auf Seite 57).

Alle für das Arbeiten mit CT-Blöcken erforderlichen Tasks werden automatisch erzeugt, wobei keine zusätzlichen Tasks zur Aktivierung von CT-Blöcken erzeugt werden können. Die Tasks `initialize_CT` und `terminate_CT` werden nur einmal für das gesamte kombinierte Projekt angelegt. Zusätzlich werden eine `simulate`-Task und eine `event`-Task für jeden im Modell enthaltenen CT-Block erzeugt. Diese Tasks werden wie andere Tasks auch eingerichtet (siehe Seite 425).

#### Definieren des Scheduling von CT-Blöcken:

- Klicken Sie auf das Register „OS“.
- Wählen Sie eine CT-Task in der Fläche „Tasks“.

- Passen Sie die Einstellungen für den Triggermodus usw. nach Bedarf an.

Mit dem hybriden Projekt kann nun wie mit einem Projekt, das nur Module enthält, experimentiert werden. In der Experimentierumgebung muss jedem CT-Block eine Integrationsmethode zugeordnet werden. Innerhalb des gleichen Projekts können den Blöcken verschiedene Integrationsmethoden zugeordnet werden, wodurch Experimente mit mehreren Geschwindigkeiten möglich sind. Die Zuordnung von Integrationsmethoden wird im Abschnitt „Experimentieren mit zeitkontinuierlichen Blöcken“ auf Seite 361 beschrieben.

#### 4.8.8 Projekteinstellungen

---

ASCET ermöglicht das Experimentieren auf verschiedenen Targetplattformen, wie zum Beispiel PC, ES1000 oder verschiedene Mikrocontroller-Targets. Für jedes Target steht eine Reihe von Build-, Experimentalcode- oder Produktionscode-Optionen zur Verfügung, ebenso Optimierungsoptionen und Optionen für die ASAM-MCD-2MC-Generierung.

##### **Hinweis**

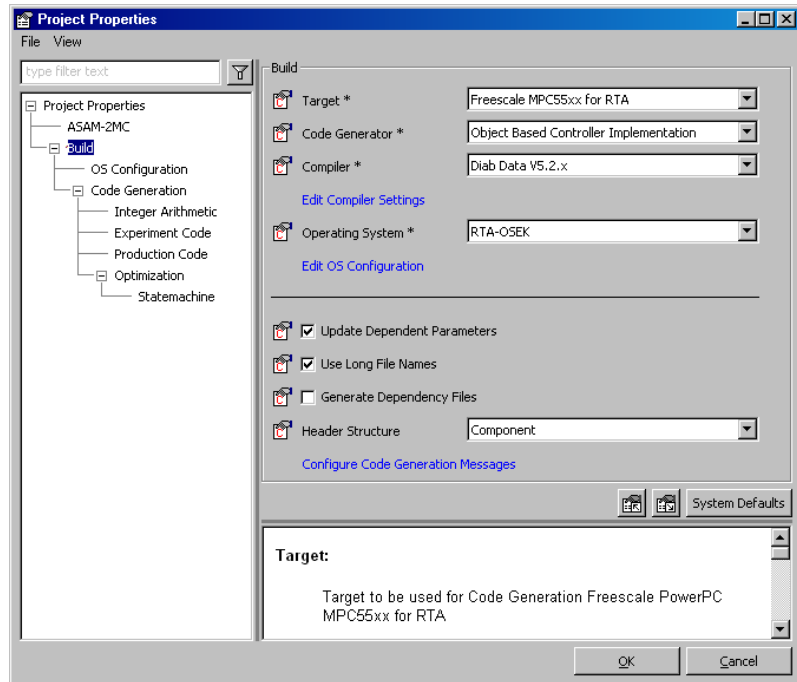
*Für ein einzelnes ASCET-Modul ohne Projektkontext kann Code nur mit dem Codegenerator `Physical Experiment` (s. Seite 443) generiert und simuliert werden. Für jeden anderen Codegenerator muss das Modul in ein Projekt integriert werden. Dafür kann zu jeder Klasse und jedem Modul auch ein sogenanntes Standard- oder Default-Projekt definiert werden. Nur auf diese Weise sind die benötigten Implementierungen verfügbar. Ohne Projektkontext fehlt die Konvertierungsformel sowie sämtliche Implementierungen importierter Größen.*

## Anpassen der Projekteinstellungen:



- Klicken Sie auf die Schaltfläche **Project Properties**.

Das Dialogfenster „Project Properties“ wird im Knoten „Build“ geöffnet.



- Wählen Sie die gewünschte Einstellung aus dem Kombikästchen.

*Oder*

- Aktivieren oder deaktivieren Sie eine Option.

*Oder*

- Geben Sie im Eingabefeld einen passenden Wert ein.

*Oder*

- Klicken Sie auf einen Link (blauer Text), um weitere Einstellmöglichkeiten zu öffnen.

- Wiederholen Sie den Vorgang für die anderen Optionen, die Sie anpassen wollen.
- Klicken Sie auf die Schaltfläche **System Defaults**, um die Standardeinstellungen wiederherzustellen.

#### **Hinweis**

Die Schaltfläche **System Defaults** beeinflusst **alle** Knoten.

- Wechseln Sie in einen anderen Knoten, um weitere Optionen zu setzen,
- oder
- klicken Sie auf **OK**, um die Einstellungen zu übernehmen und das Fenster zu schließen.

Bei den Änderungen in einigen Knoten unterhalb von „Build“ wird der in der Datenbank vorhandene generierte Code ungültig. Welche Knoten betroffen sind, ist abhängig vom gewählten Code-Generator (Seite 443); die Tabelle zeigt die verschiedenen Kombinationen.

<b>Code-Generator</b>	<b>Knoten</b>
Physical Experiment	„Code Generation“ (Seite 446)
Quantized Physical Experiment	„Experiment Code“ (Seite 451) „Optimization“ (Seite 452)
Implementation Experiment	„Code Generation“ „Integer Arithmetic“ (Seite 448) „Experiment Code“ „Optimization“
Object Based Controller Implementation	„Code Generation“ „Integer Arithmetic“ „Production Code“ (Seite 452) „Optimization“

#### **Projekteinstellungen filtern:**

Sie können die Projekteinstellungen filtern und so die Anzeige übersichtlicher gestalten.

- Filtern Sie die Projekteinstellungen wie in „Filtern der Optionen:“ auf Seite 41 für die ASCET-Optionen beschrieben.



## Projekteinstellungen speichern und laden:

---

Sie können die vorgenommenen Einstellungen in einer XML-Datei speichern und gespeicherte Einstellungen laden. Dies ist allerdings nur aus dem obersten Knoten, „Project Properties“, möglich.

### 1. Projekteinstellungen sichern



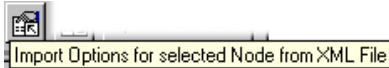
- Wählen Sie im Fenster „Project Properties“ den Knoten „Project Properties“.
- Klicken Sie auf die Schaltfläche **Export Options of Selected Node into XML file**.

Das Windows-Dateiauswahlfenster öffnet sich. Das Format \*.xml ist fest eingestellt.

- Geben Sie Pfad und Name der Datei an.
- Klicken Sie auf **Speichern**.

Ihre Einstellungen werden in der angegebenen Datei gesichert.

### 2. Projekteinstellungen laden



- Wählen Sie im Fenster „Project Properties“ den Knoten „Project Properties“.
- Klicken Sie auf die Schaltfläche **Import Options for selected Node from XML File**.

Ein Bestätigungsfenster öffnet sich.

- Wenn das Bestätigungsfenster in Zukunft nicht mehr angezeigt werden soll, deaktivieren Sie die Option **Show next time**. (Siehe auch Seite 47.)
- Bestätigen Sie das Überschreiben der vorhandenen Optionen mit **OK**.

Der Windows-Dateiauswahldialog öffnet sich. Das Format \*.xml ist fest eingestellt.

- Wählen Sie die XML-Datei, die die Projekteinstellungen enthält.
- Klicken Sie auf **Öffnen**.

Die Einstellungen in der angegebenen Datei werden geladen.

Die Projekteinstellungen sind zunächst nach ASAM-2MC-MCD-Einstellungen (Knoten „ASAM-2MC“) und Build-Einstellungen (Knoten „Build“) unterteilt. Zusammengehörige Build-Einstellungen sind in Unterknoten zusammengefasst.

### Knoten „ASAM-2MC“

---

Die folgenden ASAM-MCD-2MC-Optionen stehen zur Verfügung:

- *ROM Code* schaltet die Erzeugung von ROM-Code für Experimentaltargets um.  
Wenn aktiv, enthält die \* .cod-Datei ROM- und RAM-Code für den Standalone-Modus (siehe ASCET-RP-Benutzerhandbuch). Eine ausführbare Datei (\* .cod) für das Experimentaltarget wird erzeugt.

- *Hierarchical Names* schaltet zwischen einfachen Elementnamen und vollständigen Namen in ASAM-MCD-2MC-Dateien um.  
Die Verwendung von vollständigen Namen hilft, Namenskonflikte in großen Projekten zu vermeiden. In erzeugten ASAM-MCD-2MC-Dateien werden die vollständigen Namen in umgekehrter Reihenfolge aufgelistet:

```
<elementName> [ . <className> ] [ . <moduleName> ]
```

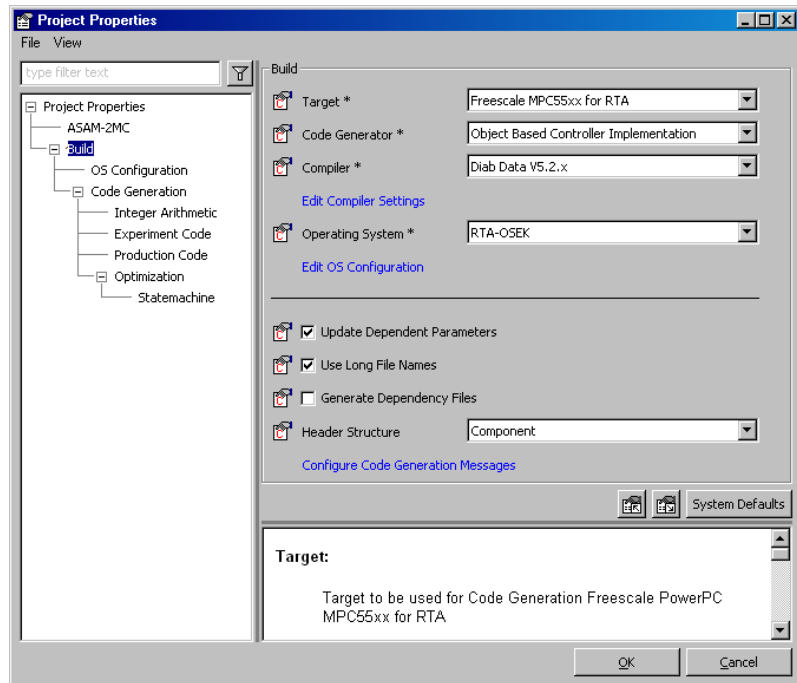
- *Formulas with Unit* legt fest, ob Maßeinheiten in Formeln geschrieben werden (aktiviert) oder nicht (deaktiviert).
- *Suppress exported Parameters* legt fest, ob exportierte Parameter in der ASAM-MCD-2MC-Generierung unterdrückt werden sollen (aktiviert) oder nicht (deaktiviert).

Manche Compiler optimieren den Zugriff auf exportierte Parameter im C-Code, indem sie sie durch ihren Wert ersetzen. Durch Kalibrieren solcher Parameter kann es zu Inkonsistenzen zwischen der exportierenden und der importierenden Komponente kommen. Die Option verhindert das Kalibrieren solcher Elemente.

- *Suppress exported Elements* legt fest, ob die exportierten Elemente von Prototyp-Komponenten bei der ASAM-MCD-2MC-Generierung ignoriert werden (aktiviert) oder nicht (deaktiviert).
- *Suppress grouping Information* legt fest, ob die Strukturierung der Mess- und Verstellgrößen mit Hilfe der Schlüsselwörter `GROUP` und `SUBGROUP` bei der ASAM-MCD-2MC-Generierung vorgenommen (deaktiviert, Standard) oder unterdrückt wird (aktiviert).

## Knoten „Build“

Die folgenden Optionen stehen im Knoten „Build“ zur Verfügung.



- *Target* bestimmt das experimentelle oder Mikrocontroller-Target. Das ASCET-Grundsystem beinhaltet nur das Target PC, ASCET-RP fügt die Experimentaltargets hinzu, die verschiedenen ASCET-SE-Ausgaben die Mikrocontrollertargets. Die tatsächlich verfügbare Auswahl hängt also davon ab, welche Produkte Sie installiert haben.
- *Code Generator* bestimmt, welche Art von Arithmetik im generierten Code verwendet wird. Es stehen folgende Optionen zur Verfügung:
  - *Physical Experiment* (Gleitkomma-Arithmetik)
  - *Quantized Physical Experiment* (quantisierte Gleitkomma-Arithmetik)
  - *Implementation Experiment* (Festkomma-Arithmetik)Ist auf Ihrem Computer ASCET-SE für ein Mikrocontroller-Target installiert, ist für dieses Target nur die folgende Option verfügbar:
  - *Object Based Controller Implementation*

- *Compiler* bestimmt den Compiler, der für die Generierung des Codes für das Projekt verwendet wird. Ein entsprechender Compiler wird mit ASCET für jedes Experimentaltarget geliefert.
- *Edit Compiler Settings* – Dieser Link öffnet das ASCET-Optionsfenster. Dort können Sie im Knoten „Compiler“ (s. Seite 62) im geeigneten Unterknoten den Compilerpfad und weitere Optionen setzen.
- *Operating System* zeigt das Betriebssystem des aktuellen Targets. Wenn Sie ASCET-SE für RTA-OSEK installiert und das entsprechende Target gewählt haben, können Sie im Eingabefeld eine Version von RTA-OSEK eingeben.
- *Edit OS Configuration* – Dieser Link öffnet den Knoten „OS Configuration“ (s. Seite 445).
- *Use Long File Names* kann deaktiviert werden, um zu gewährleisten, dass die generierten Dateinamen dem Muster 8.3 (`filename.ext`) entsprechen. Ansonsten werden lange Dateinamen (identisch zu den jeweiligen Elementnamen) generiert.

### **Hinweis**

*Diese Option ist nur verfügbar, wenn ein Compiler gewählt wurde, der lange Dateinamen unterstützt. Für die Compiler DIAB 4.1 und Borland ist die Option deaktiviert.*

- *Generate Dependency Files* schaltet die Generierung von Make-Dateien, die die Abhängigkeiten aller generierten Header-Dateien beinhalten, ein (aktiviert) oder aus (deaktiviert; Standard). Header-Dateien, die vom Benutzer mit `#include`-Statements eingebunden werden, werden dabei nicht erfasst. Für jede generierte `*.c`-Datei wird eine Make-Datei angelegt.

- *Header Structure* steuert die Erzeugung der Header-Dateien bei Verwendung von **Component** → **File Out Generated Code**. Für die Compilierung mit ASCET bleibt die modulare Header-Struktur erhalten. Zur Auswahl stehen folgende Möglichkeiten:

<b>Component</b>	Für jede Klasse und jedes Modul im Projekt wird eine eigene Header-Datei erzeugt.
<b>Module</b>	Für jedes Modul im Projekt wird eine eigene Header-Datei erzeugt. Enthält ein Modul eine oder mehrere Klassen, wird die Header-Information dieser Klassen in die Header-Datei des Moduls eingefügt. Wenn eine Klasse in mehreren Modulen des Projekts eingebunden ist, wird die Header-Information der Klasse in jede der Header-Dateien der Module eingefügt.
<b>Project</b>	Für das gesamte Projekt wird eine einzige Header-Datei erzeugt, die sämtliche Informationen aus den Modulen und Klassen des Projekts enthält. Diese Header-Datei muss in alle *.c-Dateien eingebunden werden.

### Hinweis

*Die Header von internen und externen C-Code-Klassen und der Betriebssystemkomponente werden von dieser Option nicht erfasst (es sei denn, für die externe C-Code-Klasse wurde **use header global** aktiviert).*

- *Configure Code Generation Messages* – Dieser Link öffnet das Fenster „CodeGen Message Configuration“. Dort können Sie die Meldungen aus der Codegenerierung und dem Build-Prozess konfigurieren; siehe „Meldungen im Fenster „CodeGen Message Configuration“ konfigurieren“ auf Seite 162.

### Knoten „OS Configuration“

Die Optionen in diesem Knoten sind nur verfügbar, wenn ASCET-SE für RTA-OSEK installiert ist und in den Build-Optionen (s. Seite 443) ein entsprechendes Mikrocontroller-Target ausgewählt wurde. Sie sind daher im ASCET-SE-Benutzerhandbuch beschrieben.

## Knoten „Code Generation“

---

Dieser Knoten enthält folgende Optionen:

- *Protected Division* sollte aktiviert werden, so dass der generierte Code Kontrollen auf Divisionen durch Null durchführen und diese abstellen kann. Diese Option stellt einen Systemnachteil dar und sollte deaktiviert werden, wenn eine maximale Codeeffizienz erreicht werden soll.
- *Generate Define Directives for Enum Values*: Wenn die Option deaktiviert ist (Standard), werden im Code für die Enumeratoren die zugehörigen Integerwerte generiert.

Wenn die Option aktiviert ist, erscheinen im generierten Code die Namen der Enumeratoren (z.B. *Red* oder *Green*), und zusätzlich werden die Makros der folgenden Art generiert:

```
#define Red 0
#define Green 1
```

So werden die symbolischen Namen den jeweiligen Integerwerten zugewiesen.

- *Add Comment with Implementation Information for each Assignment Statement*: Wenn die Option aktiviert ist, wird im generierten Code vor jeder Zuweisung und jeder Return-Anweisung ein Kommentar generiert. Dieser Kommentar enthält Implementierungsinformationen über die rechte Seite der Zuweisung.

Beispiel:

```
/* return with expr from doAddition:           ↵
   min=-65536, max=65534, hex=1phys+0,       ↵
   limit=(maxBitLength: true, assign: true)*/
return ((sint32)input1 + input2);
```

- *Add Comment with Specification Source for each Statement*: Wenn die Option aktiviert ist, wird im generierten Code vor jeder generierten Anweisung ein Kommentar generiert, der die Herkunft der Anweisung enthält.
  - ESDL: Zeilennummer
  - Blockdiagramm: Nummer des Sequenzaufrufs
  - Zustandsautomat: Name der Aktion/Bedingung und zugehöriger Zustand/Übergang

Beispiel (ESDL):

```
/* doAddition: line #1 */
self->cont->val = input1 + input2;
```

- *Add Comment with Generation Information for each Component:*  
Wenn die Option aktiviert ist, werden im generierten Code jeder Komponente Kommentare angelegt. Der erste Kommentar enthält Informationen zur Komponente und zu ASCET, der andere enthält alle Build- und Experimental- oder Produktionscode-Optionen.

Beispiel:

```

/*****
 * BEGIN: Generation Information
 * -----
 * Component:.....Project
 * Name:....."ControllerTest"
 * Implementation:....."Impl"
 * Dataset:....."Data"
 * Specification:.....<not applicable>
 * Version:.....<empty String>
 * -----
 * Generation Date:.....21.03.2006
 * Generation Time:.....16:34:44
 * -----
 * ASCET-MD Version:.....V5.1.4
 * ASCET-RP Version:.....V5.4.1
 * ASCET-SE Version:.....V5.2.1 CID[514]
 * -----
 * END: Generation Information
 *****/
/*****
 * BEGIN: Project Options "Build"/"Code"
 * -----
 *      Build
 * -----
 * Code Generator:.....Implementation Experiment
 * Compiler:.....Borland-C V4.5
 * Operating System:....<not applicable>
 * Target:.....PC
 * -----
 *      Code
 * -----
 * add comment with generation information for ↵
 *      each component [true]:.true
 * ...
 * -----
 *      Code.Experiment

```

```

*-----
* data logging [false]:.....false
...
*-----
* Code.FixedPoint
*-----
* allow double bit size for division numerators ↓
  [true]:.....true
...
*-----
* Code.Optimizations
*-----
* common subexpression elimination [true]:...true
...
*-----
* END: Project Options "Build"/"Code"
*****/

```

- *Force Parentheses for Binary Logical Operators* erzwingt Klammern für binäre logische Operatoren, gemäß MISRA<sup>1</sup>-C:2004 Guidelines for the use of the C language in critical systems.
- *Add parentheses for readability* fügt Klammern für verbesserte Lesbarkeit ein, gemäß MISRA<sup>1</sup>-Regel 12.1.

Wenn die Option deaktiviert ist, wird z.B. folgender Code generiert:

```

x = (a > b) ? a - b : a + b;
x = a + b * c;
if (a + b == c) {.. }

```

Wenn die Option aktiviert ist, wird für dieselben Operationen folgender Code generiert:

```

x = (a > b) ? (a - b) : (a + b);
x = a + (b * c);
if ((a + b) == c) {.. }

```

### Knoten „Integer Arithmetic“

Dieser Knoten enthält folgende Optionen:

- Im Kombikästchen *Arithmetic Service Set* wählen Sie den Satz arithmetischer Dienste (s. Kapitel 4.14) aus, den Sie verwenden wollen. Das Kombikästchen listet alle in der Datei `services.ini` des aktuellen Targets enthaltenen Sätze auf; zusätzlich können Sie mit <None> die

<sup>1</sup>. Motor Industry Software Reliability Association, <http://www.misra.org.uk>



Verwendung arithmetischer Dienste abstellen.

Mit der Schaltfläche **Edit** öffnen Sie den Schnittstelleneditor für Arithmetische Dienste; siehe Kapitel 4.14.5.

- Im Kombikästchen *Result on Division by Zero* wählen Sie das Verhalten bei Division durch Null. Wenn Sie `numerator` wählen (Standard), wird der Zähler als Ergebnis zurückgegeben, bei `user defined` wird ein Aufruf des benutzerdefinierten C-Makros `protDiv0(num, result_type_max, result_type_min)` zurückgegeben. Dieses Makro wird in einer der eingebundenen Header-Dateien spezifiziert.

### **Hinweis**

*Stellen Sie sicher, dass das Makro `protDiv0` existiert und korrekt ist, wenn Sie die Option `user defined` verwenden wollen. Eine automatische Prüfung der Existenz oder Syntax findet **nicht** statt.*

Die Argumente des Makros haben folgende Bedeutung:

- `num` ist der aktuelle Wert des Zählers,
- `num_max` ist der maximal mögliche Wert des Ergebnisdatentyps,
- `num_min` ist der minimal mögliche Wert des Ergebnisdatentyps.

Das zweite und dritte Argument werden verwendet, um – in dieser Reihenfolge – das Minimum und Maximum des Ergebnisdatentyps in der Berechnung zu verwenden. Möglich wäre z.B. die Bereitstellung eines Makros

```
#define protDiv0(num,max,min) (num<0?min:max),
```

um abhängig vom Ergebnisdatentyp den größt- bzw. kleinstmöglichen Integerwert zu erhalten.

- *Maximum bit Length (int)* bestimmt, ob der generierte Code für Integervariablen in den Ausdrücken für die Festkomma-Arithmetik 8, 16 oder 32 Bit verwendet.
- *Maximum bit Length (float)* bestimmt, ob der generierte Code für Gleitkommavariablen in den Ausdrücken für die Festkomma-Arithmetik 32 oder 64 Bit verwendet.
- *Allow Limit Service for Assignment Limitation* legt fest, unter welchen Bedingungen limitierende arithmetische Dienste verwendet werden. Ist der Schalter deaktiviert, werden limitierende Dienste nur eingesetzt, wenn ein Überlauf der Prozessorbitbreite verhindert werden muss. Ist der Schalter aktiviert, so werden limitierende Dienste auch an Datentypgrenzen mit einer geringeren Bitbreite, z.B. 16 Bit für einen 32 Bit-Prozessor, verwendet.

Im ersten Fall wird nur ein Überlauf des maximal möglichen Prozessortyps mittels limitierender Dienstfunktion abgefangen, im zweiten Fall auch ein Überlauf des jeweils implementierten Datentyps.

Für Projekte aus früheren ASCET-Versionen wird diese Option bei der Konvertierung automatisch deaktiviert.

- *Generate Limiters (may be changed locally)* erzeugt Begrenzungen für Zuweisungsoperationen, wenn die Gefahr einer Überschreitung des Wertebereichs besteht und Begrenzungen lokal für die Zielvariable aktiviert wurden. Wird die Option deaktiviert, so werden generell keine Begrenzungen generiert (Standard: aktiviert).  
Wird die Option aktiviert, gelten die lokalen Einstellungen an den Elementen.
- *Allow Double bit Size for Division Numerators* lässt das Doppelte der unter *maximum bit length* eingestellten Bitbreite für Zwischenergebnisse einer Multiplikation mit darauffolgender Division zu (Standard: aktiviert).
- *Use SHIFT Operation on Signed Values instead of DIV Operation* führt für vorzeichenbehaftete Zahlen anstelle der Division eine Rechtsverschiebung aus (Standard: aktiviert).  
Einige Compiler verwenden eine logische anstatt einer arithmetischen Rechtsverschiebung. Dies kann zu Vorzeichenfehlern führen, wenn die Option nicht deaktiviert wird.
- *Use SHIFT Operation on Signed Values instead of MUL Operation* führt für vorzeichenbehaftete Zahlen anstelle der Multiplikation eine Linksverschiebung aus (Standard: aktiviert).  
Einige Compiler verwenden eine logische anstatt einer arithmetischen Linksverschiebung. Dies kann zu Vorzeichenfehlern führen, wenn die Option nicht deaktiviert wird.
- *Generate Round Operation on float to integer Assignment* schaltet die Rundung bei der Zuweisung von Gleitkommavariablen zu Integervariablen ein (aktiviert) oder aus (deaktiviert). Die Rundung erfolgt durch Addition (positive Zahlen) bzw. Subtraktion (negative Zahlen) von 0.5. Ist die Option deaktiviert, werden die Nachkommastellen bei der Zuweisung abgeschnitten (Standard: aktiviert).
- *Temp Vars always 32 bit (integer)* erzeugt für alle temporären Integervariablen 32-Bit-Größen. Beim Deaktivieren dieser Option werden die Bitbreiten variabel nach Bedarf generiert (Standard: deaktiviert).
- *Use power of 2 approximations of literals*: Wenn die Option aktiviert ist, sucht der Codegenerator bei Multiplikationen und Divisionen mit Brüchen für Literale einen anderen Bruch als Näherung, bei dem Zähler oder Nenner eine Potenz von 2 ist.

Die Optimierung findet nur statt, wenn eine Näherung gefunden wird, die maximal 1 % vom ursprünglichen Wert abweicht, und wenn durch das genäherte Literal kein Überlauf verursacht wird.

Ein Beispiel: Bei deaktivierter Option (Standard) wird folgender Code generiert,

```
result = input * (sint32)433 / (sint16)500;
```

bei aktivierter Option wird daraus

```
result = input * 28377 >> 15;
```

Die Genauigkeit ist in beiden Fällen nahezu identisch, aber die lauffzeit-intensive Division ist durch eine effiziente und schnelle Verschiebung ersetzt.

*Knoten „Experiment Code“*

---

### **Hinweis**

*Diese Optionen sind für Mikrocontroller-Targets nicht verfügbar.*

Die folgenden Experimentalcode-Optionen stehen zur Verfügung:

- *Max Number of Loop Iterations* bestimmt die maximale Anzahl von Wiederholungen, die der generierte Code in einer Schleife ausführt. Diese Option dient dazu, ein Hängenbleiben in unendlichen Schleifen zu verhindern.
- *Prefix for Component Names* fügt eine Zeichenkette als Präfix für die Namen der Komponenten im generierten Code ein. Diese Option kann hilfreich sein, wenn Sie mehrere Versionen einer Komponente in INTECRIO verwenden wollen; das geht nicht, wenn alle Versionen identische Namen haben.
- *Data Logging* muss aktiviert werden, wenn Sie den Data-Logger im Transient-Modus nutzen wollen. Der Data-Logger wird im Abschnitt „Der Data-Logger“ auf Seite 641 beschrieben.
- *Protected Vector Indices* sollte aktiviert werden, so dass der generierte Code kontrolliert, ob Versuche unternommen werden, auf einen Array oder eine Tabelle mit einem Index außerhalb des Bereichs zuzugreifen. Wenn zum Beispiel ein Array fünf Elemente hat, werden die Fehler, die aus einem Zugriff auf das sechste Element dieses Arrays resultieren, abgefangen. Diese Option beinhaltet einen Systemnachteil und sollte deaktiviert werden, wenn eine maximale Codeeffizienz erreicht werden soll.

- *Use OID for Generation of Component Names* legt fest, ob die OID der Komponente (aktiviert) oder der Modellname (deaktiviert) im generierten Code als Name der Komponente verwendet wird. Wenn die Option deaktiviert ist, ist es die Verantwortung des Benutzers, Namenskonflikte zu vermeiden.
- *Add Comment with Specification Info for each Element*: Wenn die Option aktiviert ist, wird im generierten Code für jedes Element ein Kommentar mit Informationen zur Spezifikation generiert.

Beispiel:

```

/*----Local variables object structure-----*/
struct MODULE_C_CODE_IMPL_Obj {
    ASDObjectHeader objectHeader;
    /* model: name=cont, type=cont,           ↵
               kind=variable, scope=local,   ↵
               memory=volatile */
    real64_Obj *cont;
};

```

- *Cache Locking, Cache Lock Code, Cache Lock Data, Used Cache Size*: Cache Locking ist nur für die Arbeit mit ASCET-RP von Bedeutung. Die zugehörigen Optionen sind im ASCET-RP-Handbuch beschrieben.

### Knoten „Production Code“

Die Optionen in diesem Knoten sind nur verfügbar, wenn ASCET-SE installiert ist und in den Build-Optionen (s. Seite 443) ein Mikrocontroller-Target ausgewählt wurde. Sie sind daher im ASCET-SE-Benutzerhandbuch beschrieben.

### Knoten „Optimization“

Dieser Knoten enthält folgende Optimierungsoptionen:

- *Constant Folding on Arithmetic Operations* ersetzt arithmetische Operationen mit konstantem Ergebnis durch das Ergebnis. Beispielsweise wird die Addition zweier Konstanten,  $C1 + C2$ , durch die Konstante  $C3$  ( $C3 := C1 + C2$ ) ersetzt; der Vergleich zweier Konstanten wird durch das Ergebnis ersetzt, z.B. wird  $C1 >= C2$  durch `TRUE` (wenn  $C1 >= C2$ ) oder `FALSE` (wenn  $C1 < C2$ ) ersetzt.
- *Constant Folding on Logical Operations* ersetzt logische Operationen, bei denen das Ergebnis feststeht, durch das Ergebnis. Beispielsweise wird `X_log AND TRUE` durch `X_log` ersetzt, `FALSE OR X_log` durch `X_log` und `FALSE != FALSE` durch `FALSE`.

- *Constant Folding on Control Flow* entscheidet, ob ein Ausdruck, bei dem direkt an der `if`-Anweisung entschieden werden kann, dass der Ausdruck immer `true` oder immer `false` ist, im generierten Code optimiert wird (aktiviert; Standard) oder nicht (deaktiviert).
- *Operator Simplification with Respect to Intervals* vereinfacht Ausdrücke, bei denen aufgrund der Intervallberechnung bereits das Ergebnis feststeht. Wenn beispielsweise `x` das Intervall `[ 0 , 100 ]` hat und `y` das Intervall `[ 120 , 150 ]`, wird der Ausdruck `x < y` zu `TRUE`, weil `y` immer größer als `x` ist.
- *Operator Simplification (general)* vereinfacht Operationen. Beispielsweise wird `x - NEG(y)` durch `x + y` ersetzt, `NOT(NOT(x_log))` durch `x_log`.
- *Common Subexpression Elimination* ersetzt Operationen mit gleichen Teilausdrücken, die vereinfacht werden können. Beispielsweise wird `x / x` durch `1` ersetzt, `x < x` durch `FALSE` und `x_log AND x_log` durch `x_log`.
- *Optimize Direct Access Methods (One Level)* sollte aktiviert werden, um Direktzugriffsmethoden im generierten Code zu aktivieren. Andernfalls wird für jeden Aufruf einer Direktzugriffsmethode ein Funktionsaufruf erzeugt. Diese Option sollte bei Auftreten von Problemen bei der Codegenerierung deaktiviert werden.

### Hinweis

Für Klassen, die als Methodenargumente spezifiziert werden, wird **Optimize Direct Access** auf deren Elemente nicht unterstützt.

- *Optimize Direct Access Methods (Multiple Levels)* sollte aktiviert werden, um verschachtelte Direktzugriffsmethoden im generierten Code zu aktivieren. Die Verschachtelung von direkten Zugriffsmethoden ist nur möglich bei der Spezifikation von Komponenten in ESDL. Diese Option hat die gleiche Wirkung wie bei der Optimierung von einfachen Methodenaufrufen.

### Knoten „Statemachine“

Dieser Knoten enthält spezielle Optimierungsoptionen für Zustandsautomaten.

- *Outline Generated Methods (may be changed locally)*: Wird für verschiedene Übergänge aus dem selben Hierarchiezustand heraus identischer Code benötigt, so wird für diesen unter bestimmten Bedingungen eine separate Methode erzeugt (Outlining). Diese Methode wird dann an jeder erforderlichen Stelle aufgerufen.

Wenn Sie nicht möchten, dass separate Methoden und Methodenaufrufe erzeugt werden, deaktivieren Sie die Option. In diesem Fall wird der Code an jeder erforderlichen Stelle direkt eingefügt.

- *Auto-inline private methods (Smaller code-size - may be changed locally)*: Wenn kleine private Funktionen (oder Funktionen, die nur an wenigen Stellen aufgerufen werden) an jeder aufrufenden Stelle in den generierten Code eingefügt (Inlining) werden, spart dies Codegröße und Laufzeit.

Wenn die Option aktiviert ist, identifiziert der Codegenerator solche Funktionen automatisch und sorgt für deren Inlining.

- *Hierarchical Coode-Generation (may be changed locally)*: Code für Zustandsautomaten kann wahlweise flach (eine `switch`-Anweisung mit einem `case`-Statement für jeden Basiszustand) oder hierarchisch (verschachtelte `switch`-Anweisungen entsprechend der Hierarchie der Zustände) generiert werden. Flache Codegenerierung (deaktiviert) optimiert für Laufzeit, hierarchische Codegenerierung (aktiviert) optimiert für Codegröße.

- *Optimize Static Actions (Restricted Modeling)*: Wenn die Option deaktiviert ist (Standard), wird Code für die statische Aktion eines Hierarchiezustands für jeden Übergang, der den Hierarchiezustand nicht verlässt, neu generiert.

Wenn die Option aktiviert ist, wird Code für die statische Aktion eines Hierarchiezustands für jeden Unterzustand nur einmal generiert. Auf diese Weise wird die Codegröße reduziert.

### **Hinweis**

*Die Optimierung mit Hilfe dieser Option verändert die Reihenfolge, in der Aktionen ausgeführt und Bedingungen geprüft werden – und damit möglicherweise das Verhalten des Zustandsautomaten. Außerdem ist diese Optimierung nicht für alle Modelle möglich.*

*Eine detaillierte Beschreibung finden Sie im Kapitel „Den Zustandsautomaten optimieren“ im ASCET-Referenzhandbuch.*

- *Generate well-formed switch*: Wenn die Option aktiviert ist, wird der Code für den Zustandsautomaten so generiert, dass er den MISRA<sup>1</sup>-Regeln 14.7, 15.2 und 15.3 entspricht.
- *Initialize history variable with zero*: Wenn die Option aktiviert ist, werden im Projekt vorhandene History-Variablen von Hierarchiezuständen nicht mit deren jeweiligem Startzustand initialisiert, sondern mit 0.  
Dies bedeutet eine zusätzliche Zuweisung im generierten Code und kann die MISRA-Regel 15.3 verletzen.

#### 4.8.9 Festlegen der Implementierung für die Festkomma-Arithmetik

---

Bei der Generierung des Produktionscodes für Mikrocontroller muss oft die Arithmetik der physikalischen ASCET Spezifikation auf eine Festkomma-Arithmetik abgebildet werden, da viele in elektronischen Steuergeräten verwendeten Mikrocontroller die Gleitkomma-Arithmetik nicht unterstützen. Dieses Abbilden wird durch die Implementierungstransformation oder kurz Implementierung beschrieben.

Wie Datensätze kann jede Komponente oder jedes Projekt eine Reihe von Implementierungen umfassen. Implementierungen können wie Datensätze betrachtet, bearbeitet, umbenannt, gelöscht, durchgeblättert, exportiert (sowohl einfach als auch rekursiv), hinzugefügt und kopiert werden.

Wie bei Datensätzen gibt es spezielle Editoren für die Basiselemente, die die Blätter der hierarchischen Baumstruktur eines Projekts oder einer Komponente bilden. Die Implementierung der Basiselemente der Typen `continuous` oder `discrete` erfordert die Spezifikation eines Bereichs der physikalischen Werte und eine Umwandlungsformel.

Viele Elemente innerhalb eines Projekts haben gewöhnlich die gleiche Umwandlungsformel, wie zum Beispiel alle Geschwindigkeiten, die in der eingebetteten Software verarbeitet werden. Es ist möglich, Formeln nur einmal innerhalb eines Projekts zu definieren und sie dann auf alle Elemente in diesem Projekt anzuwenden. Formeln können auch mit einem Import/Export-Mechanismus zwischen Projekten ausgetauscht werden.

ASCET unterstützt auch den Zwischenschritt der Generierung des quantisierten Gleitkomma-Codes. Mit diesem Codetyp wird eine Festkomma-Arithmetik emuliert, wobei jedoch Verknüpfungen und die Quantisierung der Festkomma-Arithmetik interaktiv während der Ausführung des Codes verändert werden können. Mit diesem Zwischenschritt kann die optimale Implementierung für Elemente in einem spezifischen Projekt entwickelt werden.

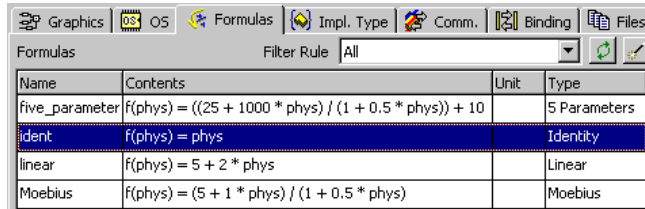
---

<sup>1</sup>. Motor Industry Software Reliability Association, <http://www.misra.org.uk>

## Formeln

### Hinzufügen einer Formel:

- Klicken Sie im Projekteditor auf das Register „Formulas“.



Name	Contents	Unit	Type
five_parameter	$f(\text{phys}) = ((25 + 1000 * \text{phys}) / (1 + 0.5 * \text{phys})) + 10$		5 Parameters
ident	$f(\text{phys}) = \text{phys}$		Identity
linear	$f(\text{phys}) = 5 + 2 * \text{phys}$		Linear
Moebius	$f(\text{phys}) = (5 + 1 * \text{phys}) / (1 + 0.5 * \text{phys})$		Moebius

Die Formel `ident` ist immer vorhanden. Sie ist in den Implementierungen neu angelegter Elemente standardmäßig ausgewählt.

- Wählen Sie **Global Formulas** → **Add**, um eine neue Formel zu erstellen.
- Geben Sie den Namen für die Formel ein und drücken Sie <ENTER>.

### Bearbeiten der Formel:

- Wählen Sie im Register „Formulas“ die Formel, die Sie bearbeiten wollen.
- Wählen Sie **Global Formulas** → **Edit**  
*oder*
- wählen Sie aus dem Kontextmenü **Edit**  
*oder*



- doppelklicken Sie auf die Formel.  
Der Formeleditor für globale Formeln öffnet sich.

The screenshot shows a dialog box titled "Edit Formula: default". It has a "Type" dropdown menu currently set to "Linear". To the right of the dropdown is an empty "Unit" text field. Below these is a formula editor showing the equation  $f(\text{phys}) = 0 + 1 * \text{phys}$ , where "0" and "1" are in input boxes and "c0" and "c1" are labels above them. At the bottom is a "Comment" text area. On the right side of the dialog are "Ok" and "Cancel" buttons.

- Wählen Sie im Kombikästchen „Type“ den zu bearbeitenden Formeltyp.
- Geben Sie die erforderlichen Werte in die entsprechenden Felder ein.
- Geben Sie eine Einheit in das Feld „Unit“ ein. Die Einheit dient ausschließlich Dokumentationszwecken und hat keinen Einfluss auf die Funktionalität.
- Geben Sie einen Kommentar in das Feld „Comment“ ein.
- Klicken Sie auf **OK**.

Die Formel `<ident>` sollte nicht umbenannt oder gelöscht werden. Wenn Sie die Formel bearbeiten und im Kombikästchen „Type“ einen anderen Typ auswählen, erscheint folgende Warnung:

The `<ident>` formula is used for newly created elements to provide an identity conversion and changing this formula may corrupt your implementation. Do you really want to change the `<ident>` formula?

Wenn Sie den Typ der Identität tatsächlich ändern wollen, klicken Sie **Yes**, andernfalls klicken Sie **No**.

Die Umwandlungsformel für die Implementierung muss eine lineare Formel sein. Zur Darstellung dieser Formel werden vier Typen von Konvertierungsformeln unterstützt. Der Bildschirm des Formeleditors passt sich an, um den ausgewählten Formeltyp wiederzugeben.

- Die *Identity*-Formel verkörpert ein Identitätsabbilden.

$$f(\text{phys}) = \text{phys}$$

- Die *Linear*-Formel verkörpert eine lineare Abbildung mit den Koeffizienten  $c_0$  (Anpassung) und  $c_1$  (Gradient).

$$f(\text{phys}) = \overset{c_0}{0} + \overset{c_1}{1} * \text{phys}$$

- Die *Moebius*-Formel verkörpert eine rationale Abbildung als Quotient von zwei linearen Abbildungen mit den Koeffizienten  $c_0$  (Offset des Zählers),  $c_1$  (Gradient des Zählers),  $d_0$  (Offset des Nenners), und  $d_1$  (Gradient des Nenners). Die Moebius-Formel wird hauptsächlich verwendet, um eine Konvertierung in das lineare Formelformat zu vermeiden, wenn die Spezifikation einer linearen Formel im Moebius-Formelformat gegeben ist.

$$f(\text{phys}) = \frac{\overset{c_0}{0} + \overset{c_1}{1} * \text{phys}}{\underset{d_0}{1} + \underset{d_1}{0} * \text{phys}}$$

- Die *Five Parameters*-Formel verkörpert ebenfalls eine rationale Abbildung gleicher Art in verschiedenem Format mit den Koeffizienten  $p_1$ ,  $p_2$ ,  $p_3$ ,  $p_4$  und  $p_5$ . Diese Formel wird ebenfalls verwendet, um eine Konvertierung in das lineare Format zu vermeiden, wenn die Spezifikation einer linearen Formel im Five-Parameter-Formelformat gegeben ist.

$$f(\text{phys}) = \frac{\overset{p_2}{0} + \overset{p_1}{1} * \text{phys}}{\underset{p_4}{1} + \underset{p_3}{0} * \text{phys}} + \overset{p_5}{0}$$

### Umbenennen einer Formel:

---

- Wählen Sie im Register „Formulas“ eine Formel.
- Wählen Sie **Global Formulas** → **Rename**  
*oder*
- wählen Sie aus dem Kontextmenü **Rename**, um die Formel umbenennen zu können.

### Löschen einer Formel:

---

- Wählen Sie im Register „Formulas“ eine oder mehrere Formeln.
- Wählen Sie **Global Formulas** → **Delete**  
*oder*
- wählen Sie aus dem Kontextmenü **Delete**, um die Formeln zu löschen.
- Bestätigen Sie den Befehl im Fenster „Confirm“ mit **OK**.

Die Formeln werden gelöscht. Im ASCET-Monitorfenster erscheint eine Liste der gelöschten Formeln.

### Anzeige sortieren:

---

- Klicken Sie auf den Namen einer Spalte.  
Die Anzeige wird nach der betreffenden Spalte sortiert. Die Spalten werden alphabetisch sortiert, ausgenommen die Spalte „Type“. In dieser Spalte gilt folgende Sortierreihenfolge:  
Identity → Linear → Moebius → 5 Parameters  
Abermaliges Sortieren kehrt die Reihenfolge um.

## Anzeige filtern:

---

Sie können die Anzeige im Register „Formulas“ unter verschiedenen Gesichtspunkten filtern.

- Wählen Sie im Kombikästchen „Filter Rule“ eine Filterregel.

Folgende Filter stehen zur Verfügung:

Filter	angezeigte Formeln
All	alle Formeln
Used in ACTIVE implementations	für die Implementierung eines Elements im Projekt verwendete Formeln
Not used in ACTIVE implementations	<i>nicht</i> für die Implementierung eines Elements im Projekt verwendete Formeln
Used in ANY implementation	für eine Element-Implementierung oder in einer derzeit unbenutzten Implementierung verwendete Formeln
Not used	unbenutzte Formeln

Wenn Sie z.B. der Filter **Not used** wählen, können Sie unbenutzte Formeln unkompliziert löschen.



- Klicken Sie auf **Update**, um die Anzeige zu aktualisieren.

Das ist erforderlich, wenn Sie die Implementierung des Projekts oder einer darin enthaltenen Komponente in einem anderen Editor bearbeitet haben.

## Fehlende Formeln hinzufügen:

---

Wenn Elemente des Projekts Formeln verwenden, die im Projekt nicht definiert sind, werden bei der Codegenerierung Fehlermeldungen ausgegeben. Um die Formeln hinzuzufügen, gehen Sie wie folgt vor.

- Wählen Sie im Projekteditor, Register „Formulas“, **Global Formulas** → **Add missing**  
*oder*



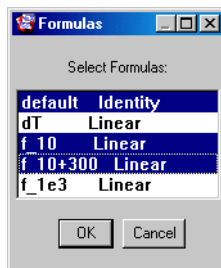
- klicken Sie auf **Add missing formulas**.  
Für jede undefinierte Formel in der aktiven Implementierung wird eine lineare Formel mit entsprechendem Namen und dem Verhalten der Identität hinzugefügt.

### Importieren von Formeln:

---

- Klicken Sie im Projekteditor auf das Register „Formulas“.
- Wählen Sie **Global Formulas** → **File In** → **Select** oder **All**.  
Das Dateiauswahfenster von Windows wird geöffnet.
- Wählen Sie die Datei aus, die die Formeln enthält, die Sie einfügen wollen.
- Klicken Sie auf **Öffnen**.

Wenn Sie den Befehl **All** ausgewählt haben, werden alle Formeln in der ausgewählten Datei importiert. Anderenfalls können Sie aus einem Dialogfeld auswählen, welche Formeln importiert werden sollen.



### Auslagern von Formeln:

---

- Wählen Sie **Global Formulas** → **File Out** → **Selected** oder **All**.  
Das Dateiauswahfenster von Windows wird geöffnet.
- Wählen Sie ein Verzeichnis und einen Dateinamen aus.
- Klicken Sie auf **Speichern**.

Wenn Sie den Befehl **All** ausgewählt haben, werden alle Formeln im aktuellen Projekt ausgelagert. Anderenfalls wird nur die ausgewählte Formel exportiert. Formeln werden im ASCII-Format ausgelagert, das mit jedem Texteditor gelesen werden kann.

Ein Array hat eine Implementierung, die wie die Implementierung eines skalaren Elements definiert ist. Eine 1D-Tabelle hat zwei Implementierungen, eine für den Eingabe- und eine für den Ausgabewert. Eine 2D-Tabelle hat drei Implementierungen, zwei für die Eingabewerte und eine für den Ausgabewert.

### *Globale Veränderungen in Implementierungen*

---

Wurde eine Formel bearbeitet oder eine vorhandene Formel durch eine andere Formel ersetzt, können Sie nun Ihre Implementierungen rekursiv aktualisieren. Sie können auch eine Formel für alle Implementierungen eines Projekts ersetzen, um Ihre Implementierungseinstellungen anzupassen.

Änderungen werden nicht automatisch verbreitet; Sie müssen Ihre Implementierungen immer aktualisieren, wenn eine Formel entweder ersetzt oder modifiziert wurde.

#### **Rekursives Ersetzen einer Formel:**

---

- Aktivieren Sie im Projekteditor das Register „Formulas“, so dass Sie ein Verzeichnis der vorhandenen Formeln sehen können.
- Wählen Sie **Extras** → **Global Replace Formula** aus.  
Ein Dialogfenster fordert Sie auf, den Namen der alten und der neuen Formel einzugeben.
- Geben Sie die Namen nach Bedarf ein und klicken Sie auf **OK**.  
Die Formel wird rekursiv in der gesamten Implementierung ersetzt.

#### **Aktualisieren aller Implementierungen:**

---

- Wählen Sie im Projekteditor **Extras** → **Update Implementations**, um Änderungen in den Formeln in allen Implementierungen vorzunehmen.  
Im ASCET Monitorfenster wird ein Protokoll der Änderungen geschrieben. Änderungen in den Implementierungen werden gemäß der

im Implementierungsektor jedes primitiven Elements ausgewählten Masterseite vorgenommen.

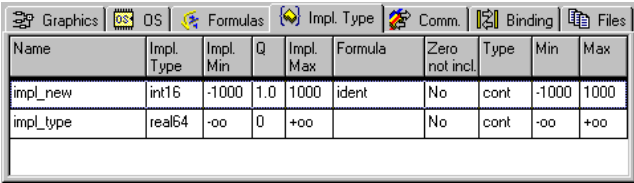
### Implementierungstypen

Um die Implementierungen einzelner Größen leichter bearbeiten zu können sowie um Elementen mit vergleichbarer physikalischer Bedeutung auf einfache Weise gleiche Implementierungen zuweisen zu können, können Sie im Projektkontext sogenannte *Implementierungstypen* definieren. Das gilt auch für das Standardprojekt (siehe Kapitel 4.8.1) einer Klasse oder eines Moduls. Diese Implementierungstypen beinhalten die wesentlichen Spezifikationen einer Implementierung und können einzelnen Elementen in deren Implementierungsektoren zugewiesen werden.

In diesem Abschnitt wird beschrieben, wie Sie Implementierungstypen anlegen und einrichten. Wie diese dann bei der Implementierung zum Einsatz kommen, ist im Abschnitt „Implementierungstypen verwenden“ auf Seite 531 beschrieben.

#### Hinzufügen eines Implementierungstyps:

- Klicken Sie im Projekteditor auf das Register „Impl. Type“.



Name	Impl. Type	Impl. Min	Q	Impl. Max	Formula	Zero not incl.	Type	Min	Max
impl_new	int16	-1000	1.0	1000	ident	No	cont	-1000	1000
impl_type	real64	-oo	0	+oo		No	cont	-oo	+oo

- Wählen Sie **Global Implementations** → **Add** → *<model type>*

oder

- wählen Sie aus dem Kontextmenü **Add** → **<model type>**.

### Hinweis

Der Modelltyp, den Sie bei der Erzeugung eines Implementierungstyps wählen, legt fest, welchen Größen Sie diesen Implementierungstyp zuweisen können (siehe „Implementierungstypen verwenden“ auf Seite 531).

Ein neuer Implementierungstyp wird mit einem Standardnamen angelegt. Entsprechend Ihrer Auswahl für **<model type>** werden automatische Voreinstellungen (siehe nachstehende Abbildung) ausgewählt.

Name	Impl. Type	Impl. Min	Q	Impl. Max	Formula	Zero not incl.	Type	Min	Max
impl_type_cont	real64	-oo	0	+oo	ident	No	cont	-oo	+oo
impl_type_sdisc	int32	-2147483648	1	2147483647	ident	No	sdisc	-2147483648	2147483647
impl_type_udisc	uint32	0	1	4294967295	ident	No	udisc	0	4294967295

### Verwalten von Implementierungstypen:

Sie können mehrere Implementierungstypen gleichzeitig löschen, aber nur einen umbenennen.

- Wählen Sie einen Implementierungstyp aus.
- Wählen Sie **Global Implementations** → **Rename**  
oder
- wählen Sie aus dem Kontextmenü **Rename**.  
Die entsprechende Zelle in der Spalte „Name“ wird zum Eingabefeld.
- Geben Sie den Namen für den Implementierungstyp ein und drücken Sie <ENTER>.
- Wählen Sie **Global Implementations** → **Delete**  
oder



- wählen Sie aus dem Kontextmenü **Delete**.  
Der Implementierungstyp wird gelöscht.

### Bearbeiten von Implementierungstypen:

- Wählen Sie im Register „Impl. Type“ den Implementierungstyp, den Sie bearbeiten wollen.
- Wählen Sie **Global Implementations** → **Edit** oder
- wählen Sie aus dem Kontextmenü **Edit** oder
- doppelklicken Sie auf den Implementierungstyp.

Der Implementierungseditor für Implementierungstypen öffnet sich im Register „Value“.

The screenshot shows the 'Implementation for: impl\_type' dialog box. It has two tabs: 'Value' and 'Comment'. The 'Value' tab is active. The dialog is divided into several sections:

- Transformation:**
  - Formula:
  - Conversion:  $f(\text{phys}) = \text{phys}$
  - Quantization: Calculated  Qu. Exp.
- Master:**
  - Model  Implementation
- Model:**
  - Type:
  - Min:
  - Max:
- Implementation:**
  - Type:
  - Min:
  - Max:
  - Zero not included
- Consistency:**

Source	Conflict

At the bottom, there are buttons for 'Auto Correction', 'OK', and 'Cancel'.

- Wählen Sie im Kombikästchen „Formula“ eine Umrechnungsformel.

- Geben Sie ggf. im Feld „Qu.Exp“ die Quantisierung ein.

Das Feld ist nur aktiviert, wenn in den Build-Optionen (s. Seite 443) der Codegenerator `Quantized Physical Experiment` gewählt ist. In allen anderen Fällen dient es lediglich der Anzeige.

- Wählen Sie im Feld „Master“ die Masterseite der Implementierung.

### 1. Master **Model**:

Diese Einstellung sollte gewählt werden, wenn physikalische Modelleigenschaften ausschlaggebend für die Wahl der Implementierung sind.

Die Felder im Bereich „Model“ können bearbeitet werden, ausgenommen der Modelltyp. Dieser wird beim Anlegen des Implementierungstyps festgelegt.

Der Bereich „Implementation“ ist bis auf die Option **Zero not included** gesperrt.

Model		Default Value
Type	cont	Copy Paste
Min	-214	$2^{17}-1$ (127)
Max	214	$2^{18}-1$ (255)

- Geben Sie in den Feldern „Min.“ und „Max.“ die Grenzen für das Intervall an.
- Um die obere oder untere Grenze eines Modelltyps („unendlich“ im Fall von „cont“) einzustellen, klicken Sie mit der rechten Maustaste in eins der Felder und wählen Sie aus dem Kontextmenü **Default Value**.

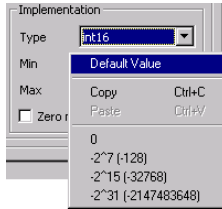
Die Werte für die Implementierung werden automatisch aktualisiert.

### 2. Master **Implementation**:

Diese Einstellung sollte gewählt werden, wenn bestimmte Code-Eigenschaften (z.B. eine gewünschte Bitbreite) ausschlaggebend für die Wahl der Implementierung sind.

Die Felder im Bereich „Implementation“ können bearbeitet werden, der Bereich „Model“ ist gesperrt.

- Wählen Sie im Kombikästchen „Type“ den Typ der Implementierung.
- Geben Sie in den Feldern „Min.“ und „Max.“ die Grenzen für das Intervall an.



- Um die Standardgrenzen für den aktuell gewählten Datentyp einzustellen, klicken Sie mit der rechten Maustaste in eins der Felder und wählen Sie aus dem Kontextmenü **Default Value**.

Die Werte für das Modell werden automatisch aktualisiert.

### 3. sonstige Einstellungen:

- Wenn das Intervall den Wert „null“ *nicht* annehmen kann, aktivieren Sie die Option **Zero not included**.

#### Hinweis

Wenn Sie die Option **Zero not included** aktiviert haben, wird im generierten Code zur Laufzeit keine Überprüfung vorgenommen, ob der Nenner einer Division null ist. Dies kann zu sicherheitskritischem Laufzeitverhalten des Codes führen. Im Zweifel muss die Option daher deaktiviert werden.

- Wählen Sie das Register „Comment“, wenn Sie einen Kommentar eingeben wollen.
- Geben Sie im Textfeld des Registers den Kommentar ein.  
Dieser Kommentar wird bei der Zuweisung zu einem Element *nicht* mit übertragen.
- Klicken Sie **OK**, um Ihre Einstellungen zu übernehmen.

Sie können für einen Implementierungstyp keine Einstellungen für Speicherbereich und Limitierungsverhalten bei Zuweisungen oder Überlauf vornehmen oder zusätzliche Informationen eingeben. Diese Einstellungen werden individuell für jedes Element vorgenommen.

#### Einstellungen von Implementierungstypen kopieren:

Wenn Sie einen Implementierungstyp eingerichtet haben, können sie die Einstellungen in einen anderen Implementierungstyp kopieren.

- Wählen Sie den Implementierungstyp, dessen Einstellungen Sie kopieren wollen.

- Wählen Sie **Global Implementations** → **Copy**

*oder*

- wählen Sie **Copy** aus dem Kontextmenü.  
Die Einstellungen werden in die Zwischenablage der Datenbank kopiert.
- Wählen Sie den Implementierungstyp, in den Sie die Einstellungen kopieren wollen.
- Wählen Sie **Global Implementations** → **Paste**

*oder*

- wählen Sie **Paste** aus dem Kontextmenü.  
Die Einstellungen werden aus der Zwischenablage der Datenbank in den Implementierungstyp kopiert.

### **Einstellungen von Implementierungstypen und skalaren Elementen austauschen:**

---

Darüber hinaus können Sie auch die Einstellungen zwischen Implementierungstypen und skalaren Elementen der Datenbank austauschen.

- Wählen Sie im Projekteditor, Register „Impl. Type“, den Implementierungstyp, dessen Einstellungen Sie kopieren wollen.
- Wählen Sie **Global Implementations** → **Copy**

*oder*

- wählen Sie **Copy** aus dem Kontextmenü.  
Die Einstellungen werden in die Zwischenablage kopiert.

1. über den Komponentenmanager, Register „Implementation“

- Wählen Sie im Komponentenmanager in der Liste „1 Database“ die Komponente, die das skalare Element enthält, in das Sie die Einstellungen kopieren wollen.
- Wählen Sie im Feld „3 Contents“ das Register „Implementation“.

- Wählen Sie im Register „Implementation“ das Element aus.
- Wählen Sie **Paste** aus dem Kontextmenü  
*oder*
- drücken Sie <CTRL> + <V>.

## 2. über den Implementierungseeditor einer Komponente

- Öffnen Sie den Implementierungseeditor der Komponente, die das skalare Element enthält, in das Sie die Einstellungen kopieren wollen.
- Wählen Sie das Register, in dem das Element enthalten ist.
- Wählen Sie das Element aus.
- Wählen Sie **Element** → **Paste Implementation From Buffer**

*oder*

- wählen Sie **Paste Implementation From Buffer** aus dem Kontextmenü.

Die Einstellungen werden aus der Zwischenablage in das skalare Element kopiert.

Wenn Sie die Implementierung eines skalaren Elements in einen Implementierungstyp kopieren wollen, gehen Sie umgekehrt vor: Kopieren Sie die Implementierung im Komponentenmanager oder im Implementierungseeditor und fügen Sie sie im Register „Impl. Type“ des Projekteditors ein.

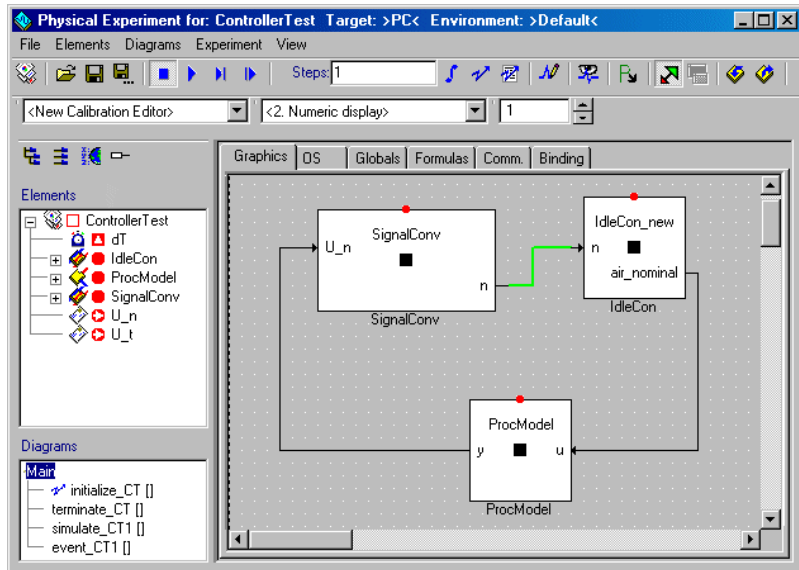
### 4.8.10 Experimentieren mit Projekten

---

Die Möglichkeiten für das Generieren von Code und für das Experimentieren sind für Projekte viel besser als für Komponenten. Für Komponenten steht nur ein Offline-Experimentieren mit Gleitkomma-Arithmetik (Build-Option `Physical Experiment`, s. Seite 443) zur Verfügung. Projekte bieten ein Online-Experimentieren, und Offline- sowie Online-Experimentieren lassen sich mit Gleitkomma-Arithmetik, quantisierter Gleitkomma-Arithmetik (Build-Option `Quantized Physical Experiment`) oder Festkomma-Arithmetik (Build-Option `Implementation Experiment`) kombinieren. Die gleichen Möglichkeiten für das Generieren von Code bestehen für Komponenten, wenn mit diesen im Rahmen eines Projekts experimentiert wird.

## Online- und Offline-Experimentieren

Es besteht die Möglichkeit, mit einem Projekt wie mit einer Komponente offline zu experimentieren. Zum Offline-Experimentieren sind alle Möglichkeiten vorhanden, und sie funktionieren ebenso wie bei Komponenten. Offline-Experimentieren mit einer Komponente gleicht dem Offline-Experimentieren mit einem Projekt, wobei das Standardprojekt die Komponente enthält. Die Offline-Experimentierumgebung ist im Detail im Kapitel „Die Experimentierumgebung“ auf Seite 603 beschrieben.



Der Hauptunterschied zwischen Offline-Experimentieren mit Komponenten und mit Projekten besteht darin, dass bei Projekten die im Betriebssystemeditor generierten Tasks im Ereignisgenerator angeregt werden, und nicht die Methoden oder Prozesse von Komponenten.

Beim Online-Experimentieren lässt sich ein Projekt unter realistischeren Bedingungen und in Echtzeit erproben. Während des Offline-Experimentierens kann der durch ASCET generierte Code auf dem PC oder auf einem experimentellen Target laufen, jedoch läuft er dabei nicht in Echtzeit. Während des Online-Experimentierens läuft der Code immer auf einem experimentellen Target in Echtzeit. Online-Experimentieren bildet die Grundlage für den Einsatz von ASCET in praktischen Anwendungen. Es zielt auf den Betriebssystem-Schedule

und das entsprechende Echtzeitverhalten des Steuersystems ab, während man sich beim Offline-Experimentieren häufig mit der Erprobung der Funktionsspezifikation eines Systems beschäftigt.

Die Hauptunterschiede zwischen Online- und Offline-Experimentieren sind:

- Online-Experimente müssen auf Echtzeit-Hardware laufen, wie z.B. auf den ETAS-Targets ES1130 und ES1135.
- Es erfolgt keine Stimulierung, d.h. es gibt keine Ereignis- und Datengeneratoren. Das Scheduling wird durch das integrierte Betriebssystem bestimmt, und die Daten werden beispielsweise aus dem technischen Prozess eingelesen.
- Die Abtastrate, mit der Daten gemessen werden, muss durch Auswahl einer Aktivierungstask für jedes Messfenster explizit vorgegeben werden.
- Das Experiment und die Messungen können unabhängig voneinander gestartet werden.

Online-Experimente sind nur im Zusammenhang mit ASCET-RP möglich; eine Beschreibung finden Sie im ASCET-RP-Benutzerhandbuch.

Für das Projekt kann Code ohne Starten der Experimentierumgebung generiert werden. Dies ist von Nutzen bei der Prüfung, ob der generierte Code syntaktisch richtig ist. Bei falschem Code zeigt das Programm die vom Compiler ausgegebenen Fehlermeldungen an.

### Generieren von Code:

---



- Wählen Sie im Projekteditor **Component** → **Generate Code**.

*oder*

- klicken Sie auf die Schaltfläche **Generate Code**.

Der Code für das Projekt wird generiert, und eventuelle Fehlermeldungen werden im Monitorfenster angezeigt.

*Oder*

- Wählen Sie **Component** → **View Generated Code**.

C-Code wird generiert und automatisch in einem Texteditor angezeigt.

Der Editor kann im ASCET-Optionsfenster, Knoten „ASCII Editor“ (s. Seite 61), gewählt werden.

## Generierten Code abspeichern:

---

- Wählen Sie **Component** → **File Out Generated Code** → **Flat** oder **Recursive**, um den Code für das Projekt in eine Datei zu schreiben.

Es erscheint das Dialogfenster „Path Selection“ von Windows.

- Wählen Sie einen Pfad aus.
- Klicken Sie auf **OK**, um den Code in das gewählte Verzeichnis zu schreiben.

Der generierte Code wird an die vorgegebene Stelle geschrieben. Sie können die Datei mit beliebigem Texteditor oder mit dem C-Code-Editor öffnen. Die Namen der generierten Dateien werden im Monitorfenster erfasst.

## Generierung von ausführbarem Code:

---

- Wählen Sie im Projekteditor **Component** → **Build**

oder



- klicken Sie auf **Build Executable Code**, um eine ausführbare Datei zu erzeugen.

Der Code für das ganze Projekt wird generiert, kompiliert und gelinkt. Wenn keine Fehler auftreten, wird eine ausführbare Datei erzeugt. Der während der Codegenerierung erzeugte Quell- und Objektcode wird in der ASCET-Datenbank abgelegt.

Oder

- Wählen Sie **Component** → **Build from directory**, um eine ausführbare Datei zu erzeugen, ohne den generierten Code in der ASCET-Datenbank abzulegen.

Die für das Abspeichern benötigte Zeit wird dabei gespart.

Diese Alternative kann insbesondere bei der Codegenerierung für große Projekte erforderlich sein, falls der verwendete Rechner nicht genügend Hauptspeicher besitzt.



Wird eine ausführbare Datei generiert, werden alle Dateien (einschließlich des Quellcodes) standardmäßig im `.\cgen\`-Verzeichnis erzeugt.

### Hinweis

Wenn im Optionsfenster die Option **Keep files in Code Generation Directory** deaktiviert ist (s. Seite 50), wird der Inhalt des Codegenerierungsverzeichnisses `.\cgen\` von ASCET bei jedem Abschluss Ihrer Arbeit mit ASCET gelöscht. Um den generierten Code dauerhaft zu speichern, speichern Sie den generierten Code wie auf Seite 472 beschrieben in einer Datei. Ein nachträgliches Aktivieren der Option hat für die laufende Arbeit keine Wirkung.

Sie haben auch die Möglichkeit, vorhandenen Objektcode ohne neue Codegenerierung und -compilierung zu einer ausführbaren Datei zu linken. Dies ist vor allem dann von Interesse, wenn Ihr Projekt externe C-Code-Dateien enthält, die im Make-Prozess eingebunden werden.

### Linken von Objektdateien zu einer ausführbaren Datei:

- Nehmen Sie in den externen Dateien die nötigen Änderungen vor und erzeugen Sie Objektdateien.
- Wählen Sie im Projekteditor **Component** → **Link Only**.

Die zum Projekt gehörigen internen Objektdateien werden aus der Datenbank auf die Festplatte geschrieben und dort mit den externen Objektdateien zu einer ausführbaren HEX-Datei verlinkt.

### Hinweis

**Link Only** funktioniert nur dann einwandfrei, wenn die internen Dateien mit **Component** → **Build** oder der Schaltfläche **Build Executable Code** erzeugt wurden, da nur in diesem Fall die Dateien **in der Datenbank** abgelegt werden.

### Experimentieren mit quantisiertem Gleitkomma-Code

Code mit quantisierter Gleitkomma-Arithmetik wird aus dem gleichen physikalischen Modell erzeugt wie Gleitkomma-Arithmetik. Die Codegenerierung muss einfach nur umgeschaltet werden.

## Umschalten auf quantisierte Gleitkomma-Arithmetik:

---



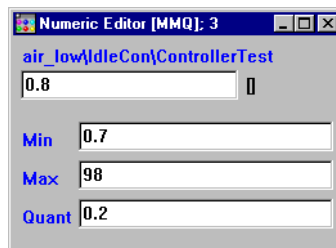
- Klicken Sie auf die Schaltfläche **Project Properties**.
- Wählen Sie in der Option „Code Generator“ **Quantized Physical Experiment** aus.
- Klicken Sie auf **OK**.  
Beim nächsten Starten des Experiments wird Ihr Code mit den aktuellen Einstellungen generiert.

Das Experimentieren mit Modellen mit quantisierter Gleitkomma-Arithmetik funktioniert ebenso wie mit normaler Gleitkomma-Arithmetik. Der Unterschied besteht darin, dass es beim Experimentieren mit Modellen mit quantisierter Gleitkomma-Arithmetik einen speziellen numerischen Editor zum Einstellen der Intervallgrenzen sowie zur Quantisierung des Wertes gibt.

## Öffnen des quantisierten numerischen Editors:

---

- Wählen Sie in der Fläche „Elements“ der Experimentierumgebung das zu verstellende Element aus.
- Wählen Sie **Elements** → **Calibrate**.  
Es erscheint ein Dialogfenster zur Auswahl des Editors.
- Wählen Sie aus der Liste den `[Numeric Editor [MMQ]]` und klicken Sie auf **OK**.  
Der quantisierte numerische Editor wird mit den Werten des aktuellen Datensatzes sowie mit den Werten aus der aktuellen Implementierung geöffnet.



Quantisierte Gleitkomma-Arithmetik beeinflusst nur die Zuweisung zu Variablen. Jedesmal, wenn eine Variable geschrieben oder ihr Wert im numerischen Editor geändert wird, wird der Wert an die Quantisierungsinformation angepasst.

### Arbeiten mit dem quantisierten numerischen Editor:

---

- Wählen Sie das Wertefeld.
- Ändern Sie den Wert nach Wunsch.
- Drücken Sie <ENTER>.

Der Wert wird im Experiment geändert und automatisch an die gewählte Quantisierung und das Intervall angepasst.

- Wählen Sie eines der Felder „Min“, „Max“ oder „Quant“ und ändern Sie den Wert wunschgemäß.
- Drücken Sie <ENTER> oder wählen Sie ein anderes Feld.

Die neuen Angaben werden zum Experiment übertragen und der Wert automatisch an das neue Intervall und die Quantisierung angepasst.

Quantisierter Gleitkomma-Arithmetik kann auch beim Experimentieren mit Komponenten Anwendung finden, jedoch nur bei Komponenten, die nicht in C-Code beschrieben sind. Die zum Experimentieren vorgesehene Komponente ist aus dem Projekteditor anstatt aus dem Komponentenmanager zu öffnen.

### Experimentieren mit einer Komponente in quantisierter Gleitkomma-Arithmetik:

---



- Klicken Sie im Projekteditor auf die Schaltfläche **Specify Code Generation Settings**.
- Wählen Sie in der Option „Code Generator“ **Quantized Physical Experiment** aus.
- Wählen Sie die Komponente in der Fläche „Elements“.
- Wählen Sie **Element** → **Edit Component**.  
Es wird der entsprechende Spezifikationseditor geöffnet.
- Experimentieren Sie mit der Komponente.

#### 4.8.11 Generieren von Applikationsdaten

---

Im Projekteditor können Sie alle Dateien erzeugen, die benötigt werden, um Ihren Code auf einem Controller-Target auszuführen und mit einem Applikationssystem (z.B. INCA) zu messen und zu kalibrieren. ASCET benutzt die ASAM-MCD-Norm zum Generieren der im Applikationssystem erforderlichen Informationen. Die ASAM-MCD-2MC-Datei stellt die Schnittstelle zu allen Applikationssystemen dar, die das Standard-ASAM-MCD-Format erkennen.

##### **Erzeugen von Applikationsdateien:**

---

- Wählen Sie im Projekteditor **ASAM-2MC** → **Write**, um die ASAM-MCD-2MC-Dateien für Ihr System zu erzeugen.  
Ein Dateiauswahlfenster wird geöffnet.
- Wählen Sie einen Pfad und geben Sie einen Dateinamen für die ASAM-MCD-2MC-Dateien an.
- Klicken Sie auf **Speichern**, um die Applikationsdateien im Dateisystem zu speichern.  
ASCET erzeugt die ASAM-MCD-2MC-Datei und die Hexadezimaldatei, die den Code für Ihr System enthält.

Im Fenster „Project Properties“ können Sie im Knoten „ASAM-2MC“ Optionen für die Generierung von ASAM-MCD-2MC-Dateien festlegen. Das Vorgehen ist im Abschnitt „Projekteinstellungen“ auf Seite 438 beschrieben.

#### 4.9 Container

---

Die seit ASCET V5.0 vorhandenen Container dienen als Behälter für Projekte, Klassen und Module. Ihr Zweck ist es, Modelle und Datenbanken zu strukturieren und verschiedene Datenbankobjekte unter eine gemeinsame Versionskontrolle zu stellen. Beim Wechsel von früheren ASCET-Versionen werden Netzwerke in Container konvertiert (s. Kapitel 4.9.2). Anders als die Netzwerke in früheren ASCET-Versionen haben Container jedoch keine weitere Funktionalität.

Container können mit Ausnahme von Ordnern alle Arten von Datenbankobjekten enthalten, auch andere Container. Dabei sind direkte (der Container enthält sich selbst) und indirekte (der Container enthält einen zweiten Container, der wiederum den ersten enthält) Rekursionen zulässig. Allerdings kann jedes Datenbankobjekt nur einmal im selben Container enthalten sein.

## 4.9.1 Arbeiten mit Containern

---

Container werden (wie Enumerationen) nicht in einem eigenen Editor bearbeitet, sondern im Komponentenmanager. Dort steht Ihnen in der Containeransicht im Register „Container components“ ein Kontextmenü ähnlich dem anderer Ansichten zur Verfügung.

### **Containeransicht aufrufen:**

---

- Markieren Sie im Komponentenmanager im Feld „1 Database“ einen existierenden Container

*oder*



- erzeugen Sie einen neuen Container (siehe Seite 79).

Im Feld „3 Contents“ wird das Register „Container components“ angezeigt.

- Klicken Sie irgendwo in das Register „Container components“, um es in den Fokus zu bringen.

### **Datenbankobjekt aufnehmen (Kontextmenü/Tastatur):**

---

- Wählen Sie aus dem Kontextmenü **Add**

*oder*

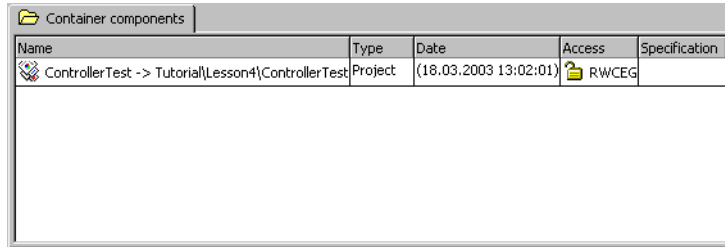
- drücken Sie auf <Ins>.

Das Fenster „Select Item“ öffnet sich.

- Wählen Sie dort aus der Liste „1 Database“ die Komponente aus, die Sie hinzufügen möchten.

- Klicken Sie auf **OK**, um das Objekt aufzunehmen und das Fenster „Selected Items“ zu schließen.

Das Objekt wird in den Container aufgenommen und im Register „Container components“ angezeigt.



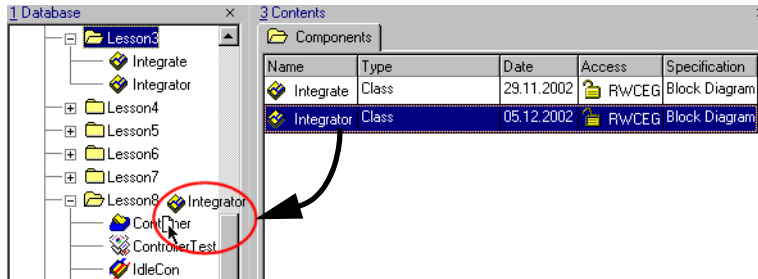
Name	Type	Date	Access	Specification
ControllerTest -> Tutorial\Lesson4\Controllertest	Project	{18.03.2003 13:02:01}	RWCEG	

### Datenbankobjekt aufnehmen (Drag & Drop):

- Klappen Sie im Komponentenmanager in der Liste „1 Database“ den Ordner auf, der den Container enthält, der das Objekt aufnehmen soll.
- Wählen Sie in der Liste „1 Database“ den Ordner aus, der die Datenbankobjekte enthält, die in den Container aufgenommen werden sollen.

Im Feld „3 Contents“ wird der Inhalt des Ordners angezeigt.

- Ziehen Sie die Datenbankobjekte aus dem Register „Components“ auf den Container in der Liste „1 Database“.



### Hinweis

Sie können das Datenbankobjekt **nicht** aus der Liste „1 Database“ in den Container ziehen.

Die Objekte werden in den Container aufgenommen.

Name	Type	Date	Access	Specification
ControllerTest -> Tutorial\Lesson4\ControllerTest	Project	{18.03.2003}	RWCEG	
Integrator -> Tutorial\Lesson3\Integrator	Class	{08.07.2002}	RWCEG	Block Diagram

Sie können auf dieselbe Weise Objekte aus einem Container in einen anderen ziehen. Die Objekte werden dabei nicht aus dem ersten Container entfernt.

### **Aufgenommene Objekte umbenennen:**

#### Hinweis

Wenn Sie ein Objekt in einem Container umbenennen, wird – anders als beim Umbenennen von aufgenommenen Objekten in Komponenten oder Projekten – das ursprüngliche Datenbankobjekt ebenfalls umbenannt.

- Markieren Sie im Register „Container components“ das Objekt, das Sie umbenennen wollen.
- Wählen Sie aus dem Kontextmenü **Rename** oder



- drücken Sie <F2>. Der Name des Objekts wird hervorgehoben.
- Geben Sie einen neuen Namen ein und drücken Sie <RETURN>. Das Objekt wird sowohl im Container als auch in der Datenbank umbenannt.

### Objekte bearbeiten:

---

Sie können ein Objekt, das Sie bearbeiten wollen, aus dem Container wie aus einem Ordner heraus öffnen.

- Markieren Sie das Objekt, das Sie bearbeiten wollen.
- Wählen Sie **Component** → **Edit Item**  
*oder*
- wählen Sie aus dem Kontextmenü **Edit**

#### Hinweis

*Diese beiden Optionen stehen für in Containern enthaltene Container nicht zur Verfügung.*

*oder*

- drücken Sie auf <RETURN>

*oder*

- doppelklicken Sie auf das Objekt, um den Komponenteneditor zu öffnen.

Wenn das Objekt ein Container ist, wird dieser im Feld „1 Database“ ausgewählt und im Feld „3 Container components“ angezeigt.

### Objekte aus dem Container entfernen:

---

- Markieren Sie im Register „Container components“ die Objekte, die Sie löschen wollen.
- Wählen Sie aus dem Kontextmenü **Select All**  
*oder*
- drücken Sie <CTRL>+<A>, wenn Sie alle Objekte auswählen wollen.



- Wählen Sie aus dem Kontextmenü **Delete** oder

- drücken Sie <DEL>, um die ausgewählten Objekte aus dem Container zu entfernen.

Die Objekte werden nur aus dem Container entfernt, nicht aus der Datenbank gelöscht.

### Containeransicht sortieren:

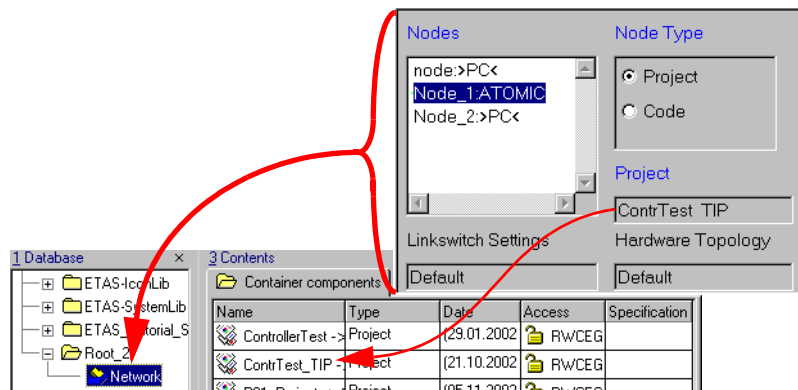
- Wählen Sie aus dem Kontextmenü **Sort by** → **<Spalte>**

oder

- klicken Sie auf den Namen einer Spalte, um die Anzeige nach dieser Spalte zu sortieren.

## 4.9.2 Container und Netzwerke

Die Container in ASCET 5.2 ersetzen die Netzwerke früherer ASCET-Versionen. Wenn Sie eine alte Datenbank öffnen, die Netzwerke enthält, werden diese automatisch in Container konvertiert. Dabei werden die Projekte, die den verschiedenen Knoten des Netzwerks zugeordnet sind, in den Container eingefügt. Weitere im Netzwerk vorhandene Information wird *nicht* übernommen.



## 4.10 Bearbeiten von Elementen

Elemente sind Schnittstellenelemente, Variable, Tabellen, Arrays und komplexe Elemente (referenzierte Komponenten).

Jedes Element hat eine Anzahl von Eigenschaften, die für jedes Element entsprechend den Anforderungen im Modell konfiguriert werden können.

Die gewählte Konfiguration gilt für alle Vorkommen eines Elements innerhalb einer Komponente.

### *Instanzen und Vorkommen*

---

Ein Element kann mehrere Vorkommen haben. Ein Vorkommen eines Elements im Blockdiagramm ist das Äquivalent zum Schreiben des Namens des Elements in einer textbasierten Programmiersprache. Änderungen in einem Vorkommen eines Elements beeinflussen alle anderen Vorkommen dieses Elements.

Wenn referenzierte Komponenten als Elemente verwendet werden, ist Vorsicht geboten. In diesem Falle kann jede Komponente mehrere Instanzen haben, von denen jede wiederum mehrere Vorkommen haben kann.

#### 4.10.1 Elementkonfiguration

---

Das Bearbeiten der Konfiguration eines Elements erfolgt im Elementeditor. Der Aufruf des Elementeditors kann aus jedem Spezifikationseditor sowie dem Komponentenmanager erfolgen.

#### **Öffnen des Elementeditors:**

---

- Markieren Sie im Spezifikationseditor im Feld „Elements“ das zu bearbeitende Element.

– Wählen Sie **Element** → **Edit**

*oder*

- Klicken Sie mit der rechten Maustaste auf das Element und wählen Sie **Edit** aus dem Kontextmenü

*oder*

- Doppelklicken Sie auf das gewünschte Element in der Fläche „Elements“.

*Oder*

- Klicken Sie im Zeichenbereich (BDE) bzw. im Register „Graphics“ (Projekteditor) mit der rechten Maustaste auf das Element und wählen Sie **Edit** aus dem Kontextmenü.

*Oder*

- Wählen Sie in der Browser-Ansicht des Spezifikationseditors bzw. im Komponentenmanager das Register „Elements“.
- Markieren Sie das zu bearbeitende Element.

- Drücken Sie <ENTER>

*oder*

- klicken Sie mit der rechten Maustaste im Register „Elements“ und wählen Sie **Edit** aus dem Kontextmenü.

Das Dialogfenster „Element Editor“ öffnet sich.

Wenn die Option **Edit primitive Elements** (oder **Edit Implementation Cast**) im ASCET-Optionsfenster aktiviert ist (s. „Optionen für Bestätigungsfenster“ auf Seite 47), wird der Elementeditor bei der Erzeugung eines neuen Elements automatisch geöffnet.

Sie können diese Option mit der in allen Elementeditoren vorhandene Option **Always show editor for new elements** überschreiben.

Je nach Art des ausgewählten Elements kann der Elementeditor unterschiedlich aussehen, da nicht alle Felder immer vorhanden sind. Die Funktionen der vorhandenen Felder ist in allen Fällen gleich.

**Element Editor for: cont::cont**

Name:

Unit:

Comment:

Dimension:

Model Type:

- Logic
- Signed Discrete
- Unsigned Discrete
- Continuous
- Enumeration

Kind:

- Constant
- System Constant
- Parameter
- Variable
- Input
- Output

Scope:

- Local
- Imported
- Exported

Existence:

- Virtual
- Non-Virtual

Dependency:

- Dependent
- Independent

Memory:

- Volatile
- Non-Volatile

Calibration:

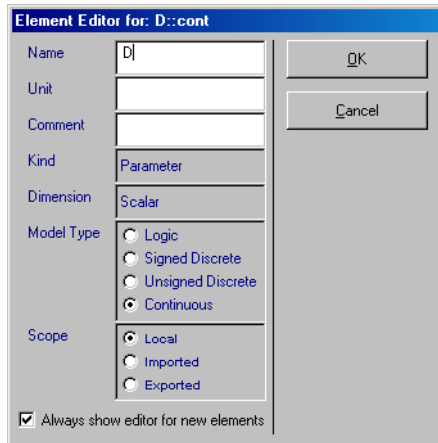
- Yes
- No

Always show editor for new elements

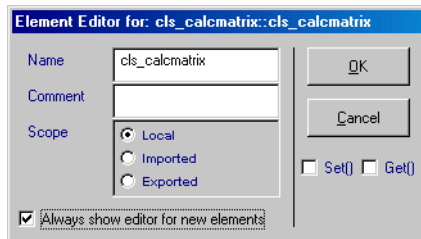
Buttons: OK, Cancel, Formula

Checkboxes: Variants, Set(), Get()

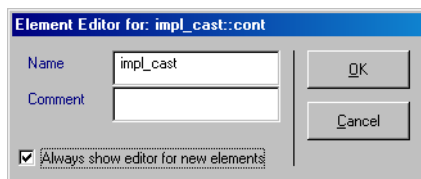
**Abb. 4-2** Elementeditor für Basiselemente. Je nach Element sind verschiedene Optionen oder Felder deaktiviert.



**Abb. 4-3** Elementeditor für Elemente eines CT-Blocks. Die Einstellmöglichkeiten für Get- und Set-Port sowie einige Felder fehlen ganz, andere Felder dienen nur noch der Anzeige.



**Abb. 4-4** Elementeditor für eine aufgenommene Komponente. Er enthält nur wenige Felder.



**Abb. 4-5** Elementeditor für einen Implementation-Cast. Lediglich der Name und der Kommentar können bearbeitet werden.

## Bearbeiten der Konfiguration eines Elements:

---

Konfigurieren Sie das Element wie gewünscht. Gehen Sie dazu wie folgt vor:

- Geben Sie im Elementeditor eine Maßeinheit in das Feld „Unit“ ein.  
Jedem Element kann eine Maßeinheit zugeordnet werden. Die Maßeinheit ist nur für Dokumentationszwecke bestimmt und hat keinen Einfluss auf die Funktionalität des Elements.
- Geben Sie einen Kommentar in das Feld „Comment“ ein.  
Der Kommentar wird angezeigt, wenn die Dokumentation für die Komponente generiert wird.
- Legen Sie im Feld „Model Type“ den Typ des Elements fest.  
Wenn Sie den Typ **Enumeration** wählen, wird das Kombikästchen neben dem Feld „Model Type“ (s. Abb. 4-2) aktiv. Es enthält alle in der Datenbank verfügbaren Enumerationen.
- Legen Sie im Feld „Kind“ die Art des Elements fest.  
An dieser Stelle können Sie Konstanten oder Systemkonstanten (s. „Die Art von Elementen“ im ASCET Referenzhandbuch) definieren.  
Die Arten **Input** und **Output** sind nur für die Ein- und Ausgänge in Zustandsautomaten verfügbar.
- Geben Sie im Feld „Scope“ den Geltungsbereich an (s. Abschnitt „Der Geltungsbereich von Elementen“ im ASCET Referenzhandbuch).
- Legen Sie im Feld „Existence“ fest, ob das Element virtuell ist oder nicht.  
Dieses Feld ist nur für Parameter und Variablen (ausgenommen Enumerationen) verfügbar.

- Geben Sie im Feld „Dependency“ an, ob es sich um einen abhängigen Parameter handelt. Dieses Feld ist nur für Parameter (ausgenommen Enumerationen) verfügbar.
- Legen Sie im Feld „Memory“ fest, ob das Element in den flüchtigen (**Volatile**) oder nichtflüchtigen (**Non-Volatile**) Speicher des Steuergeräts geschrieben wird. Daten im nichtflüchtigen Speicher des SG werden bei der Initialisierung nicht überschrieben.

### Hinweis

Die Zustandsvariable  $s_m$  eines Zustandsautomaten kann nicht im Elementeditor bearbeitet werden. Um dieser Variablen das Attribut **non-volatile** zuzuweisen, rufen Sie im Feld „Elements“ das Kontextmenü der Variablen auf und wählen die Funktion **Settings** → **Non-Volatile**.

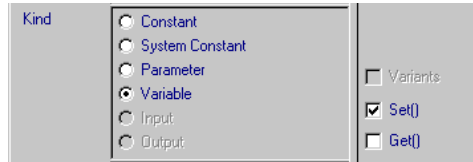
- Legen Sie im Feld „Calibration“ fest, ob ein Parameter in einem Applikationssystem verstellt werden kann (**Yes**) oder nicht (**No**). Wenn Sie **No** wählen, wird bei der ASAM-MCD-2MC-Generierung für einen Parameter das Schlüsselwort `READ ONLY` generiert. Auf andere Elemente oder die Experimentierumgebung hat dieses Feld *keine* Auswirkung.
- Klicken Sie auf **OK**, um den Elementeditor zu schließen.

Es ist möglich, einzelne Elemente innerhalb einer Komponente freizugeben. Die Freigabe eines Elements bedeutet, dass von außerhalb der aktuellen Komponente auf das Element zugegriffen werden kann. Parameter und Konstanten können aus der aktuellen Komponente ausgelesen werden. Variablen hingegen können ein- und ausgelesen werden.

## Freigeben oder Sperren von Elementen:

---

- Öffnen Sie den Elementeditor.



- Klicken Sie auf die Option **Set()**.  
Im Kästchen erscheint ein Häkchen. Das Element ist freigegeben und kann von außerhalb beschrieben werden.
- Um die Einstellung rückgängig zu machen klicken Sie nochmals auf die Option **Set()**.
- Einen Ausgang fügen Sie hinzu, in dem Sie die Option **Get()** aktivieren.
- Klicken Sie **OK**.  
Im Layout wird ein Ein- oder Ausgang für das Element hinzugefügt.

### Hinweis

Nur wenn für Messages **Get()** und **Set()** explizit gesetzt werden, sind die entsprechenden Anschlüsse im Layout des Moduls vorhanden. Diese dienen jedoch allein der Visualisierung des Datenflusses; die Zuweisung erfolgt über Namensgleichheit.

Wenn Sie Get-/Set-Ports verwenden, ohne dass eine der Codeoptimierungsoptionen **Optimize Direct Access Methods (...)** (s. Abschnitt „Knoten „Optimization““) aktiviert ist, werden im generierten Code für den Direktzugriff auf die betreffenden Elemente eigene Methoden angelegt, die über einen Funktionsaufruf aufgerufen werden.

Wenn die Option **Optimize Direct Access Methodsethods (one level)** aktiviert ist, werden keine eigenen Methoden generiert, sondern der Direktzugriff verwendet.

Wenn die Option **Optimize Direct Access Methods (multiple levels)** aktiviert ist, gilt das auch für Direktzugriffe in verschachtelten Klassen.



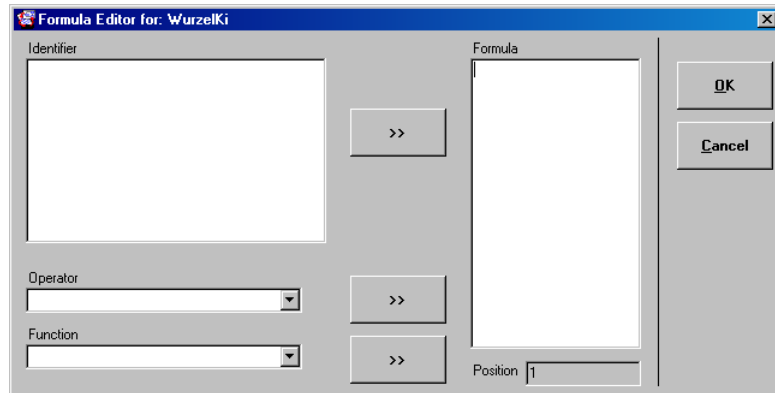
Der im Layout dargestellte Anschlussstyp hängt von der Art des Elements ab, das freigegeben wird. Für Parameter (auch Kennlinien und -felder) und Konstanten und Systemkonstanten sowie Send-Message kann nur ein Ausgang hinzugefügt werden. Für Receive-Message kann nur ein Eingang hinzugefügt werden. Für Variablen und Send&Receive-Message können sowohl ein Eingangs- als auch ein Ausgangsanschluss dargestellt werden.

#### 4.10.2 Abhängige Elemente

Abhängige Elemente können Parameter und Modellkonstanten sein. Das bedeutet, dass sie von einem anderen Element beeinflusst werden.

##### **Erstellen der Formel für abhängige Parameter:**

- Wählen Sie im Elementeditor aus dem Feld „Dependency“ die Option **Dependent**. Die Schaltfläche **Formula** wird aktiv.
- Klicken Sie auf die Schaltfläche **Formula**. Der Formeleditor für abhängige Elemente wird geöffnet.



Hier können Sie die Abhängigkeit des Elements in Form einer Formel (C-Syntax) darstellen. Im Feld „Position“ wird die Cursorposition im Feld „Formula“ angegeben.

- Klicken Sie mit der rechten Maustaste in das Feld „Identifizier“ und wählen Sie **Add** aus dem Kontextmenü.

So erzeugen Sie formale Parameter. Diese Parameter sind unabhängig vom Modell.

- Geben Sie im Feld „Identifizier“ einen Namen ein.

Im Kombikästchen „Operator“ finden Sie eine Liste der verfügbaren Operatoren.

Im Kombikästchen „Function“ finden Sie eine Liste der verfügbaren Funktionen.

Aus formalen Parametern, Operatoren und Funktionen könne Sie im Feld „Formula“ die Formel für die Abhängigkeit erstellen.

Dazu haben Sie zwei Möglichkeiten:

1. Geben Sie die Formel direkt im Feld „Formula“ ein.

*Oder*

2. Verwenden Sie die Schaltflächen >>.

- Wählen Sie im Kombikästchen „Function“ eine Funktion.
- Klicken Sie auf die nebenstehende Schaltfläche >>.



Die ausgewählte Funktion erscheint an der Cursorposition in der Fläche „Formula“.

- Wählen Sie im Kombikästchen „Operator“ einen Operator.
- Klicken Sie auf die nebenstehende Schaltfläche >>, um den Operator zur Formel hinzuzufügen.
- Markieren Sie in der Fläche „Formal Parameter“ einen formalen Parameter.





- Fügen Sie ihn mit der nebenstehenden Schaltfläche >> in die Formel ein.  
Operatoren, Funktionen und formale Parameter können in beliebiger Anzahl und Reihenfolge eingefügt werden, allerdings immer nur einer zur Zeit.

### **Bearbeiten der Formel eines abhängigen Parameters:**

---

Die Formel eines abhängigen Parameters kann nachträglich geändert werden.

- Öffnen Sie den Formeleditor für das abhängige Element, dessen Formel Sie bearbeiten wollen.
- Bearbeiten Sie die Formel wie im vorigen Abschnitt beschrieben.
- Klicken Sie mit der rechten Maustaste im Feld „Formal Parameter“ auf einen formalen Parameter, den Sie umbenennen wollen, und wählen Sie **Edit** aus dem Kontextmenü.
- Geben Sie den neuen Namen ein.
- Klicken Sie mit der rechten Maustaste im Feld „Formal Parameter“ auf einen formalen Parameter, den Sie löschen wollen, und wählen Sie **Delete** aus dem Kontextmenü.

#### **Hinweis**

---

*Die Änderungen durch Löschen oder Umbenennen betreffen nur das Feld „Formal Parameter“. die Formel im Feld „Formula“ muss von Hand angepasst werden.*

Um die Abhängigkeit des Elements im Modell zur Wirkung zu bringen, müssen die formalen Parameter entsprechend dem Modell spezifiziert werden.

### **Spezifizieren der formalen Parameter:**

---

- Markieren Sie das zu bearbeitende Element in der Fläche „Elements“.
- Wählen Sie **Element** → **Edit Data**  
oder

- Klicken Sie mit der rechten Maustaste auf das Element in der Fläche „Elements“ oder im Zeichenbereich und wählen Sie **Edit Data** aus dem Kontextmenü.

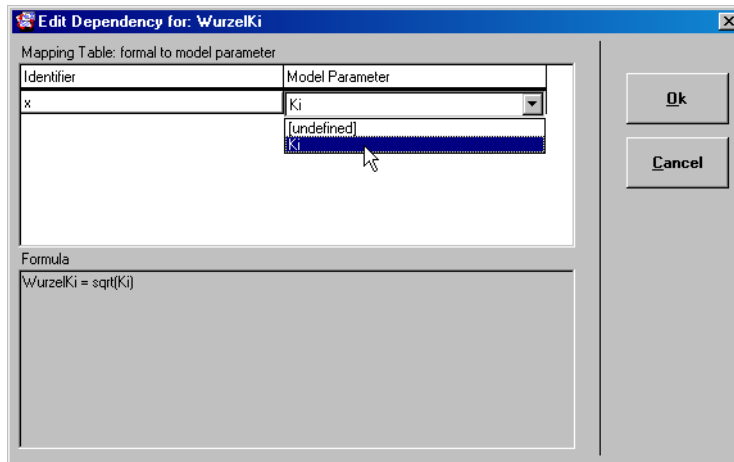
*Oder*

- Wählen Sie in der Browser-Ansicht das Register „Data“ .
  - Markieren Sie das zu bearbeitende Element.
  - Drücken Sie <ENTER>

*oder*

- klicken Sie mit der rechten Maustaste im Register „Data“ und wählen Sie **Edit** aus dem Kontextmenü.

Das Dialogfenster „Edit Dependency“ öffnet sich.



- Klicken Sie in der Spalte „Model Parameter“ auf das Feld [undefined].

Ein Kombikästchen öffnet sich, in dem die in der Komponente vorhandenen Parameter aufgelistet sind.

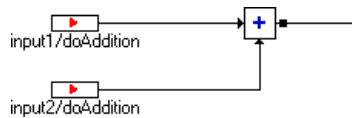
- Wählen Sie eines der aufgelisteten Elemente.  
Auf diese Weise haben Sie dem formalen Parameter einen Modellparameter zugewiesen. Dieser bestimmt nun den Wert des abhängigen Parameters.
- Schließen Sie das Fenster „Edit Dependency“.

Temporäre Variable können verwendet werden, um die Ergebnisse von Anweisungen oder Operationen zu speichern und eine mehrfache Ausführung innerhalb derselben Methode oder desselben Prozesses zu vermeiden (siehe Abschnitt „Temporäre Variable“ im Kapitel „Die Art von Elementen“ des ASCET Referenzhandbuchs).

#### Verwenden temporärer Variablen:

- Klicken Sie mit der rechten Maustaste auf den Operator, für den Sie eine temporäre Variable verwenden möchten.
- Wählen Sie aus dem Kontextmenü **Temporary Variable**.

Im Operator wird ein ausgefülltes Rechteck am Ausgangsanschluss angezeigt, um darauf hinzuweisen, dass das Ergebnis der Operation in einer temporären Variablen gespeichert wird.



- Um die Einstellung rückgängig zu machen, wählen Sie nochmals denselben Befehl.

## 4.11 Bearbeiten von Daten

Beim Spezifizieren einer Komponente können Sie jedem Element in Ihrer Spezifikation einen Ausgangswert zuordnen. Diese Werte lassen sich alle später verändern. Sie können verschiedene Datensätze spezifizieren, d.h. Sätze von Initialisierungswerten, zwischen denen Sie hin- und herschalten können, oder Sie können einzelne Werte während der Experimente ändern. Dieser Abschnitt beschreibt die verschiedenen Editoren für die verschiedenen Arten von Elementen.

Gewöhnlich wird ein Dateneditor zuerst aus der Umgebung der Komponentenentwicklung, z.B. dem Blockdiagrammeditor, heraus aufgerufen, um einem Element einen Standardwert zuzuordnen. Danach kann der Editor erneut aus der Experimentierumgebung heraus geöffnet werden, um den Wert des Elements während eines Experiments zu kalibrieren (siehe „Numerischer Editor“ auf Seite 668). Dateneditoren können ebenfalls zum Festlegen von Datensätzen für Komponenten oder Projekte verwendet werden.

Datensätze sind unabhängig von Implementierungen (vgl. Kapitel 4.12). Dennoch können zwischen ihnen Inkonsistenzen entstehen. Im Projektzusammenhang können Datensätze und Implementierungen so kombiniert werden, dass der im Datensatz eingestellte Wert eines Elements außerhalb des in der Implementierung eingestellten Wertebereichs liegt. Der Anwender ist hier selbst für konsistente Projekteinstellungen verantwortlich.

### Hinweis

*Zum Zeitpunkt der Codegenerierung müssen die Initialisierungswerte der Grundelemente innerhalb der Wertebereiche liegen, die durch ihre Implementierungen vorgegeben sind. Andernfalls wird für Parameter eine Fehlermeldung, für Variablen eine Warnung erzeugt.*

### Öffnen eines Editors

---

- Markieren Sie in der Fläche „Elements“ des Komponenteneditors das gewünschte Element.
- Wählen Sie in der Menüzeile **Element** → **Edit Data**.

Analog zum Öffnen des Elementeditors (vgl. Kapitel 4.10.1) gibt es auch weitere Möglichkeiten, den Dateneditor zu öffnen.

Entsprechend Ihrer Auswahl wird einer der nachfolgend beschriebenen Editoren geöffnet.

#### 4.11.1 Editoren für skalare Typen

---

##### *Numerischer Editor*

---

Der numerische Editor gibt den Wert eines numerischen Elements an und ermöglicht Ihnen die Bearbeitung dieses Wertes.



##### **Bearbeiten eines Wertes:**

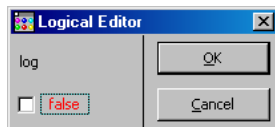
---

- Geben Sie den gewünschten Wert ein.  
*oder*
- Klicken Sie in die numerische Anzeige und bearbeiten Sie den Wert wie gewünscht.
- Klicken Sie auf **OK** um Ihre Änderung zu übernehmen.

##### *Logischer Editor*

---

Der logische Editor gibt den Wert eines logischen Elements an und ermöglicht Ihnen die Bearbeitung dieses Wertes.



##### **Bearbeiten eines logischen Wertes:**

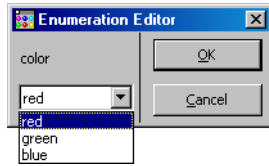
---

- Setzen Sie das Optionskästchen aktiv, um den Wert auf `true` zu setzen.
- Setzen Sie das Optionskästchen inaktiv, um den Wert auf `false` zu setzen.
- Klicken Sie auf **OK**, um Ihre Änderung zu übernehmen.

## Enumeration-Editor

---

Der Enumeration-Editor gibt den dem Element zugewiesenen Enumerator an und ermöglicht Ihnen die Auswahl eines anderen Enumerators aus der aktuellen Enumeration.



### Auswahl eines Enumerators:

---

- Klicken Sie auf das Kombikästchen.
- Wählen Sie den gewünschten Enumerator aus.
- Klicken Sie auf **OK**, um Ihre Änderungen zu übernehmen.

## 4.11.2 Der Editor für zusammengesetzte Typen (Tabelleneditor)

---

Tabellen sind ASCET Datenstrukturen zum Beschreiben von Kennlinien und -feldern. Die Kurven in einer Tabelle werden nicht durch eine mathematische Funktion, sondern durch Festlegen einzelner Datenpunkte definiert, die die Ausgabe von bestimmten Eingabewerten bilden. Die maximale Anzahl der Datenpunkte wird im Spezifikationseditor nach Auswahl eines Tabellenelements festgelegt. ASCET unterstützt ein- und zweidimensionale Tabellen.

Wie Sie eine Tabelle erzeugen, ist im Abschnitt „Arrays, Matrizen, Kennlinien und -felder“ auf Seite 222 beschrieben. Das Tabellenelement lässt sich nun aus der Fläche „Elements“ heraus editieren. Der Tabelleneditor (auch Multi-Editor) enthält eine Menüzeile, das Kombikästchen „v:“ mit dem (den) Namen der derzeit geöffneten Tabellen, einen Datenbereich und eine variable Anzahl von Einstellfeldern für Tabellengröße und Interpolation. Im Datenbereich werden die Abtastpunkte (bei Arrays/Matrizen Indexwerte) und die Stützstellenwerte (Ausgabewerte) angezeigt und können - je nach Art der editierten Tabelle - bearbeitet werden. Die in den folgenden Abschnitten kurz vorgestellten Editoren sind die verschiedenen Formen des Tabelleneditors. Eine ausführliche Beschreibung finden Sie in Kapitel 6.2.3 „Arbeiten mit Verstellfenstern“.

Obwohl Sie im Tabelleneditor während der Spezifikation und während der Experimente Punkte dynamisch hinzufügen können, darf die tatsächliche Anzahl von Abtastpunkten die bei der Erzeugung des Elements festgelegte

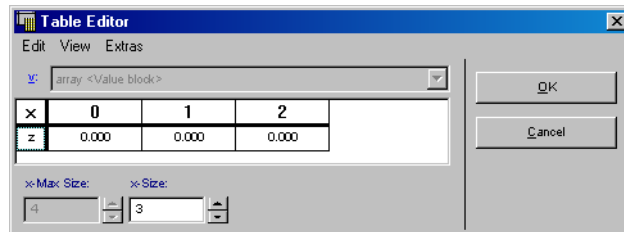


maximale Anzahl nicht überschreiten. Die maximale Anzahl von Werten kann nur im Spezifikationseditor mit dem Befehl **Element** → **Max Size** geändert werden.

### Array-Editor

Ein Array ist eine eindimensionale Tabelle, in der die Werte auf der x-Achse festgelegt sind, d.h. sie beginnen immer bei 0 und werden immer mit 1 inkrementiert. Ein Array in ASCET entspricht einem eindimensionalen Array in einer herkömmlichen Programmiersprache.

Der Array-Editor ist die einfachste Form des Tabelleneditors.



Im Datenbereich werden die Indexwerte und deren zugehörige Ausgabewerte angezeigt; zwei Einstellfelder für die Arraygröße sind vorhanden.

### Einrichten eines Arrays:

- Passen Sie die Anzahl der tatsächlichen Punkte auf der x-Achse im Feld „x-Size“ an.

Die entsprechende Anzahl von Zellen wird angelegt. Weitere Punkte auf der x-Achse können hinzugefügt werden, jedoch nur bis zur im Feld „x-Max-Size“ angegebenen maximalen Größe.

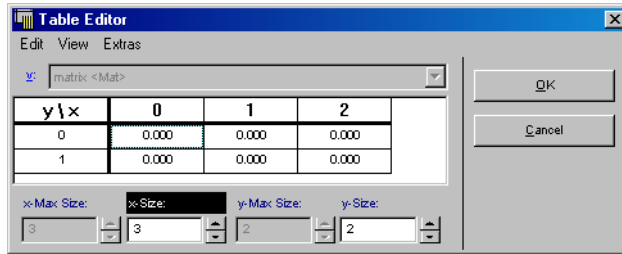
- Doppelklicken Sie im Editorfenster auf eine Zelle auf der z-Achse.

Die Zelle wird zum Eingabefeld.

- Geben Sie einen Wert ein und drücken Sie <ENTER>.
- Wiederholen Sie den Vorgang für alle anderen Werte.

This small screenshot shows a portion of the table editor. The row labeled 'z' is highlighted, and the cell containing '0.000' is selected, indicating it is ready for input.

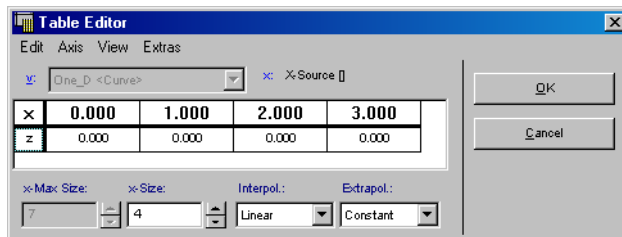
Zweidimensionale Arrays sind in der Form von Matrizen ebenfalls verfügbar.



Der Matrixeditor zeigt die Index- und Ausgabewerte beider Achsen an und hat insgesamt vier Einstellfelder für die Größe. Ansonsten funktioniert er wie der Array-Editor.

### 1D-Tabelleneditor

Eine eindimensionale Tabelle hat einen Eingabe- und einen Ausgabewert. Intern wird eine Anzahl von Abtastpunkten festgelegt und jedem Abtastpunkt ein Stützstellenwert zugeordnet. Haben Sie zum Beispiel an einem Abtastpunkt mit dem Wert 0.5 den Stützstellenwert 2 festgelegt, ist die Ausgabe der Tabelle immer dann 2, wenn der Eingabewert 0.5 beträgt. Liegt der Eingabewert zwischen zwei Abtastpunkten, wird das Ergebnis zwischen den beiden Werten interpoliert.



Im Datenbereich werden die Abtastpunkte und deren zugehörige Stützstellenwerte angezeigt; zwei Einstellfelder für die Tabellengröße sowie Einstellfelder für die Interpolation sind vorhanden.

Der 1D-Tabelleneditor funktioniert auf die gleiche Weise wie der Array-Editor. Der einzige Unterschied zwischen den Editoren besteht darin, dass im Tabelleneditor die Werte auf beiden Achsen bearbeitet werden können.

## Bearbeiten einer 1D-Tabelle:

---

- Klicken Sie im Blockdiagrammeditor mit der rechten Maustaste auf ein 1D-Tabellen-Element und wählen im Kontextmenü **Edit Data**.

Oder

- Markieren Sie das 1D-Tabellen-Element in der Fläche „Elements“ und wählen Sie in der Menüzeile **Element** → **Edit Data**.

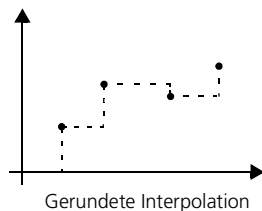
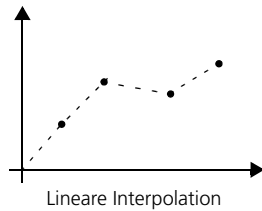
Der 1D-Tabelleneditor wird geöffnet.

- Ändern Sie die Werte auf der z-Achse wie auf Seite 497 beschrieben.
- Passen Sie die Anzahl der Abtastpunkte im Feld „x-Size“ an.

Damit wird eine Tabelle mit  $n$  Abtastpunkten auf der x-Achse der Tabelle angelegt. Ihnen werden standardmäßig Werte von 0 bis  $n-1$  zugeordnet.

- Wählen Sie einen Interpolationsmodus aus dem Kombikästchen „Interpol.“ aus.
  - **Linear** bedeutet, dass zwei Abtastpunkte durch eine gerade Linie verbunden werden; haben wir zum Beispiel einen Abtastpunkt 1 mit dem Wert 2 und einen Abtastpunkt 2 mit dem Wert 4, ist das Ergebnis für einen Eingabewert 1.5 gleich 3.
  - **Rounded** bedeutet, dass die Werte zwischen zwei Abtastpunkten immer gleich dem am kleineren Abtastpunkt sind. Mit den beiden Abtastpunkten, die wir im vorstehenden Punkt verwendet haben, würde das Ergebnis für jeden Eingabewert über 1 und unter 2 gleich 2 sein.

- Klicken Sie auf **OK**.



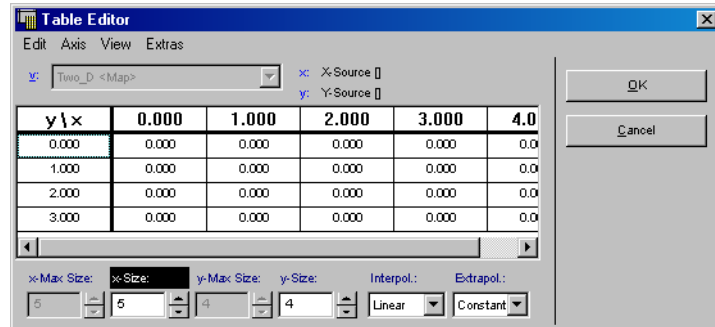
Der Extrapolationsmodus steht immer auf **Constant**. Bei allen x-Werten, die größer als der größte in der Tabelle festgelegte x-Wert sind, ist der angezeigte Wert der z-Wert des höchsten x-Wertes. Bei Werten, die unter dem kleinsten x-Wert liegen, wird der z-Wert des kleinsten x-Wertes angegeben.

Die Abtastpunkte werden auf der x-Achse des Datenbereichs angezeigt, die Werte befinden sich auf der z-Achse. Standardmäßig wird die Anzahl von Abtastpunkten, die Sie in das Dialogfenster der 1D-Tabelle eingeben haben, angelegt. Die standardmäßigen Abtastpunkte bilden einen Bereich von 0 bis zur Anzahl der Abtastpunkte minus 1.

### 2D-Tabelleneditor

Der Unterschied des 2D-Tabelleneditor zum 1D-Tabelleneditor besteht darin, dass die Abtastpunkte zwei Dimensionen anstatt nur einer Dimension haben. Die 2D-Tabelle hat zwei Eingaben und zeigt einen Ausgabewert für jedes Eingabepaar an.

Die Abtastpunkte werden in der obersten Zeile und äußersten linken Spalte der Tabelle im Datenbereich angegeben. Der 2D-Tabelleneditor ist die aufwendigste Form des Tabelleneditors.



### Feste Tabellen

Eine feste Tabelle ist eine Tabelle, in der der Abstand zwischen den Achsenpunkten nicht geändert werden kann. Für den Abstand kann ein beliebiger Wert ausgewählt werden, jedoch ist dieser Wert für alle Punkte gleich. Wenn zum Beispiel für den Abstand zwischen den Abtastpunkten der Wert 3 und der Offsetwert 4 eingegeben wurde, hat der erste Achsenpunkt den Wert 4, der zweite den Wert 7 und der dritte den Wert 10 usw.

#### Einrichten der Achsenpunkte:

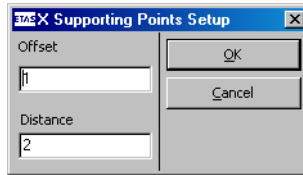


- Öffnen Sie die Festkennlinie, die Sie bearbeiten wollen, im Tabelleneditor.

Die nebenstehende Abbildung enthält ein einfaches Beispiel.

- Wählen Sie im Tabelleneditor **Axis** → **X Supporting Points Setup**.

Das Fenster „X Supporting Points Setup“ öffnet sich.



- Geben Sie im Feld „Offset“ den Versatz für den ersten Achsenpunkt ein (im Beispiel: 1).
- Geben Sie im Feld „Distance“ den Abstand zwischen den Achsenpunkten ein (im Beispiel: 2).

### Hinweis

Geben Sie in beiden Feldern nur **ganze Zahlen** ein, da sonst bei der Arbeit mit Implementierungen Fehler auftreten.

- Klicken Sie auf **OK**.

Die eingegebenen Werte werden übernommen, die Achsenpunkte werden angepasst. Die nebenstehende Abbildung enthält das aktualisierte Beispiel.

Fixed_ID <Fixed curve>						
x	↑ 1.0	↑ 3.0	↑ 5.0	↑ 7.0	↑ 9.0	
z	0.0	0.0	0.0	0.0	0.0	

Oder

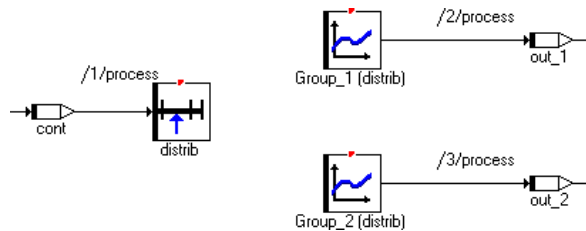
- Klicken Sie auf **Cancel**, um die Werte zu verwerfen und zum Ausgangszustand zurückzukehren.

Feste Tabellen sind als 1D-Tabellen und als 2D-Tabellen verfügbar. Die Achsenpunkte der Y-Achse einer festen 2D-Tabelle bearbeiten Sie mit **Axis** → **Y Supporting Points Setup**. Abgesehen von der Bearbeitung der Achsenpunkte funktioniert der Editor für feste Tabellen wie der Tabelleneditor.

## Gruppentabellen und Verteilungen

Gruppentabellen sind Tabellen, die die gleiche Verteilung von Achsenpunkten, jedoch verschiedene Anzeigewerte haben. Auf diese Weise ist es möglich, verschiedene Anzeigewerte für die gleiche Eingabe zu erhalten. Die Verteilungen der Achsenpunkte und der einzelnen Gruppentabellen werden in gesonderten Elementen abgelegt; erstere heißen Verteilungen oder Distributionen.

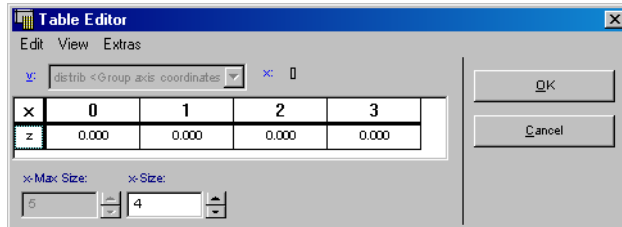
In einer Verteilung werden nur Achsenpunkte, jedoch keine Anzeigewerte festgelegt. Sobald eine Verteilung festgelegt wurde, können die Gruppentabellen, für die diese Verteilung gelten soll, bestimmt werden. In einem Blockdiagramm hat eine Verteilung nur eine Eingabe, wohingegen Gruppentabellen, wie nachfolgend beschrieben, nur eine Ausgabe haben.



Gruppentabellen sind als 1D-Tabellen und als 2D-Tabellen verfügbar.

**Gruppentabellen-Editor:** Der Gruppentabellen-Editor funktioniert wie der Tabelleneditor. Wenn Sie eine Gruppentabelle anlegen wollen, werden Sie aufgefordert, eine Verteilung festzulegen. Diese Verteilung wird dann der Tabelle zugeordnet, d.h. alle Stützpunkte, die in der Verteilung festgelegt sind, werden auch für die Gruppentabelle festgelegt. Es ist nicht möglich, Änderungen an der x-Achse in diesem Editor vorzunehmen, jedoch ist es möglich, aus dem Spezifikationseditor mit **Element** → **Edit Distribution** auf eine andere Verteilung umzuschalten. Die Zuordnung von Werten auf der z-Achse funktioniert wie im 1D-Tabelleneditor.

**Verteilungseditor:** Der Verteilungseditor funktioniert wie der Array-Editor. Der Hauptunterschied besteht darin, dass die Achsenwerte der Stützpunkte zur z-Achse der Tabelle hinzugefügt werden. Die x-Achse dient nur zu Navigationszwecken und kann nicht geändert werden. In diesem Editor steht keiner der Befehle zum Ändern der x-Achse zur Verfügung.



### 4.11.3 Datensätze

Jede Komponente bzw. jedes Projekt kann eine Reihe von Datensätzen umfassen. Datensätze bestimmen die Ausgangswerte der Elemente, d.h. Parameter und Variable, und der Komponenten. Ein Datensatz enthält einen Initialisierungswert für jedes Element der Komponente oder des Projekts. Somit definieren Datensätze Varianten von Komponenten und Projekten.

Der Umgang mit Datensätzen ist bei Komponenten und Projekten gleich. Jeder Datensatz einer Komponente umfasst die Ausgangswerte für alle in dieser Komponente verwendeten Grundelemente. Bei komplexen Elementen, die ihre eigenen Datensätze haben, verfügt der Datensatz der Komponente über eine Referenz zu einem der Datensätze des komplexen Elements.

Jede Komponente kann über verschiedene Datensätze verfügen. Wird eine Komponente zum ersten Mal angelegt, wird mit ihr ein Standarddatensatz mit dem Namen Data erstellt.

#### **Ansehen von Datensätzen:**

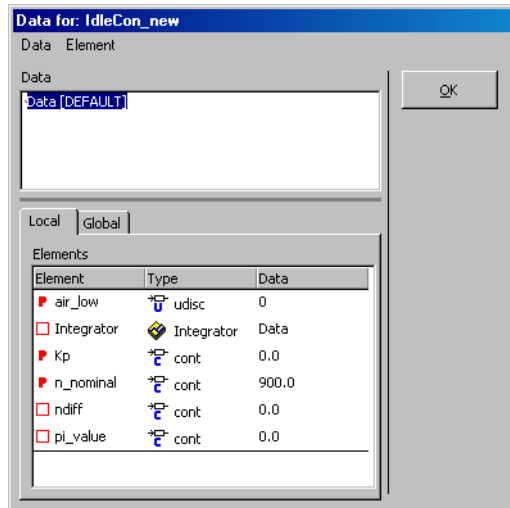
- Wählen Sie im Spezifikationseditor **Component** → **Edit Data**, um sich die Daten für das aktuelle Projekt oder die aktuelle Komponente anzusehen.

Das Dialogfenster „Data for: <Component>“ wird geöffnet.

Die Namen aller Datensätze werden in der Fläche „Data“ angezeigt.

Alle im Projekt oder der Komponente enthaltenen Komponenten werden zusammen mit

den Elementen in der Fläche „Elements“ angezeigt. Bei Basiselementen wird hinter dem Elementnamen und -typ der Wert angegeben. Bei komplexen Elementen wird der Name des referenzierten Datensatzes angezeigt.



- Wählen Sie in der Fläche „Data“ den Datensatz aus, den Sie sich anschauen wollen.  
Die Werte für den ausgewählten Datensatz sind in der Fläche „Elements“ angegeben.
- Klicken Sie auf **OK**.

Der aktuell ausgewählte Datensatz, für den die Werte angezeigt werden, wird zum aktiven Datensatz, wenn Sie den Dateneditor schließen. Sie können den Dateneditor auch aus dem Komponentenmanager öffnen, indem Sie im Register „Data“ eine referenzierte Komponente wählen.



## Erstellen/kopieren eines neuen Datensatzes:

---

- Wählen Sie **Data** → **Add**.

Ein neuer Datensatz wird angelegt. Alle Elemente haben einen Standardwert. Dieser Wert ist entweder 0.0 oder `true` bei Basiselementen oder der entsprechende Standarddatensatz bei komplexen Elementen.

Oder

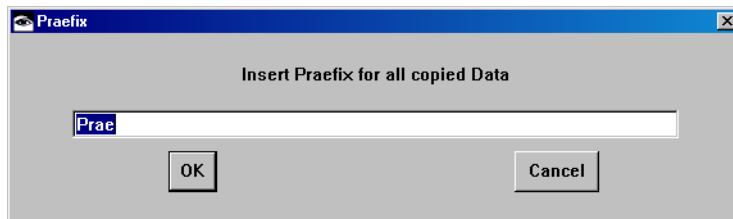
- Wählen Sie **Data** → **Copy** → **Flat**.

Der aktive Datensatz wird kopiert, d.h. der neu angelegte Datensatz enthält die gleichen Werte. Sind Referenzen zu anderen Datensätzen vorhanden, werden diese Referenzen mit kopiert.

Oder

- Wählen Sie **Data** → **Copy** → **Recursive**.

Ein Eingabefenster öffnet sich.



- Geben Sie ein Präfix für die Namen der kopierten Datensätze ein und klicken Sie **OK**.

Der aktive Datensatz wird kopiert und von allen referenzierten Datensätzen werden rekursive Kopien erstellt, d.h. die Kopien der Datensätze sind ebenfalls rekursiv. Der neue Datensatz enthält Referenzen auf die Kopien der referenzierten Datensätze.

### Hinweis

*Dieser Prozess funktioniert nur bei lokalen Variablen.*

### Löschen eines Datensatzes:

---

- Wählen Sie den zu löschenden Datensatz im Dialogfenster „Data for: <Component>“ aus.
- Wählen Sie **Data** → **Delete**.

Der Datensatz wird gelöscht. Wurde dieser Datensatz von anderen Datensätzen referenziert, erfolgt nun die Referenz auf den Datensatz, der nach dem Löschen aktiv wird. Der Standarddatensatz kann nicht gelöscht werden.

### Umbenennen eines Datensatzes:

---

- Wählen Sie den umzubenennenden Datensatz aus.
- Wählen Sie **Data** → **Rename**.
- Geben Sie den neuen Namen des Datensatzes ein und drücken Sie <ENTER>.

Wird ein Datensatz im Dialogfenster „Data for: <Component>“ ausgewählt, wird er zum aktiven Datensatz. Wenn Sie den Komponenten- oder den Projekteditor schließen und ihn danach erneut öffnen, ist wieder der Standarddatensatz aktiv.

Immer ein Standarddatensatz ist mit jeder Komponente und jedem Projekt verbunden, und zwar zunächst der, der automatisch erstellt wird, wenn die Komponente oder das Projekt angelegt werden.

### Auswählen eines Datensatzes als Standarddatensatz:

---

- Öffnen Sie das Dialogfenster „Data for: <Component>“.
- Wählen Sie den Datensatz aus, den Sie zum Standarddatensatz machen wollen.
- Wählen Sie **Data** → **Become Default**.

Der ausgewählte Datensatz wird zum Standarddatensatz. Immer wenn die Komponente oder das Projekt geöffnet werden, wird dieser Datensatz aktiv.

Enthält ein Datensatz Referenzen auf andere Datensätze, ist es möglich, die Datensätze durchzublättern, auf die die Referenzen verweisen.

## Durchblättern eines Datensatzes:

---

- Wählen Sie den Datensatz aus, den Sie durchblättern wollen.
- Wählen Sie das komplexe Element (Komponente) aus, dessen Datensatz Sie durchblättern wollen.
- Wählen Sie **Element** → **Edit**

*oder*

- doppelklicken Sie im Feld „Elements“ auf das gewünschte komplexe Element.

Das Dialogfenster „Data for: <Component>“ wird für den referenzierten Datensatz geöffnet.

- Folgen Sie dieser Vorgehensweise rekursiv, wenn Sie durch eine gesamte Datensatzhierarchie blättern wollen.

## Bearbeiten von Daten in einem Datensatz:

---

Sie haben verschiedene Möglichkeiten, die Daten einzelner Elemente zu bearbeiten.

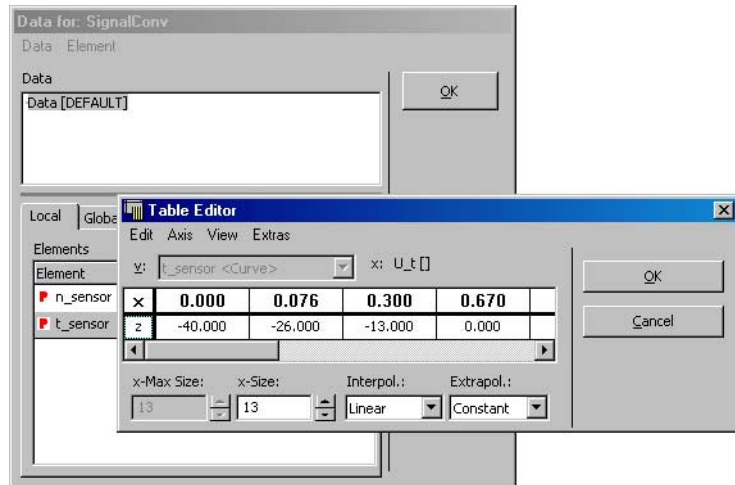
### 1. Dateneditor öffnen

- Wählen Sie das Basiselement aus, das Sie bearbeiten wollen.

Wenn Sie ein Basiselement in einem referenzierten Datensatz bearbeiten wollen, müssen Sie diesen zuerst wie vorstehend beschrieben öffnen.

- Wählen Sie **Element** → **Edit**.

Der entsprechende Dateneditor wird geöffnet, z.B. wird ein 1D-Tabelleneditor geöffnet, wenn es sich bei dem Element um eine Kennlinie handelt.



## 2. Daten kopieren

- Wählen Sie das Basiselement, dessen Daten Sie kopieren wollen.
- Wählen Sie **Element** → **Copy Data To Buffer** oder
- wählen Sie **Copy Data To Buffer** aus dem Kontextmenü.

Die Daten des Elements werden in die Zwischenablage der Datenbank kopiert.

- Wählen Sie das Basiselement, in das Sie die Daten kopieren wollen.

### Hinweis

*Sie können Daten nur zwischen Elementen des gleichen Typs austauschen. Bei zusammengesetzten Typen (Array, Matrix, Tabelle) muss außerdem die aktuelle Größe übereinstimmen.*

- Wählen Sie **Element** → **Paste Data From Buffer**

oder

- wählen Sie **Paste Data From Buffer** aus dem Kontextmenü.

Die Daten werden aus der Zwischenablage der Datenbank in das Element kopiert.

Der Datensatz eines Projektes oder einer Komponente kann von der funktionalen Spezifikation getrennt exportiert werden. Mit diesem Merkmal lassen sich die Datensätze und die Spezifikation parallel entwickeln.

#### **Exportieren eines Datensatzes:**

---

- Wählen Sie im Dialogfenster „Data for: <Component>“ den zu exportierenden Datensatz aus.

- Wählen Sie **Data** → **Export**.

Das Dialogfenster von Windows zur Dateiauswahl wird geöffnet.

- Geben Sie einen Dateinamen ein und drücken Sie <ENTER>.

Der ausgewählte Datensatz wird exportiert.

Die gleichen Befehle sind ebenfalls in den Hauptmenüs der Spezifikationseditoren sowie des Projekteditors (**Component** → **Export Data**) verfügbar.

#### **Anzeigen von Unterschieden zwischen zwei Datensätzen:**

---

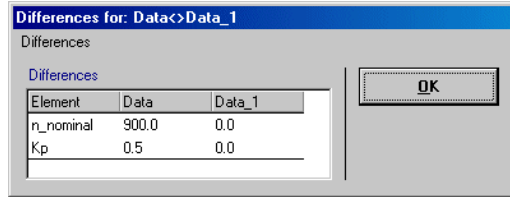
- Wählen Sie zwei Datensätze der Komponente oder des Projekts in der Fläche „Data“ aus.

Sie können mehrere Datensätze durch Anklicken auswählen, wenn Sie die Taste <CTRL> gedrückt halten.

- Wählen Sie **Data** → **Show Differences** → **Flat**.

Die Datensätze werden auf der ersten Ebene verglichen, d.h. nur die Werte der im Projekt oder der Komponente enthaltenen Basiselemente werden verglichen. Referenzierte Datensätze werden nicht berücksichtigt. Das Dialogfenster „Differences“ wird geöffnet und zeigt die Unterschiede in den Datensät-

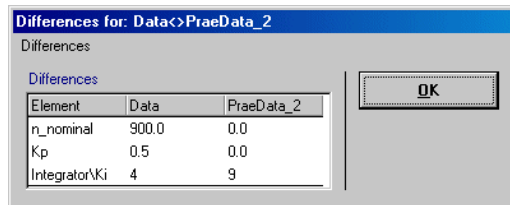
zen. Die Unterschiede für skalare Elemente werden direkt angegeben; nichtskalare Elemente mit unterschiedlichen Daten werden nur aufgelistet.



Oder

- Wählen Sie **Data** → **Show Differences** → **Recursive**.

Die Datensätze werden rekursiv verglichen, d.h. unabhängig von den referenzierten Datensätzen werden die Werte der Basiselemente in allen referenzierten Komponenten verglichen.



- Wählen Sie im Fenster „Differences“ **Differences** → **Copy To Clipboard**, um das Ergebnis des Vergleichs in die Zwischenablage zu kopieren.

Von dort kann das Ergebnis dann in eine andere Anwendung eingefügt werden.

Es ist möglich, die Daten einer Tabelle oder eines Arrays in eine Datei zu schreiben und sie wieder zu lesen. Die Daten werden im tab-abgegrenzten ASCII-Format geschrieben.

### Schreiben der Daten eines Arrays oder einer Tabelle in eine Datei:

---

- Wählen Sie in der Fläche „Elements“ des Dateneditors oder Spezifikationseditors eine Tabelle oder einen Array aus.
- Wählen Sie **Element** → **File Out Data**.  
Das Dialogfenster von Windows zur Dateiauswahl wird geöffnet.
- Wählen Sie einen Pfad und einen Dateinamen aus.
- Klicken Sie auf **Speichern**.  
Die Daten der ausgewählten Tabelle oder des ausgewählten Arrays werden in die Datei geschrieben.

### Einlesen der Daten eines Arrays oder einer Tabelle aus einer Datei:

---

- Wählen Sie in der Fläche „Elements“ eine Tabelle oder einen Array aus.
- Wählen Sie **Element** → **File In Data**.  
Das Dialogfenster von Windows zur Dateiauswahl wird geöffnet.
- Wählen Sie die Datei aus, die die Daten enthält, die Sie einlesen wollen.
- Klicken Sie auf **Öffnen**.

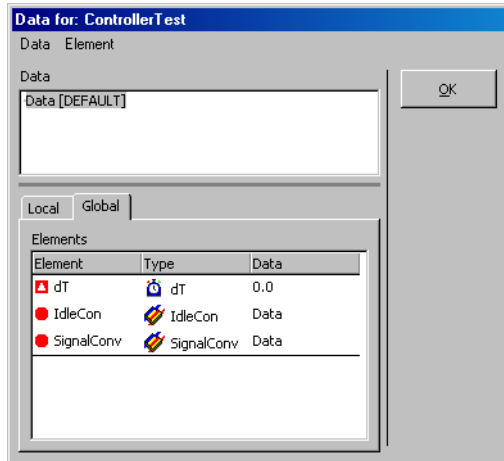
Entspricht das Format der Datei nicht dem ausgewählten Element, bleibt der Datensatz unverändert und es erscheint ein Mitteilungsfenster. Dies passiert zum Beispiel dann, wenn die Abtastpunkte nicht in einer monoton ansteigenden Reihenfolge vorliegen. Enthält die Datei fehlende Zahlen, werden diese als Nullen behandelt (zum Beispiel dann, wenn die Zahl der Abtastwerte nicht der Zahl der Abtastpunkte entspricht). Enthält die Datei ungültige Zeichen, werden diese als Null interpretiert. Die Größe des Elements wird den Daten angepasst, die eingelesen werden.

Zwischen Komponenten und Projekten besteht ein wesentlicher Unterschied, nämlich der, dass Projekte sowohl globale Elemente als auch die Komponenten, die sie referenzieren, umfassen. Im Unterschied zu den Daten für die Komponenten, für die ein Projekt verschiedene Datensätze haben kann, gibt es nur einen Datensatz für die globalen Elemente.

## Arbeiten mit dem Datensatz von globalen Elementen:

---

- Wählen Sie im Projekteditor **Component** → **Edit Data** aus, um das Dialogfenster des Dateneditors zu öffnen.
- Klicken Sie auf das Register „Global“ im Dialogfenster, um sich die Daten für die globalen Elemente im Projekt anzusehen.



Selbst wenn in der Fläche „Data“ mehrere Datensätze angezeigt werden, können Sie diese nicht auswählen oder bearbeiten.

ASCET bietet einen zusätzlichen Satz von Tools für den Datenaustausch an. Benutzer können mit verschiedenen Filter- und Datenaustauschoptionen sowie Dateiformaten einfache und rekursive Einlese- und Schreiboperationen für die Parameter in Datensätzen durchführen.

### Einstellen der Datenaustauschoptionen:

---

- Öffnen Sie im Komponentenmanager das ASCET-Optionsfenster.
- Stellen Sie im Knoten „Data Exchange“ die gewünschten Optionen ein.

Die Bedeutung der Optionen ist im Abschnitt „Optionen für den Datenaustausch“ auf Seite 68 beschrieben.



Nachdem Sie die Optionen für den Datenaustausch für Ihr System festgelegt haben, können Sie die Datenaustausch-Tools für den Import und Export von Datensätzen verwenden. Die Tools stehen über die entsprechende Menüauswahl im Dateneditor zur Verfügung.

#### **Verwenden der Tools für den Datenaustausch:**

---

- Wählen Sie im Dateneditor den Datensatz aus, den Sie bearbeiten wollen.
- Wählen Sie **Data** → **File Out Recursive**, um den ausgewählten Datensatz und alle referenzierten Datensätze aus der Datei herauszuschreiben.
- Wählen Sie **Data** → **File Out**, um den ausgewählten Datensatz herauszuschreiben.
- Wählen Sie **Data** → **File In Recursive**, um einen Datensatz mit allen referenzierten Datensätzen einzulesen.
- Wählen Sie **Data** → **File In**, um einen Datensatz einzulesen.
- Wählen Sie **Data** → **Show Data File**, um sich eine Datendatei anzusehen.
- Wählen Sie **Data** → **Show File In Log File**, um sich die Log-Datei für Lesevorgänge anzusehen.
- Wählen Sie **Data** → **Show File OutLog File**, um sich die Log-Datei für Schreibvorgänge anzusehen.

## 4.12 Bearbeiten von Implementierungen

---

Es gibt drei Implementierungseditoren, einen für Komponenten und Projekte, einen für Basiselemente – d.h. Variablen, Parameter und Systemkonstanten – und einen für Methoden und Prozesse. Im Implementierungseditor für Basiselemente sind je nach Art des Elements Felder deaktiviert oder zusätzliche Register vorhanden, aber der Grundaufbau ist immer derselbe.

Implementierungen sind unabhängig von Datensätzen (vgl. Kapitel 4.11). Dennoch können zwischen ihnen Inkonsistenzen entstehen. Im Projektzusammenhang können Datensätze und Implementierungen so kombiniert werden,

dass der im Datensatz eingestellte Wert eines Elements außerhalb des in der Implementierung eingestellten Wertebereichs liegt. Der Anwender ist hier selbst für konsistente Projekteinstellungen verantwortlich.

### Hinweis

*Zum Zeitpunkt der Codegenerierung müssen die Initialisierungswerte der Grundelemente innerhalb der Wertebereiche liegen, die durch ihre Implementierungen vorgegeben sind. Andernfalls wird für Parameter eine Fehlermeldung, für Variablen eine Warnung erzeugt.*

Für Literale wird die geeignetste Implementierung automatisch durch die Codegenerierung abgeleitet, so dass sie nicht spezifiziert werden muss. Ihre Implementierung wird über die erste auf sie stattfindende Zuweisung festgelegt. Dasselbe gilt für Konstanten.

#### 4.12.1 Implementierung von Komponenten/Projekten

Sie können den Implementierungseditor für Komponenten/Projekte

- aus dem Komponentenmanager (s. „Bearbeiten der Implementierung:“ auf Seite 90),
- aus einem beliebigen Spezifikationseditor  
*oder – bei enthaltenen Komponenten –*
- aus dem Implementierungseditor der enthaltenden Komponente

öffnen und die Werte für die aktuelle Komponente oder das aktuelle Projekt bearbeiten.

#### Implementierungseditor für Komponenten öffnen:

1. editierte Komponente/Projekt
  - Wählen Sie im Spezifikationseditor **Component** → **Edit Implementation**, um die Implementierung für das aktuelle Projekt oder die aktuelle Komponente zu bearbeiten.
2. enthaltene Komponente (Spezifikationseditor)
  - Markieren Sie in der Liste „Elements“ eine enthaltene Komponente.
  - Wählen Sie **Element** → **Implementation**  
*oder*

- wählen Sie aus dem Kontextmenü **Edit Implementation**

oder

- verwenden Sie die Browse-Ansicht des Spezifikationseditors:



- Klicken Sie auf die Lasche **Browse**.
- Öffnen Sie in der Browse-Ansicht den Implementierungsektor für die aufgenommene Komponente wie auf Seite 90 beschrieben.

### 3. enthaltene Komponente (Implementierungsektor)

- Markieren Sie in der Liste „Elements“ eine enthaltene Komponente.

- Wählen Sie **Element** → **Edit**

oder

- doppelklicken Sie auf die Komponente

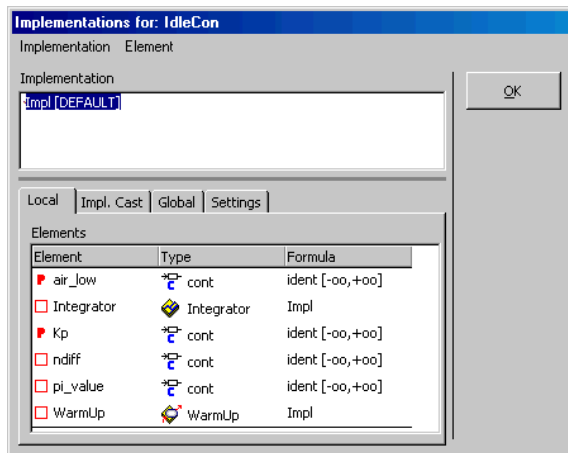
oder

- wählen Sie aus dem Kontextmenü **Edit**

oder

- drücken Sie die <RETURN>-Taste.

Der Implementierungsektor für die gewählte Komponente öffnet sich.



Die Namen aller für diese Komponente oder dieses Projekt definierten Implementierungen werden in der Fläche „Implementation“ angezeigt.

Alle im Projekt oder der Komponente enthaltenen lokalen Elemente und Komponenten werden im Register „Locals“ angezeigt, das Register „Impl. Cast“ enthält die Implementation-Casts (siehe „Implementation-Casts“ im ASCET-Referenzhandbuch). Im Register „Globals“ werden die globalen Elemente angezeigt. Bei Basiselementen wird hinter dem Elementnamen und -typ die Formel angegeben. Bei komplexen Elementen wird der Name der referenzierten Implementierung angezeigt.

Das Register „Settings“ enthält verschiedene Einstellmöglichkeiten.

### **Auswählen einer Implementierung:**

---

- Wählen Sie in der Fläche „Implementation“ die Implementierung aus, die Sie bearbeiten wollen.

Die Formeln für die ausgewählte Implementierung sind in der Fläche „Elements“ angegeben.

Durch einen Doppelklick auf einen Eintrag im Feld „Elements“ öffnen Sie den Implementierungseditor für das Element oder die aufgenommene Komponente.

- Klicken Sie auf **OK**, um das Fenster zu schließen.

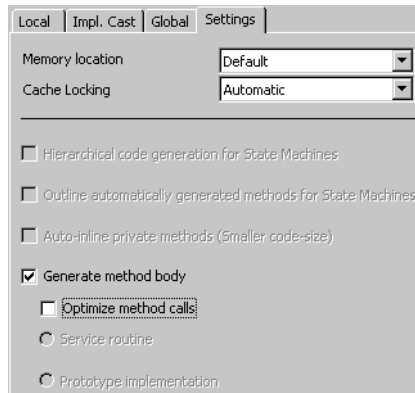
Alle Operationen, die für Datensätze zur Verfügung stehen, sind auch für Implementierungen verfügbar und funktionieren wie im Kapitel „Datensätze“ beschrieben. Sie können die Implementierung für lokale und globale Elemente sowie für Implementation-Casts spezifizieren, indem Sie das entsprechende Register im Implementierungseditor aktivieren.

## Einstellungen für die Implementierung vornehmen:

### Hinweis

Wenn in den Build-Optionen eine andere Einstellung als Object Based Controller Implementation ausgewählt wurde, bleiben die Einstellungen im Register „Settings“ wirkungslos.

- Wählen Sie das Register „Settings“.



Nur die Kombikästchen „Memory location“ und „Cache Locking“ sind für alle Komponenten (Klassen, Module und Projekte) verfügbar, die Optionen sind für Projekte und Module ausgegraut.

- Wählen Sie im Kombikästchen „Memory location“ den Speicherbereich, in dem die Datenstruktur für die Komponente angelegt wird. Die Speicherklasse für den Code hingegen wird im Implementierungseditor der jeweiligen Methode oder des Prozesses (s. Kapitel 4.12.8 auf Seite 540) definiert.

- Wenn Sie die Implementierung einer Klasse bearbeiten, achten Sie darauf, dass die Option **Generate method body** markiert ist, damit Code für die Klasse generiert wird.

Die Optionen **Hierarchical code generation for State Machines**, **Outline automatically generated methods for State Machines** und **Auto-inline private methods (Smaller code-size)** sind nur für Zustandsautomaten verfügbar. Ihre Verwendung ist unter „Outlining von Aktionen/Bedingungen ein-/ausschalten:“ und „Auto-Inlining ein-/ausschalten:“ beschrieben.

Cache-Locking ist im ASCET-RP-Benutzerhandbuch beschrieben, die übrigen Einstellmöglichkeiten sind im ASCET-SE Benutzerhandbuch beschrieben.

#### 4.12.2 Implementierung von skalaren nichtlogischen Elementen

---

Die Implementierung von nichtlogischen Elementen beschreibt die Umwandlung aus einer unendlichen Modelldomäne (entweder *continuous* oder *discrete*) in eine endliche Implementierungsdomäne. Aus diesem Grund muss der Bereich für die Werte im Modell mit Intervallgrenzen versehen sein. Darüber hinaus ist eine lineare Formel zur Beschreibung der Umwandlung von der physikalischen Darstellung in die Implementierungsdarstellung zu definieren.

Beim Modelltyp *continuous*, der eine unendlich feine Auflösung hat, bestimmt die Formel die Quantisierung in der Implementierungsdomäne bei Verwendung der speziellen Festkomma-Arithmetik. Die Quantisierung ist die Umkehrung des Gradienten der linearen Formel, da für die Implementierung die Ganzzahlen-Arithmetik angenommen wird.

##### **Implementierungseditor für Basiselemente öffnen:**

---

1. aus dem Komponentenmanager:
  - Wählen Sie in der Liste „1 Database“ eine Komponente oder ein Projekt.
  - Wählen Sie im Feld „3 Contents“, Register „Implementation“, das Basiselement, dessen Implementierung Sie bearbeiten wollen.
  - Wählen Sie **Component** → **Edit Item**

oder

- doppelklicken Sie auf das Element

oder

- wählen Sie aus dem Kontextmenü **Edit**

oder

- drücken Sie die <RETURN>-Taste.

2. aus dem Spezifikationseditor:

- Wählen Sie im Feld „Elements“ das Basiselement, dessen Implementierung Sie bearbeiten wollen.

- Wählen Sie **Element** → **Edit Implementation**

oder

- wählen Sie aus dem Kontextmenü **Edit Implementation**

oder

- verwenden Sie die Browse-Ansicht des Spezifikationseditors:



- Klicken Sie auf die Lasche **Browse**.
- Öffnen Sie in der Browse-Ansicht den Implementierungseitor wie auf Seite 90 beschrieben.

3. aus dem Implementierungseitor einer Komponente/ eines Projekts:

- Wählen Sie im Feld „Elements“ das Basiselement, dessen Implementierung Sie bearbeiten wollen.

- Wählen Sie **Element** → **Edit**

oder

- doppelklicken Sie auf das Element

oder

- wählen Sie aus dem Kontextmenü **Edit**

oder

- drücken Sie die <RETURN>-Taste.

Der Implementierungseitor für das Basiselement öffnet sich. Der Modelltyp wird vom Element vorgegeben.

Weitere Möglichkeiten zum Öffnen des Implementierungseditors bestehen analog zum Elementeditor für Basiselemente (vgl. Kapitel 4.10.1).

Bei neu angelegten Elementen geht das System nicht von der Verwendung eines im Projektkontext definierten Implementierungstyps, (s. „Implementierungstypen“ auf Seite 463), sondern von einer lokalen Implementierung aus. Für die Modelltypen `cont` und `disc` wird der Implementierungsdatentyp eingestellt, der im Optionsfenster, Knoten „Implementation“, angegeben ist (siehe Seite 53). Für die Wertebereiche werden die zum Typ passenden Maximalgrenzen eingestellt.

Für skalare, nichtlogische Elemente sieht der Implementierungseditor so aus:

The dialog box is titled "Implementation for: n Project: ControllerTestPC". It has two tabs: "Value" and "Additional Information". The "Additional Information" tab is selected. The dialog is divided into several sections:

- Implementation:** Includes a checkbox for "Use Implementation Type" and a dropdown menu.
- Transformation:** Includes a "Formula" dropdown set to "ident", a "Conversion" field with the text "f(phys) = phys", and a "Quantization" section with "Calculated" set to "1.0" and "Qu. Exp." set to "0".
- Master:** Includes radio buttons for "Model" and "Implementation" (which is selected).
- Model:** Includes a "Type" dropdown set to "cont", and "Min" and "Max" input fields with values "-2147483648.0" and "2147483647.0" respectively.
- Implementation:** Includes a "Type" dropdown set to "int32", "Min" and "Max" input fields with values "-2147483648" and "2147483647" respectively, and a checkbox for "Zero not included".
- Implementation Interval Adaptation:** Includes checked checkboxes for "Limit Assignments" and "Limit to maximum bit length" (set to "Automatic").
- Memory Location:** Includes a dropdown menu set to "Default".
- Cache Locking:** Includes a dropdown menu set to "Automatic".
- Consistency:** Includes a table with "Source" and "Conflict" columns.

At the bottom of the dialog are three buttons: "Auto Correction", "OK", and "Cancel".

Das Dialogfenster „Implementation for“ bietet die Möglichkeit, mit der Option **Use Implementation Type** und dem zugehörigen Kombikästchen einen vordefinierten Implementierungstyp (siehe „Implementierungstypen“ auf Seite 463) auszuwählen oder im Bereich „Implementation“ eine individuelle



Implementierung zu spezifizieren. In jedem Fall werden das Begrenzungsverhalten (Feld „Implementation Interval Adaptation“) und der Speicherbereich, in dem das Element abgelegt wird (Kombikästchen „Memory Location“, nur für Mikrocontroller-Targets relevant) individuell bestimmt. Im Feld „Consistency“ werden Warnungen und Fehlermeldungen angezeigt.

Im Register „Additional Information“ können Sie Kommentare oder Informationen eingeben, die sich speziell auf einzelne Codegeneratoren beziehen und nur nach Erfordernis ausgewertet werden. Die genaue Art der Informationen, die Sie hier eingeben können, ist von Ihrem Target und vom Codegenerator abhängig.

### *Individuelle Implementierung spezifizieren*

---

Für skalare Basiselemente ist die Option **Use Implementation Type** standardmäßig deaktiviert; das zugehörige Kombikästchen ist ausgegraut. Sie müssen an der Stelle nichts ändern, um eine individuelle Implementierung zu spezifizieren.

#### **Auswählen einer Formel:**

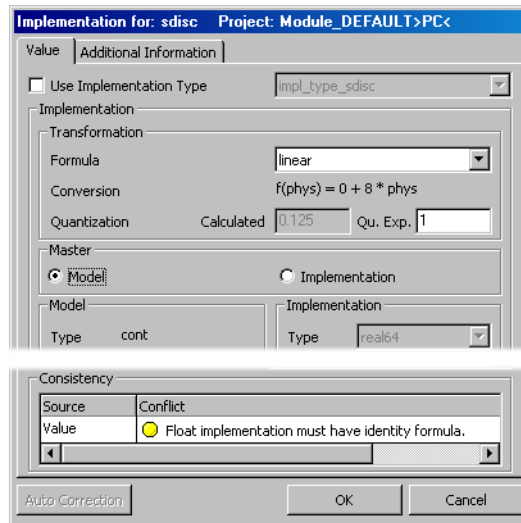
---

- Wählen Sie aus dem Kombikästchen „Formel“ die Umwandlungsformel aus. Alle im Projekt enthaltenen Formeln stehen zur Verfügung.

Die Quantisierung im Feld „Calculated“ wird aufgrund der Formel automatisch bestimmt. Wenn der Implementierungsdatentyp `real*` gewählt ist, wird die Quantisierung 1 angezeigt.

Zusätzlich kann eine Quantisierung in ein eigenes Feld („Qu. exp.“) eingegeben werden, die ausschließlich für das `Quantized Physical Experiment` verwendet wird.

Wenn die Größe den Modelltyp `cont` und den Implementierungsdatentyp `real32` oder `real64` oder einen anderen Modelltyp als `cont` hat, sollte nur die Identität als Formel gewählt werden, da nur diese von der Codegenerierung unterstützt wird. Wenn Sie eine andere Formel auswählen, wird eine Warnung ausgegeben:



- Wenn Sie die Identität auswählen wollen, wählen Sie im Kombikästchen „Formel“ `ident`.
- Klicken Sie auf **OK**, um den Implementierungseditor zu schließen.

*Oder*

- Klicken Sie auf **OK**, ohne die Identität auszuwählen.  
Die gewählte Formel wird übernommen.

*Oder*

- Klicken Sie **Cancel**, um den Dialog zu schließen und den ursprünglichen Zustand wieder herzustellen.

Wenn Sie ein Element vom Typ `sdisc` oder `udisc` bearbeiten, für das in einer früheren ASCET-Version eine andere Formel als die Identität ausgewählt wurde, erscheint eine Fehlermeldung.

### **Hinweis**

*Modelle aus früheren ASCET-Versionen können unverändert weiter verwendet werden. Jedoch werden bei der Codegenerierung Warnungen erzeugt, für den Fall dass sie Umrechnungsformeln für diskrete Elemente enthalten.*

Wenn Sie Änderungen in einer Formel vornehmen oder eine Formel rekursiv in der Implementierung eines Projekts ersetzen, können Sie sich Ihre gesamten Implementierungen automatisch aktualisieren lassen, so dass sie den neuen Wertebereichen und Datentypen entsprechen. Informationen zum Ersetzen der Formel und Aktualisieren der Implementierungen finden Sie im Abschnitt „Globale Veränderungen in Implementierungen“ auf Seite 462.

### **Einstellen der Masterseite einer Implementierung:**

ASCET kann entweder die Modell- oder die Implementierungsseite einer Implementierung als Ausgangspunkt für eine automatische Aktualisierungen nutzen. Der Ausgangspunkt wird mit den Optionen im Feld „Master“ festgelegt. Welche Seite standardmäßig als Masterseite voreingestellt ist, legen Sie im Optionsfenster im Knoten „Implementation“ fest (siehe „Implementierungsoptionen“ auf Seite 53).

- Klicken Sie im Implementierungseditor auf die Option **Model**, um festzulegen, dass die Einstellungen unter „Model“ als Ausgangspunkt für die Aktualisierung einer Implementierung verwendet werden.

Die Felder „Min“ und „Max“ in der Fläche „Model“ werden aktiviert.

Der Typ des Modells kann nicht bearbeitet werden, er ist durch das Element fest vorgegeben und wird im Feld „Type“ lediglich angezeigt.

Die Felder in der Fläche „Implementation“ sind bis auf die Option **Zero not included** deaktiviert und dienen nur der Anzeige.

Oder

- Klicken Sie auf die Option **Implementation**, um festzulegen, dass die Einstellungen unter „Implementation“ als Ausgangspunkt für die Aktualisierung einer Implementierung verwendet werden.

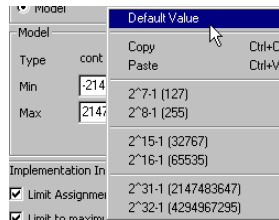
Die Felder in der Fläche „Implementation“ werden aktiviert.

Die Felder in der Fläche „Model“ sind deaktiviert und dienen nur der Anzeige.

### Implementierung spezifizieren (Master Model):

Die Option **Model** sollte als Master gewählt werden, wenn physikalische Modelleigenschaften ausschlaggebend für die Wahl der Implementierung sind. In diesem Fall gehen Sie bei der Spezifizierung wie folgt vor:

- Geben Sie in den Feldern „Min“ und „Max“ die Grenzen für das physikalische Intervall an.
- Um die Standardgrenzen für den Modelltyp einzustellen, klicken Sie mit der rechten Maustaste in eins der Felder und wählen Sie aus dem Kontextmenü **Default Value**.



Die Maximalwerte für den jeweiligen Modelltyp werden eingesetzt:

Typ	Min.	Max.
cont	-∞	+∞
sdisc	-2147483648	2147483647
udisc	0	4294967295

Die Werte für die Implementierung werden automatisch anhand der eingegebenen Werte und der Formel aktualisiert.

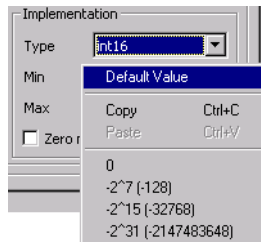
### Implementierung spezifizieren (Master Implementation):

Die Option **Implementation** sollte als Master gewählt werden, wenn bestimmte Code-Eigenschaften (z.B. eine gewünschte Bitbreite) ausschlaggebend für die Wahl der Implementierung sind. In diesem Fall gehen Sie bei der Spezifizierung wie folgt vor.

- Wählen Sie im Kombikästchen „Type“ den Implementierungsdatentyp.  
Das Kombikästchen enthält alle für das Element zur Verfügung stehenden Typen.
- Geben Sie in den Feldern „Min“ und „Max“ die Grenzen für das Intervall an.

### Hinweis

Wenn Sie den Implementierungsdatentyp `real*` gewählt haben, ignoriert die Codegenerierung die eingegebenen Intervallgrenzen und verwendet stattdessen  $\pm\infty$  für Modell und Implementierung.



- Um die Standardgrenzen für den Implementierungsdatentyp einzustellen, klicken Sie mit der rechten Maustaste in eins der Felder und wählen Sie aus dem Kontextmenü **Default Value**.

Die Maximalwerte für den jeweiligen Typ werden eingesetzt:

Typ	Min.	Max.
<code>real64<sup>a</sup></code>	<code>-∞</code>	<code>+∞</code>
<code>real32<sup>a</sup></code>	<code>-∞</code>	<code>+∞</code>
<code>int32</code>	<code>-2147483648</code>	<code>2147483647</code>
<code>uint32</code>	<code>0</code>	<code>4294967295</code>
<code>int16</code>	<code>-32768</code>	<code>-32767</code>
<code>uint16</code>	<code>0</code>	<code>65535</code>
<code>int8</code>	<code>-128</code>	<code>127</code>
<code>uint8</code>	<code>0</code>	<code>255</code>

a: nur für Modelltyp `cont`

Die Werte für das Modell werden automatisch aktualisiert.

Die für das Modell oder die Implementierung eingegebenen Werte werden zusammen mit der Formel auf ihre Konsistenz geprüft.

- Ist die Modellseite der Master, so werden die Min.- und Max.-Werte auf der Implementierungsseite über die Umrechnungsformel automatisch angepasst.

Wird dadurch das Intervall in der Spalte „Implementation“ größer, als der gewählte Datentyp erlaubt, wird der Datentyp automatisch angepasst (Ausnahme: Der Implementierungsdattentyp `real*` ist gewählt). Dabei wird der erste zum Intervall passende Datentyp gewählt, der größer oder gleich dem in den ASCET-Optionen gewählten Minimaltyp (s. „Implementierungsoptionen“ auf Seite 53) ist.

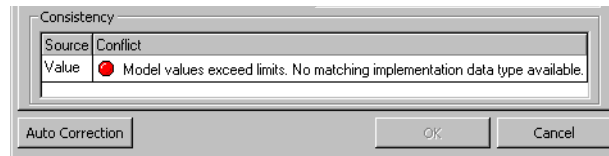
Beispiel: Wenn für eine Größe in der Spalte „Implementation“ der Typ `uint8` eingestellt ist und sich aus den Einstellungen unter „Model“ das Intervall `[-100 . . 255]` ergibt, wird stattdessen

- der Typ `int16` gewählt, wenn der „Minimum cont Data Type“ `int16` oder kleiner ist,
- der „Minimum cont Data Type“ gewählt, wenn dieser größer als `int16` ist.

Für die „Größe“ der Datentypen wird dabei die folgende Reihenfolge angenommen:

`int8` → `uint8` → `int16` → `uint16` → `int32` → `uint32`

Ist selbst `uint32` nicht ausreichend wird eine Fehlermeldung ausgegeben (auch wenn es sich um eine Größe vom Typ `cont` handelt und die Implementierungsdattentypen `real32` und `real64` verfügbar sind).



Wenn Sie die Fehlermeldung mit **Auto Correction** beantworten, wird auf der Masterseite ein Wert eingesetzt, der so bemessen ist, dass die *Implementierungsseite* nach der automatischen Aktualisierung die maximale Grenze für eines der beiden 32 Bit-Integer-Formate aufweist. Wenn Sie auf **Cancel** klicken, wird der Dialog geschlossen und der ursprüngliche Zustand wieder hergestellt.

Wird auf der Modellseite später ein kleineres Intervall gewählt, so wird ggf. der Implementierungsdattentyp wieder nach unten angepasst. Er wird jedoch nie kleiner, als der „Minimum ... Data Type“.

- Ist die Implementation-Seite der Master, so werden die Werte auf der Modellseite über die Umrechnungsformel automatisch angepasst.

Beim Erzeugen eines Elements erhält dieses zunächst den in den ASCET-Optionen gewählten Defaulttyp (s. „Implementierungsoptionen“ auf Seite 53). Wählt der Anwender nun in der Spalte „Implementation“ ein größeres Intervall, als der gewählte Datentyp erlaubt, wird der Datentyp automatisch angepasst. Dabei wird der kleinste zum Intervall passende Datentyp gewählt. Im Gegensatz dazu wird jedoch *keine* automatische Anpassung durchgeführt, wenn der Anwender ein Intervall angibt, das einen kleineren Datentypen erlauben würde.

Beispiel: Wenn für eine `cont` Größe in der Spalte „Implementation“ der Typ `uint8` eingestellt ist, und der Anwender

- das Intervall `[-100 . . 255]` vorgibt, wird stattdessen automatisch der Typ `int16` gewählt.
- das Intervall `[0 . . 100]` vorgibt, bleibt der Typ `uint8` erhalten.

Für die „Größe“ der Datentypen gilt wiederum die folgende Liste:

`int8` → `uint8` → `int16` → `uint16` → `int32` → `uint32`

Ist selbst `uint32` nicht ausreichend, so wird der Wertebereich automatisch auf die maximalen Grenzen für eines der beiden 32 Bit-Integer-Formate eingestellt, ohne dass eine Warnung ausgegeben wird.

- Steht der neue Wert im Widerspruch zu einem der anderen Werte, zum Beispiel dann, wenn der neue Mindestwert größer als der Höchstwert ist, so erscheint eine Fehlermeldung, die Sie mit **Auto Correction** oder **Cancel** quittieren können.
- Wenn die Implementierung der Master ist und Sie ein Element mit Modelltyp `udisc` und Implementierungsdattentyp `int*` bearbeiten, so erscheint im Feld „Consistency“ ebenfalls eine entsprechende Fehlermeldung, wenn Sie für das Implementierungsintervall negative Werte eingeben:

```
Model type udisc minimum cannot store
<negative value>.
```

- Wenn die berechneten Minimal- und Maximalwerte nicht mit den gespeicherten Werten übereinstimmen, wird davon ausgegangen, dass der Fehler in der Formel liegt. Folgende Fehlermeldung erscheint:

```
Values for min/max are not consistent with
current formula.
```

Der Fall tritt z.B. ein, wenn Sie im Projekt eine Formel ändern, ohne anschließend die Implementierungen zu aktualisieren (s. Seite 462).

Die folgenden Einstellungen können Sie immer vornehmen, egal, welche Seite als Master ausgewählt ist.

### **Ausschließen der Null aus dem Implementierungsintervall:**

---

Standardmäßig geht die Codegenerierung davon aus, dass das Implementierungsintervall die Null enthalten kann, d.h. die Option **Zero not included** ist deaktiviert. Es wird überprüft, ob der Nenner einer Division die Null enthält, und gegebenenfalls von der Codegenerierung C-Code erzeugt, der eine mögliche Division durch 0 zur Programmlaufzeit abfängt (s. Seite 449). Sie können diese Überprüfung abstellen.

- Wenn das Intervall der Implementierung die Null *nicht* enthält, aktivieren Sie die Option **Zero not included**.

Die Codegenerierung geht nun davon aus, dass die Größe nicht null werden kann. Wird die Größe als Nenner einer Division verwendet, findet keine Überprüfung statt.

#### **Hinweis**

*Eine Division durch Null kann kritische Zustände herbeiführen. Wenn Sie die automatische Überprüfung deaktivieren, liegt es in Ihrer Verantwortung, durch geeignete Maßnahmen im ASCET-Modell die Division durch Null zu verhindern.*

Wenn Sie mit älteren Modellen arbeiten, wird **Zero not included** für alle Elemente deaktiviert, die nicht mit einem implementierten Operator verbunden sind. Die Behandlung von Operatorimplementierungen aus früheren ASCET-Versionen ist in Kapitel 4.12.11 beschrieben.



## Einstellen der Limitierung:

---

Durch eine Berechnung können Werte außerhalb der Intervallgrenzen einer Variablen entstehen. Der Limiter berücksichtigt die Intervallgrenzen einer Variablen bei allen Zuweisungen an diese Variable, d.h. der Codegenerator erzeugt einen Begrenzungscode. Die Limiter ersparen die manuelle Limitierungsarbeit für die einzelnen Variablenwerte.

### Hinweis

Die Option **Limit Assignment** ist bei neu erzeugten Elementen standardmäßig aktiviert. Wenn Sie den Modelltyp auf `udisc` setzen (s. Seite 486), wird die Option automatisch deaktiviert.

Wenn für ein Element der Implementierungsdatentyp `real64` oder `real32` ausgewählt wurde, kann die Option **nicht** bearbeitet werden.

- Aktivieren Sie die Option **Limit Assignments**.

Damit ist festgelegt, dass der durch „Min“ und „Max“ definierte Wertebereich einer Variablen bei Zuweisungen durch den Codegenerator berücksichtigt wird.

Der Wert einer Variablen wird durch den Wertebereich begrenzt. Liegt ein zugewiesener Wert außerhalb des Wertebereichs, wird der relevante Grenzwert „Min“ bzw. „Max“ verwendet.

Oder

- Deaktivieren Sie die Option **Limit Assignments**.

Damit ist festgelegt, dass der definierte Wertebereich einer Variablen bei Zuweisungen durch den Codegenerator nicht berücksichtigt wird.

Der Wert einer Variablen wird in diesem Fall nicht durch den Wertebereich begrenzt, die maximale Grenze wird durch den Implementierungsdatentyp (`int8`, `int16`, usw.) festgelegt.

In früheren Versionen von ASCET gab es im Implementierungseeditor anstelle der Option **Limit Assignments** das Feld „Use Limiters“ mit den Optionen **Yes** und **No**. Wenn Sie nun mit älteren Modellen arbeiten, wird **Limit Assignments** entsprechend der Einstellung dieses Feldes gesetzt.

### Überlaufbehandlung festlegen:

---

Sie können das Verhalten bei Überlauf definieren. Die meisten Möglichkeiten stehen hier zur Verfügung, wenn Sie Arithmetische Dienste verwenden (s. Kapitel 4.14 „Arithmetische Dienste“).

- Aktivieren Sie die Option **Limit to maximum bit length**, wenn das Ergebnis einer Operation im Fall eines Überlaufs begrenzt werden soll.

Wenn der Wertebereich des Ergebnisses den im Knoten „Integer Arithmetic“ (s. Seite 448) spezifizierten Bereich überschreitet, wird Code erzeugt, der den Überlauf verhindert.

Das Kombikästchen neben der Option wird aktiviert.

- Wählen Sie aus dem Kombikästchen den Eintrag **Reduce Resolution**, wenn die Auflösung beim Begrenzen reduziert werden soll.

Diese Option vermeidet bei Bedarf einen Überlauf durch Rechtsverschiebung bei der Eingabe. Die Verschiebung wird automatisch ermittelt.

- Wählen Sie **Keep Resolution**, wenn die Auflösung nicht reduziert werden soll.

Wenn arithmetische Dienste aktiviert sind, verhindert diese Option bei Bedarf einen Überlauf am Ergebnis durch die Verwendung eines begrenzenden Dienstes.

Andernfalls wird die Codegenerierung mit einer Fehlermeldung abgebrochen.

- Wählen Sie `Automatic`, um die Überlaufbehandlung von der Verwendung arithmetischer Dienste abhängig zu machen.

Ist diese aktiviert, wird im Fall einer Überlaufbegrenzung die Methode „Keep resolution“ verwendet. Andernfalls wird die Methode „Reduce resolution“ verwendet.

Wenn Sie mit älteren Modellen arbeiten, wird **Limit to maximum bit length** entsprechend dem früheren Feld „Use Limiters“ gesetzt. `Automatic` wird für alle Elemente ausgewählt. Die Behandlung von Operatorimplementierungen aus früheren ASCET Versionen ist in Kapitel 4.12.11 beschrieben.

#### **Speicherbereich auswählen:**

---

- Wählen Sie im Kombikästchen „Memory Location“ den Speicherbereich, in dem das Element abgelegt wird.

Die zur Verfügung stehende Auswahl hängt vom Target ab.

#### *Implementierungstypen verwenden*

---

Anstelle der individuellen Implementierung können Sie auch einen vordefinierten Implementierungstyp zuweisen. Wie Sie diesen erstellen, ist im Abschnitt „Implementierungstypen“ auf Seite 463 beschrieben; hier geht es um die Verwendung.

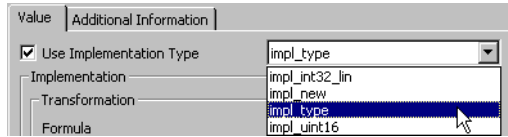
Ihnen stehen die Implementierungstypen aus dem aktuellen Projektkontext zur Verfügung. Das ist das Standardprojekt (s. Kapitel 4.8.1), wenn Sie die Implementierung aus einem Komponenteneditor heraus bearbeiten, oder das Projekt, aus dem heraus Sie die Implementierung bearbeiten.

#### **Implementierungstyp zuweisen:**

---

- Öffnen Sie den Implementierungseditor für ein Basiselement.

- Aktivieren Sie die Option **Use Implementation Type**.



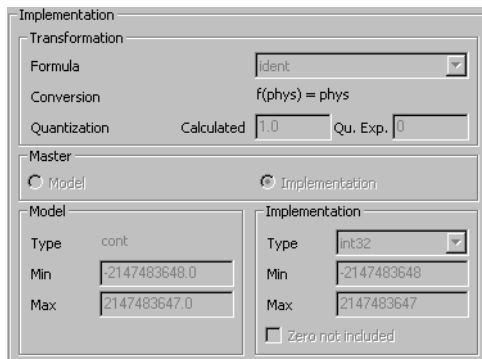
Das Kombikästchen neben der Option wird verfügbar. Darin sind alle verfügbaren Implementierungstypen enthalten.

- Wählen Sie einen Implementierungstyp aus.

### Hinweis

*Sie können einen Implementierungstyp nur für Größen mit dem gleichen Modelltyp zuweisen, d.h. ein Implementierungstyp mit dem Modelltyp cont kann nur kontinuierlichen Größen zugewiesen werden, nicht solchen vom Typ sdisc oder udisc.*

Die Einstellungen für Modell und Implementierung werden übernommen. Diese Werte können nicht geändert werden, sämtliche Felder im Bereich „Implementation“ sind für Eingaben gesperrt.



- Stellen Sie die Limitierung ein wie auf Seite 529 beschrieben ein.

- Wählen Sie im Kombikästchen „Memory Location“ den Speicherbereich für das Element aus.

Implementierungstypen werden von den Implementierungen der Basiselemente referenziert. Soll eine Kopie eines Implementierungstyps verwendet werden, z. B. um diese anschließend zu modifizieren, so kann zunächst ein Implementierungstyp aktiviert und dann wieder deaktiviert werden. In diesem Fall werden die Einstellungen des Implementierungstyps in die Implementierung des Basiselements übernommen.

#### **Hinweis**

*Sobald für ein Basiselement ein Implementierungstyp aktiviert wird, geht die vorher eingestellte Implementierung verloren.*

### 4.12.3 Implementierung von methoden-/prozesslokalen Variablen

---

Methoden- und prozesslokale Variable können automatisch oder explizit implementiert werden. Standard ist die automatische Implementierung; hierbei wird die Implementierung der ersten Größe übernommen, die der methoden-/prozesslokalen Variablen zugewiesen wird. Wenn Sie den Standard nicht verwenden wollen, gehen Sie wie folgt vor:

#### **Implementieren einer methoden-/prozesslokalen Variablen:**

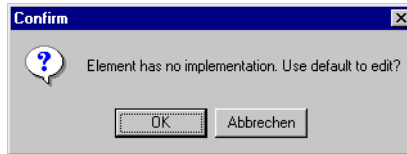
---

- Wählen Sie im Implementierungseditor der Komponente das Register „Local“.
- Wählen Sie im Feld „Elements“ die methoden-/prozesslokale Variable.

Der Menüpunkt **Element** → **Use automatic Implementation** ist markiert und ausgegraut, wenn keine explizite Implementierung eingegeben wurde.

- Öffnen Sie den Implementierungseeditor für die methoden-/prozesslokale Variable.

Wenn die automatische Implementierung aktiviert ist, öffnet sich ein Fenster mit einer Warnung:



- Bestätigen Sie mit **OK**.  
Der Implementierungseeditor wird geöffnet.
- Implementieren Sie die Variable wie in „Implementierung von skalaren nichtlogischen Elementen“ auf Seite 518 beschrieben.

Der Menüpunkt **Element** → **Use automatic Implementation** ist nun verfügbar. Er ist nicht länger markiert.

Die automatische Implementierung einer methoden-/prozesslokalen Variablen kann nur im Implementierungseeditor der Komponente wieder aktiviert werden.

### **Aktivieren der automatischen Implementierung:**

---

- Wählen Sie in der Fläche „Elements“ die lokale Variable aus.
- Wählen Sie **Element** → **Use automatic Implementation**

*oder*

- klicken Sie mit der rechten Maustaste auf den Namen der Variablen und wählen Sie aus dem Kontextmenü **Use automatic Implementation**.

Damit ist die automatische Implementierung der methoden-/prozesslokalen Variablen aktiviert. Der Menüpunkt **Element** → **Use automatic Implementation** ist markiert und ausgegraut.

Die automatische Implementierung wird deaktiviert, indem die Variable explizit implementiert wird.

#### 4.12.4 Implementierungen für temporäre Variablen

---

An den Ausgängen von Operatoren und komplexen Modellelementen können temporäre Variablen eingefügt werden. Für diese temporären Variablen bestimmt der Codegenerator die Implementierung automatisch: Sobald der Variablen in einem Modell erstmalig eine implementierte Größe zugewiesen wird, erhält die temporäre Variable die entsprechende Konvertierungsformel und den entsprechenden Wertebereich. Der Implementierungsdatentyp wird dann so gewählt, dass er mit der Konvertierungsformel und dem Wertebereich übereinstimmt.

##### **Hinweis**

*Das Einfügen temporärer Variablen in einen mathematischen Ausdruck wirkt sich nicht auf die Generierung mathematischer Operationen für diesen Ausdruck aus. Temporäre Variablen sollten nicht in unterschiedlichen Verzweigungen des Kontrollflusses (z.B. bei den Zweigen einer IF-Abfrage) verwendet werden. Das Ergebnis und die Implementierung (z.B. Quantisierung) können für unterschiedliche Verzweigungen voneinander abweichen. Dies kann zu ernsthaften arithmetischen Fehlern im generierten Code führen.*

#### 4.12.5 Implementierung von Arrays, Matrizen und Tabellen

---

Arrays und Matrizen haben Implementierungen, die wie die Implementierung eines skalaren Elements definiert sind. Eine 1D-Tabelle hat zwei Implementierungen, eine für den Eingabe- und eine für den Ausgabewert. Eine 2D-Tabelle hat drei Implementierungen, zwei für die Eingaben und eine für den Ausgabewert.

## Definieren der Implementierung von Tabellen:

- Öffnen Sie den Implementierungseditor für eine 1D- oder 2D-Tabelle.

Implementation for: Tab\_2D Project: Module\_copyimpl\_DEFAULT...

Value | X Distribution | Y Distribution | Additional Information

Use Implementation Type

Implementation

Transformation

Formula: ident

Conversion:  $f(\text{phys}) = \text{phys}$

Quantization: Calculated 1.0 Qu. Exp. 0

Master

Model  Implementation

Model

Type: cont

Min: -2147483648.0

Max: 2147483647.0

Implementation

Type: int32

Min: -2147483648

Max: 2147483647

Zero not included

Implementation Interval Adaptation

Limit Assignments

Limit to maximum bit length: Automatic

Memory Location: Default

Cache Element: Automatic

Consistency

Source	Conflict

Auto Correction OK Cancel

Die Register „Value“, „X Distribution“ und „Y Distribution“ (nur 2D-Tabelle) oben im Editor bestimmen, welche Implementierung bearbeitet wird.

### Hinweis

Wenn Sie eine Festkennlinie oder ein Festkennfeld bearbeiten, achten Sie darauf, dass für Offset und Abstand der Achsenpunkte (s. Seite 500) ganzzahlige Werte eingegeben sind. Andernfalls kommt es zu Fehlern.

- Wählen Sie das Register für die Implementierung, die Sie bearbeiten wollen.



- Bearbeiten Sie die Implementierung.  
Dies funktioniert wie die Bearbeitung der Implementierung eines skalaren Elements (siehe Kapitel 4.12.2 auf Seite 518).
- Wählen Sie ein anderes Register, um eine andere Implementierung zu bearbeiten.
- Klicken Sie auf **OK**.

#### 4.12.6 Implementierung von logischen Elementen

##### **Spezifizieren einer Implementierung für ein logisches Baselement:**

- Öffnen Sie den Implementierungseditor für das logische Element wie in „Implementierungseditor für Baselemente öffnen:“ auf Seite 518 beschrieben.

The screenshot shows the 'Implementation for: log' dialog box. The title bar indicates the project is 'Module\_DEFAULT>PC<'. The dialog is divided into several sections:

- Value** and **Additional Information** tabs.
- Use Implementation Type**: A checkbox that is currently unchecked.
- Implementation** section:
  - Transformation**: Includes a 'Formula' dropdown, 'Conversion' set to 'not applicable', and 'Quantization' with 'Calculated' set to '1.0' and 'Qu. Exp.' set to '1'.
  - Master**: Two radio buttons, 'Model' (unchecked) and 'Implementation' (checked).
  - Model** sub-section: 'Type' is 'log', 'Min' is '0', and 'Max' is '1'.
  - Implementation** sub-section: 'Type' is 'int8', 'Min' is '0', 'Max' is '1', and a 'Zero not included' checkbox is unchecked.
- Implementation Interval Adaptation**:
  - 'Limit Assignments' checkbox is checked.
  - 'Limit to maximum bit length' checkbox is unchecked, with a dropdown set to 'Automatic'.
- Memory Location**: A dropdown menu set to 'Default'.
- Cache Element**: A dropdown menu set to 'Automatic'.
- Consistency**: A table with two columns, 'Source' and 'Conflict', and an empty row below.
- At the bottom, there are three buttons: 'Auto Correction', 'OK', and 'Cancel'.

Bis auf die Kombikästchen „Type“, „Memory Location“ und „Cache Locking“ sind alle Felder im Register „Value“ deaktiviert, da sie für die Implementierung logischer Elemente keine Rolle spielen.

- Wählen Sie im Kombikästchen „Type“ den Implementierungsdatentyp aus.  
Ihnen stehen die Typen `bit`, `int8`, `int16`, `int32`, `uint8`, `uint16` und `uint32` zur Verfügung.  
Das logische Element wird durch eine Variable des ausgewählten Datentyps dargestellt.
- Wählen Sie im Kombikästchen „Memory Location“ den Speicherbereich, in dem das Element abgelegt wird.  
Die vorhandenen Auswahlmöglichkeiten hängen vom aktuellen Target ab.  
Die Einstellung ist nur für Mikrocontroller-Targets relevant und wird bei Experimentalcode ignoriert.
- Geben Sie im Register „Additional Info“ Informationen für Ihren Codegenerator ein.  
Diese Informationen werden nur nach Erfordernis ausgewertet. Die genaue Art der Informationen, die Sie hier eingeben können, ist von Ihrem Target und vom Codegenerator abhängig.

## 4.12.7 Implementierung von Enumerationen

### Spezifizieren einer Implementierung für eine Enumeration:

- Öffnen Sie den Implementierungseditor für die Enumeration wie in „Implementierungseditor für Basiselemente öffnen:“ auf Seite 518 beschrieben.

The screenshot shows the 'Implementation for: enum\_1' dialog box. The 'Value' tab is active. The 'Use Implementation Type' checkbox is disabled. The 'Implementation' section has a 'Transformation' sub-section with 'Formula' (dropdown), 'Conversion' (not applicable), and 'Quantization' (Calculated, undefined, Qu. Exp. 1). The 'Master' section has 'Model' selected. The 'Model' section has 'Type' (enum), 'Min' (0), and 'Max' (3). The 'Implementation' section has 'Type' (uint8), 'Min' (0), 'Max' (3), and 'Zero not included' (checkbox). The 'Implementation Interval Adaptation' section has 'Limit Assignments' checked and 'Limit to maximum bit length' (Automatic). The 'Memory Location' dropdown is set to 'Default'. The 'Cache Element' dropdown is set to 'Automatic'. The 'Consistency' table has 'Source' and 'Conflict' columns. At the bottom are 'Auto Correction', 'OK', and 'Cancel' buttons.

Bis auf die Kombikästchen „Memory Location“ und „Cache Locking“ sind alle Felder im Register „Value“ deaktiviert, da sie für die Implementierung von Enumerationen keine Rolle spielen.

- Wählen Sie im Kombikästchen „Memory Location“ den Speicherbereich, in dem das Element abgelegt wird.  
Die vorhandenen Auswahlmöglichkeiten hängen vom aktuellen Target ab.  
Die Einstellung ist nur für Mikrocontroller-Targets relevant und wird bei Experimentalcode ignoriert.
- Geben Sie im Register „Additional Info“ Informationen für Ihren Codegenerator ein.  
Diese Informationen werden nur nach Erfordernis ausgewertet. Die genaue Art der Informationen, die Sie hier eingeben können, ist von Ihrem Target und vom Codegenerator abhängig.

#### 4.12.8 Methoden- und Prozessimplementierungen

---

In ASCET können Implementierungen nicht nur für Datensätze sondern auch für die in Modulen und Klassen definierten Prozesse oder Methoden erfolgen. Damit ist es möglich, das Gesamtverhalten Ihres Systems zu verbessern.

Die Implementierung von Methoden und Prozessen wird aus dem Spezifikationseditor bearbeitet. Sie lässt sich nicht direkt aus dem Projekteditor bearbeiten. Prozess- und Methodenimplementierung stehen sowohl für Blockdiagramme als auch für ESDL-Komponenten zur Verfügung.

Die Implementierung eines Prozesses (oder einer Methode) bestimmt, wie der Prozess im generierten C-Code dargestellt und wo er während der Ausführung des Programm angeordnet ist.

##### **Bearbeiten einer Prozess-/Methodenimplementierung:**

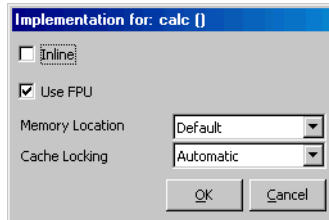
---

- Öffnen Sie den Spezifikationseditor für die zu bearbeitende Klasse bzw. das Modul.
- Wählen Sie in der Fläche „Diagrams“ einen Prozess bzw. eine Methode aus.
- Wählen Sie **Diagram** → **Edit Implementation**

*oder*

- wählen Sie aus dem Kontextmenü **Edit Implementation**.

Der Implementierungseditor für die Methode bzw. den Prozess wird geöffnet.



- Aktivieren Sie die Option **Inline**, damit der Programmcode der Methode direkt in den Programmcode des Modells eingefügt wird.

### Hinweis

Für Prozesse ist die Option **Inline** nicht verfügbar.

Auf diese Weise entfällt der Funktionsaufruf für die Methode. Dadurch wird die Laufzeit optimiert, allerdings auf Kosten von zusätzlichem Speicherbedarf, wenn die Methode mehrfach verwendet wird.

- Aktivieren Sie die Option **Use FPU**, wenn Sie bei einem Taskwechsel die Register der Floating Point Unit Ihres Mikrocontroller-Targets sichern wollen.

Diese Option ist Teil des Betriebssystems ERCOS<sup>EK</sup>. Sie ist nur in Verbindung mit einigen Mikrocontroller-Targets von Bedeutung; Details finden Sie im ASCET-SE V5.2 Benutzerhandbuch sowie dem ERCOS<sup>EK</sup> User's Guide.

- Wählen Sie aus der Liste „Memory Location“ den Speicherbereich aus, in dem der Code laufen soll.

Allgemein gilt, dass zwecks besseren Leistungsverhaltens langsame und selten aufgerufene Prozesse in einem externen Speicher,

während schnelle und häufig aufgerufene Prozesse in einem internen Speicher laufen sollten.

#### 4.12.9 Implementierung von Argumenten und Rückgabewert einer Methode

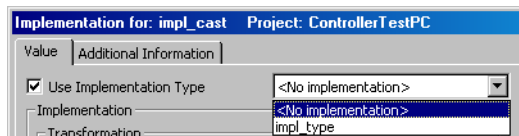
Die Argumente und der Rückgabewert einer Methode werden genau wie Elemente desselben Typs implementiert (siehe Kapitel 4.12.1, 4.12.2, 4.12.5, 4.12.6 und 4.12.7).

Bei skalaren und logischen Argumenten und Rückgabewerten und Argumenten/Rückgabewerten vom Typ `<enumeration>` ist das Kombikästchen „Memory Location“ ausgegraut. Diese Elemente werden in der Speicherklasse `STACK` abgelegt.

Werden Referenzen als Argumente und Rückgabewerte benutzt, d.h. der Typ `<array[*]>`, `<mat[*]>` oder `<user defined>` gewählt, kann im Kombikästchen „Memory Location“ eine Speicherklasse angegeben werden. Diese Speicherklasse gilt in diesem Fall *nicht* für die Referenz, sondern für das Ziel der Referenz.

#### 4.12.10 Implementierung von Implementation-Casts

Implementation-Casts können wie skalare Elemente implementiert werden (siehe Kapitel 4.12.2). Neu ist hier nur die Möglichkeit, die Objekte überhaupt nicht zu implementieren – durch die Auswahl von `<No implementation>` im Kombikästchen neben **Use Implementation Type**. `<No implementation>` ist auch die Standardeinstellung für neu angelegte Implementation-Casts.



##### **Hinweis**

*Wenn für einen Implementation-Cast `<No implementation>` gewählt ist, so wird dieser bei jedem Codegenerierungslauf als nicht existent behandelt.*

#### 4.12.11 Operatorimplementierungen

---


In früheren Versionen von ASCET konnten in Blockdiagrammen auch für Operatoren Implementierungen definiert werden. Damit war es möglich, die Genauigkeit von Operationen in Festkomma-Arithmetik zu verbessern.

##### **Hinweis**

Die Implementierungsoptionen **Limit to maximum bit length** und **Zero not included** sind an die Stelle der alten Operatorimplementierung getreten. Darüber hinaus können Umquantisierungen in verketteten arithmetischen Operationen mit Hilfe der Implementation-Casts eingefügt werden, ohne zusätzlichen Bedarf an Speicherplatz zu erzeugen.

Aus diesem Grund können keine neuen Operatorimplementierungen angelegt werden. Vorhandene Operatorimplementierungen aus älteren Projekten können angeschaut, durch Implementation-Casts ersetzt (siehe „Automatische Konvertierung von Operatorimplementierungen“) oder entfernt, aber nicht bearbeitet werden.

Operatoren besitzen keinen Namen in der Grafik und erscheinen nicht in der Liste „Elements“. Somit ist ein Zugriff auf den Implementierungs-Betrachter für einen Operator nur über sein Symbol im Grafikblock möglich.

Ist eine Implementierung spezifiziert, ist der Operator in der Grafik durch eine kleine Linie in der linken oberen Ecke gekennzeichnet (). So ist für den Benutzer ersichtlich, wo ein Eingriff erfolgt ist.

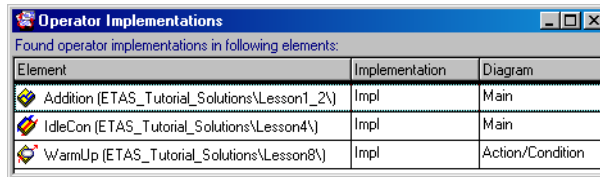
Zum leichteren Auffinden von Operatorimplementierungen in der gesamten Datenbank verwenden Sie den Befehl **Tools** → **Database** → **List Operator Implementations** im Komponentenmanager.

## Suchen nach Operatorimplementierungen:

---

- Wählen Sie im Komponentenmanager **Tools** → **Database** → **List Operator Implementations**.

Die Datenbank wird durchsucht. Je nach Größe kann die Suche etliche Sekunden dauern. Die gefundenen Operatorimplementierungen werden im Fenster „Operator Implementations“ angezeigt.



Element	Implementation	Diagram
Addition (ETAS_Tutorial_Solutions\Lesson1_2\)	Impl	Main
IdleCon (ETAS_Tutorial_Solutions\Lesson4\)	Impl	Main
WarmUp (ETAS_Tutorial_Solutions\Lesson8\)	Impl	Action/Condition

Die Spalte „Element“ enthält alle Komponenten/Projekte, in denen Operatorimplementierungen enthalten sind. Die Spalte „Implementation“ gibt den Namen der jeweiligen Implementierung an, und die Spalte „Diagram“ gibt das Diagramm an, in dem die Operatorimplementierung enthalten ist.

- Doppelklicken Sie in der Spalte „Element“ auf eine Komponente.

Der Editor für die Komponente wird geöffnet. Die betroffenen Operatoren sind markiert. Sie können die Implementierungen ansehen oder entfernen.

## Entfernen einer einzelnen Operatorimplementierung:

---

- Öffnen Sie das betreffende Blockdiagramm.
- Klicken Sie mit der rechten Maustaste auf einen Operator mit Implementierung und wählen Sie **Implementation** → **Reset** aus dem Kontextmenü.

Die Implementierung wird gelöscht, die Linie in der Ecke des Operators wird entfernt.



## Entfernen aller Operatorimplementierungen einer Komponente/der Datenbank:

---

- Wählen Sie im Komponentenmanager im Feld „1 Database“ eine Komponente.
- Wählen Sie **Component** → **Remove Operator Implementation** → **Flat**, um die Operatorimplementierungen der Komponente zu entfernen.
- Wählen Sie **Component** → **Remove Operator Implementation** → **Recursive**, um die Operatorimplementierungen der Komponente selbst und der darin referenzierten Komponenten zu entfernen.

*Oder*

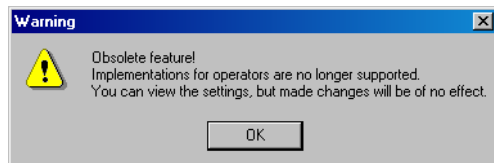
- Wählen Sie **Tools** → **Database** → **Convert** → **Reset Operator Implementations**, um alle Operatorimplementierungen der ganzen Datenbank zu entfernen.

## Ansehen einer Operatorimplementierung:

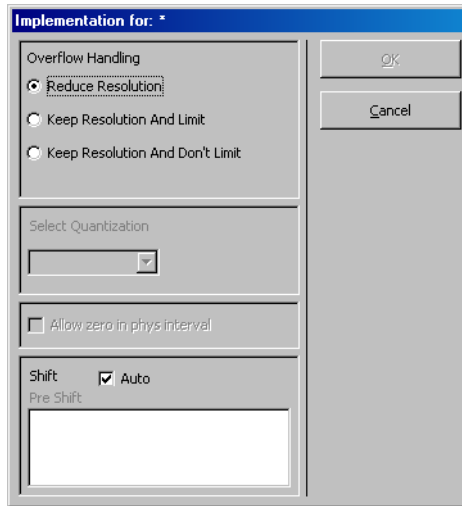
---

- Öffnen Sie das betreffende Blockdiagramm.
- Klicken Sie mit der rechten Maustaste auf den Operator, dessen Implementierung Sie betrachten wollen, und wählen Sie **Implementation** → **View** aus dem Kontextmenü.

Eine Warnmeldung erscheint mit dem Hinweis, dass Operatorimplementierungen nicht mehr unterstützt werden.



- Klicken Sie **OK**, um das Fenster „Implementation for“ zu öffnen.



Das Fenster dient lediglich der Ansicht, Sie können keine Einstellungen ändern.

- Klicken Sie auf **OK**.

Die Implementierung für einen mathematischen Operator bestimmt die Verfahrensweise bei einem Überlauf und/oder die Quantisierung (d.h. die Genauigkeit) im Ergebnis. Wie bei Variablen können diese Informationen bei den einzelnen Implementierungen für das betreffende Modul unterschiedlich sein. Eine Erläuterung der Informationen findet sich nachstehend für jeden Operator.

### Hinweis

*Der ganzzahlige Codegenerator behandelt die Grundoperationen völlig unterschiedlich. Somit sind zusätzliche Informationen auch von der speziellen Operation abhängig.*

### **Addition und Subtraktion:**

Verfahrensweise bei Überlauf:

- *Reduce Resolution* vermeidet einen Überlauf durch Rechtsverschiebung bei der Eingaben. Die Verschiebung wird automatisch ermittelt.

- *Keep Resolution And Limit* führt keine Verschiebung aus und nutzt, wenn vorhanden, mathematische Serviceroutinen mit Clipping zum Begrenzen.
- *Keep Resolution And Don't Limit* führt keine Verschiebung aus, nutzt normale Arithmetik und lässt damit einen Überlauf zu. Diese Variante wird beispielsweise für Zähler genutzt, die zyklisch überlaufen sollen.

Wahl der Quantisierung:

- *Auto* wählt eine Quantisierung unter Verwendung einer Optimierungsstrategie.
- *1* wählt Quantisierung und Datentyp der ersten Eingabe.
- *2* wählt Quantisierung und Datentyp der zweiten Eingabe.

### **Multiplikation:**

Verfahrensweise bei Überlauf:

- *Reduce Resolution* vermeidet einen Überlauf durch Rechtsverschiebung beider Eingaben. Die Verschiebung wird automatisch ermittelt.
- *Keep Resolution And Limit* führt keine Verschiebung aus und nutzt, wenn vorhanden, mathematische Serviceroutinen zum Begrenzen.
- *Keep Resolution And Don't Limit* führt keine Verschiebung aus, nutzt normale Arithmetik und lässt damit einen Überlauf zu.

Vorverschiebung:

Beide Operanden können vor der Operation verschoben werden, um einem Überlauf vorzubeugen und jeden Operanden auf numerisch sinnvolle Weise neu zu skalieren. Bei der Vorverschiebung kann auch der Wertebereich in vollem Umfang ausgenutzt werden. Für beide Operanden sind ganze Zahlen zwischen -31 und 31 möglich; eine positive Zahl zeigt Linksverschiebung, eine negative Zahl Rechtsverschiebung an. Null bedeutet, dass keine Verschiebung stattfindet. Es gibt folgende Möglichkeiten:

- Die aktivierte Option *Auto* zeigt an, dass keine Vorverschiebung stattfindet.
- Wenn *Auto* deaktiviert ist, wird die Vorverschiebung gemäß der Eingaben im Feld „Pre Shift“ durchgeführt, und zwar wie folgt:
  - Die Zeile 1 bestimmt eine Verschiebung für den ersten Operanden (von -31 bis 31).
  - Die Zeile 2 bestimmt eine Verschiebung für den zweiten Operanden (von -31 bis 31).

### **Division:**

Verfahrensweise bei Überlauf:

- *Reduce Resolution* vermeidet einen Überlauf durch Rechtsverschiebung beider Eingaben. Die Verschiebung wird automatisch ermittelt.
- *Keep Resolution And Limit* führt keine Verschiebung aus und nutzt, wenn vorhanden, mathematische Serviceroutinen zum Begrenzen.
- *Keep Resolution And Don't Limit* führt keine Verschiebung aus, nutzt normale Arithmetik und lässt damit einen Überlauf zu.

Die Option **Allow zero in phys. interval** konnte aktiviert werden, wenn im generierten Code darauf verzichtet werden sollte, auf Division durch Null zu prüfen, obwohl der Wertebereich die Null enthält. Es handelt sich also um eine Zusicherung an den Codegenerator, dass der Anwender selbst dafür sorgt, dass der Nenner der Division nicht Null wird.

### **Hinweis**

*Ein falscher Gebrauch dieser Option kann zu schweren Ausnahmefehlern im Steuergerät führen.*

Vorverschiebung:

Gibt an, wenn der Zähler zwecks größerer numerischer Genauigkeit automatisch durch Linksverschieben zu maximieren ist. Es gibt folgende Möglichkeiten:

- *Auto* führt eine automatische Linksverschiebung des Zählers durch.
- Wenn *Auto* deaktiviert ist, wird die Vorverschiebung gemäß der Eingaben im Feld „Pre Shift“ durchgeführt.
  - Die Zeile 1 bestimmt eine Verschiebung für den ersten Operanden (von -31 bis 31).
  - Die Zeile 2 bestimmt eine Verschiebung für den zweiten Operanden (von -31 bis 31).

### **Multiplexer, Maximum, Minimum:**

Wahl der Quantisierung:

- *Auto* wählt eine Quantisierung unter Verwendung einer Optimierungsstrategie.
- 1 wählt Quantisierung und Datentyp der ersten Eingabe.
- 2 wählt Quantisierung und Datentyp der zweiten Eingabe.

Da bei diesen Operationen keine Berechnungen durchgeführt werden, ist kein Umgang mit einem Überlauf erforderlich.

## Automatische Konvertierung von Operatorimplementierungen

Sie können Operatorimplementierungen in älteren Modellen löschen (s. Seite 544) oder automatisch durch die neu eingeführten Implementation-Casts (Kapitel „Implementation-Casts“ im ASCET-Referenzhandbuch) ersetzen. Das automatische Ersetzen geht allerdings nur für die gesamte Datenbank, nicht für einzelne Komponenten.

**Regeln für die automatische Konvertierung:** Die folgenden Bedingungen müssen erfüllt sein, damit eine Operatorimplementierung automatisch konvertiert werden kann.

- Die Operatorimplementierung darf keine andere Quantisierung als **Auto** enthalten (Addition, Subtraktion, MIN, MAX und MUX; s. z.B. Seite 546).

### Hinweis

*Für MIN-, MAX- und MUX-Operatoren ist dies die einzige Bedingung, die für eine automatische Konvertierung erfüllt sein muss. Die weiteren Bedingungen gelten nur für +, -, \*, /.*

- Der Operatorausgang muss verbunden sein.
- Der Operatorausgang darf nur mit primitiven Elementen verbunden sein.  
Er darf weder mit einer Komponente noch mit einem Operator oder einer Hierarchie verbunden sein.
- Wenn ein Implementation-Cast mit dem Operatorausgang verbunden ist, muss für diesen Implementation-Cast im Kombikästchen neben der Option **Use Implementation Type** etwas anderes als `<No implementation>` ausgewählt sein.
- Die Operatorimplementierung darf keine spezielle Vorverschiebung beinhalten (Multiplikation und Division; s. Seite 547).
- Wenn der Operator ein Divisionsoperator ist und in der Operatorimplementierung die Option **Allow zero in phys. interval** aktiviert ist, gelten zusätzlich folgende Regeln für den Nenner-Eingang:
  - Der Nenner-Eingang muss verbunden sein.
  - Der Nenner-Eingang darf nur mit primitiven Elementen verbunden sein.  
Er darf weder mit einer Komponente noch mit einem Operator oder einer Hierarchie verbunden sein.

- Wenn ein Implementation-Cast mit dem Nenner-Eingang verbunden ist, muss für diesen Implementation-Cast im Kombikästchen neben der Option **Use Implementation Type** etwas anderes als `<No implementation>` ausgewählt sein.

Wenn eine dieser Bedingungen in keiner Implementierung der Komponente (s. Kapitel 4.12.1) erfüllt ist, muss der betreffende Operator manuell konvertiert werden.

### **Operatorimplementierung durch Implementation-Cast ersetzen:**

---

- Wählen Sie im Komponentenmanager **Tools → Database → Convert → Operator Implementations to Impl. Casts**.

Die Operatorimplementierungen der gesamten Datenbank werden entsprechend der obigen Regeln in Implementation-Casts konvertiert.

Wenn ein Operator (ausgenommen MIN, MAX, MUX) automatisch konvertiert werden kann, geschieht folgendes:

- Auf jeder Verbindung des Operatorausgangs wird ein Implementation-Cast angelegt.
- Wenn der Operator ein Divisionsoperator ist und in der Operatorimplementierung die Option **Allow zero in phys. interval** aktiviert ist, wird auf der Verbindung zum Nenner-Eingang ein Implementation-Cast angelegt.
- Für jeden Implementation-Cast am Ausgang eines implementierten Operators wird die Implementierungsinformation des nachfolgenden Elements übernommen.  
Das gilt nicht für den Modelltyp; dieser ist für Implementation-Casts immer `cont`.
- Für Implementation-Casts, die am Nenner-Eingang eines Divisionsoperators eingefügt wurden, wird die Implementierungsinformation (ausgenommen der Modelltyp) vom vorhergehenden Element übernommen.

#### **Hinweis**

*Für die Implementierungen der Komponente (s. Kapitel 4.12.1), in denen der Operator keine Implementierung besitzt, wird für neu angelegte Implementation-Casts `<No implementation>` ausgewählt.*

- Die Überlaufbehandlung wird nach folgendem Schema konvertiert:

Implementation-Cast: Operator- implementierung:	Limit to maximum bit length	Reduce Resolution	Keep Resolution
Reduce resolution	X	X	
Keep resolution and limit	X		X
Keep resolution and don't limit			X

In jeder Zeile sind die Einstellungen markiert, die für den Implementation-Cast gesetzt werden, um die jeweilige Einstellung der Operatorimplementierung zu ersetzen.

- Die Operatorimplementierung wird entfernt.

Wenn ein MIN-, MAX- oder MUX-Operator automatisch konvertiert werden kann, wird lediglich die Operatorimplementierung entfernt. Es wird kein Implementation-Cast eingefügt.

Nach den oben genannten Regeln würden Implementation-Casts unter bestimmten Bedingungen mit der selben Implementierung angelegt, wie das mit ihrem Ausgang verknüpfte Element. In diesen Fällen wird kein neuer Implementation-Cast eingefügt.

Wenn ein Operator nicht automatisch konvertiert werden kann, geschieht folgendes:

- Auf jeder Verbindung des Operatorausgangs – auch mit Komponenten, Operatoren usw. – wird ein Implementation-Cast angelegt. Für diese Implementation-Casts ist in allen Implementierungen der Komponente `<No implementation>` ausgewählt.

Diesem Implementation-Cast geben Sie bei der manuellen Konvertierung der Operatorimplementierung die relevanten Implementierungsinformationen.

Wenn auf einer dieser Verbindungen ein solcher Implementation-Cast bereits existiert, wird auf dieser Verbindung kein weiterer Implementation-Cast eingefügt.

- Wenn in der Operatorimplementierung eines Divisionsoperators die Option **Allow zero in phys. interval** aktiviert ist, wird auf der Verbindung des Nenner-Eingangs ein Implementation-Cast mit `<No implementation>` angelegt.

Wenn ein solcher Implementation-Cast bereits existiert, wird kein weiterer eingefügt.

- Die Operatorimplementierung bleibt unverändert.

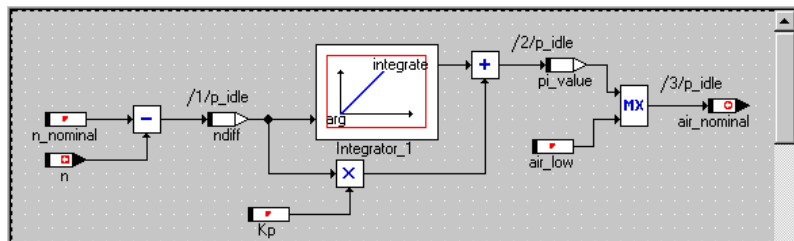
Wenn nicht alle Operatorimplementierungen in der Datenbank automatisch konvertiert werden können, wird folgender Hinweis ausgegeben:

Not all operator implementations could be replaced automatically. Please do the conversion manually.

Bestätigen Sie diese Meldung mit **OK**. Das Fenster „Operator Implementations“ (siehe Seite 544) öffnet sich; es zeigt die Komponenten an, die die verbleibenden Operatorimplementierungen enthalten. Sie können diese nun manuell konvertieren.

## 4.13 Bearbeiten des Layouts einer Komponente

Wenn Sie eine Komponente im Zeichenbereich des Blockdiagrammeditors oder des Projekteditors einfügen, so erscheint sie als ein grafischer Block, wobei die Schnittstellenelemente der Komponente durch die Eingangs- und Ausgangsanschlüsse des Blocks dargestellt werden.



Sie können das Aussehen des grafischen Blocks mit dem Layout-Editor ändern. Es ist möglich, ein Symbol hinzuzufügen, die Eingangs- und Ausgangsanschlüsse hin- und herzubewegen sowie die Größe und Farbe des Blocks zu ändern. Je nach Komponententyp können Sie auch Methoden oder Prozesse der Komponente freigeben oder sperren, um ihre öffentliche Schnittstelle zu bestimmen.

Der Layout-Editor ist aus allen Komponenteneditoren sowie dem Komponentenmanager aus aufrufbar.

### Öffnen des Layout-Editors:

- Öffnen Sie den Spezifikationseditor für die Komponente, deren Layout Sie ändern möchten.
- Wählen Sie **Component** → **Edit Layout**.

*Oder*



- Wählen Sie in der Browser-Ansicht das Register „Layout“.
- Doppelklicken Sie auf den grafischen Block.  
Der Layout-Editor für die ausgewählte Komponente wird geöffnet.

Wenn Sie eine Komponente in eine andere Komponente oder ein Projekt aufnehmen wollen, können Sie festlegen, ob das Layout der aufgenommenen Komponente im Blockdiagrammeditor oder im Projekteditor geändert werden kann (*flexibles Layout*). Global schalten Sie das flexible Layout im ASCET-Optionsfenster über die Option **Activate flexible layout** ein (siehe Abschnitt „Optionen für Blockdiagramme“ auf Seite 56). Unabhängig von dieser Option können Sie das flexible Layout lokal für jede Klasse oder jedes Modul einzeln freischalten oder sperren. Standardmäßig ist das flexible Layout global wie lokal gesperrt.

#### **Flexibles Layout aktivieren (lokal):**

---

- Wählen Sie im Layout-Editor der Komponente **Layout** → **Enable flexible layout**.

Das flexible Layout wird aktiviert, die Menüfunktion ist mit einem Häkchen markiert. Sie können nun das Layout jedes Vorkommens der Komponente in einem Blockdiagramm bearbeiten (siehe Abschnitt „Layout von aufgenommenen Komponenten“ auf Seite 241).

Um das flexible Layout zu deaktivieren, wählen Sie erneut **Layout** → **Enable flexible layout**.

#### 4.13.1 Bearbeiten des Layouts einer Klasse

---

Im Layout-Editor können Sie das Aussehen des Blockes ändern sowie einzelne Methoden freigeben und sperren. Wenn eine Methode im Layout-Editor gesperrt ist, kann sie nicht verwendet werden, wenn die Klasse durch eine andere Komponente referenziert wird.

#### **Ändern des Diagrammblocks:**

---

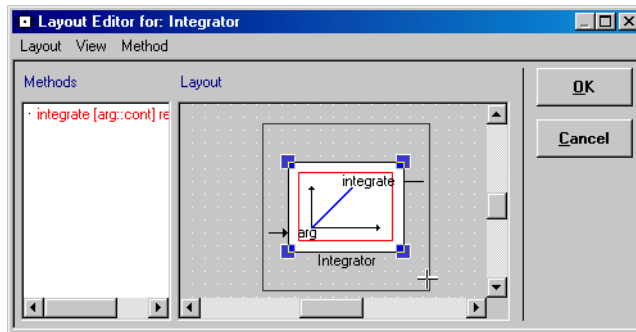
- Klicken Sie im Layout-Editor auf den Block im Zeichenbereich.

Wenn Sie ein Layout bearbeiten, das ein Symbol enthält, kann es leicht passieren, dass Sie das Symbol anstelle des Blocks auswählen – vor allem, wenn beide etwa gleich groß sind.

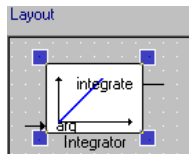
Das Symbol kann jedoch im Layout-Editor nicht verändert werden.  
Klicken Sie in diesem Fall auf die *Kante* des Blocks.

Der Block wird ausgewählt; an allen Ecken erscheinen Anfassers.

- Ziehen Sie die Anfassers, um die Größe des Blockes zu ändern.



Wenn die Größe des Blocks *kleiner* ist als die des Symbols, können Sie sie im Layout-Editor nicht mehr verändern, da Sie immer das Symbol auswählen.



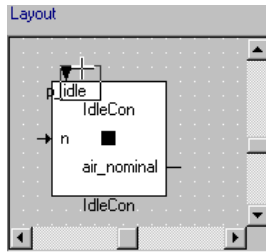
Um die Größe des Blocks zu ändern, müssen Sie die Komponente in ein Blockdiagramm aufnehmen, im Zeichenbereich ablegen, das Layout wie im Abschnitt „Layout von aufgenommenen Komponenten“ auf Seite 241 beschrieben bearbeiten und entsprechend Seite 246 als Standard definieren.

### **Anschlüsse verschieben:**

Die Standardposition für Eingänge befindet sich auf der linken, die für Ausgänge auf der rechten Seite. Die Standardposition für Methoden befindet sich auf der oberen Seite. Sie können sämtliche Anschlüsse verschieben.

- Ziehen Sie die Eingänge und Ausgänge an den Seiten des Blocks entlang in die gewünschte Position.

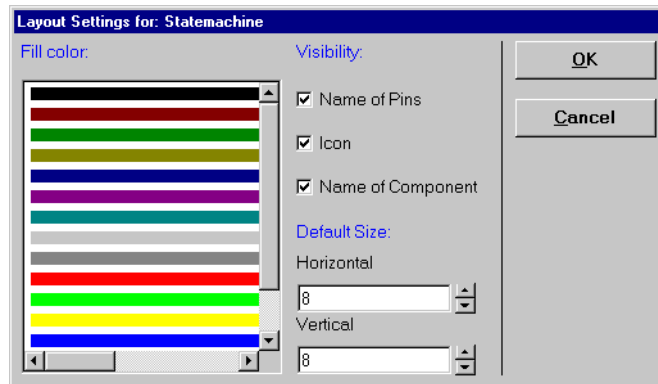
- Ziehen Sie die Methoden an den Seiten des Blocks entlang in die gewünschte Position.



Sie können einen Anschluss an keine Stelle schieben, an der sich schon ein anderer Anschluss befindet.

### Ändern der Blockattribute:

- Wählen Sie im Layout-Editor **Layout** → **Modify Attributes**, um das Dialogfenster „Layout Settings“ zu öffnen.



- Wählen Sie im Feld „Fill color“ die Füllfarbe aus.
- Stellen Sie die Blockgröße ein, indem Sie Werte aus den Feldern „Horizontal“ und „Vertical“ auswählen. Dies hat dieselbe Wirkung wie ein Ziehen der Anfassers.

- Stellen Sie die Sichtbarkeitsoptionen ein, indem Sie die Optionen aktivieren bzw. deaktivieren:
  - **Name of Pins:** Die Eingangs- und Ausgangsanschlüsse sind mit den Namen der Schnittstellenelemente gekennzeichnet.
  - **Icon:** Das durch die Komponente referenzierte Symbol ist in der Mitte des Blocks dargestellt.
  - **Name of Component:** Der Name, der der Komponente im Komponentenmanager zugeordnet wurde. Der Instanzname, d.h. der Name, der einer konkreten Instanz einer Komponente zugeordnet wird, wenn sie durch eine andere Komponente referenziert wird, kann im Zeichbereich sichtbar oder unsichtbar gemacht werden.
- Klicken Sie auf **OK**, um die Attribute für den aktuellen Block zu ändern.

Es ist möglich, die Standardattribute, die für alle neuen Blöcke in einer Datenbank gelten, zu ändern.

#### **Ändern der Standardattribute für neue Blöcke:**

---

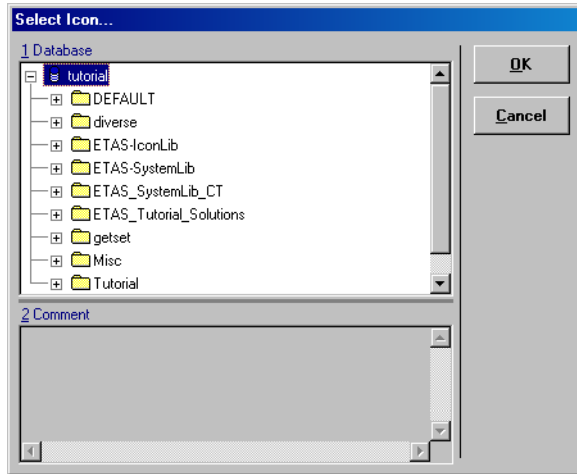
- Wählen Sie **Layout** → **Edit Default Attributes**.  
Dies bewirkt das erneute Erscheinen des Dialogfensters „Layout Settings“. Sie können dieselben Einstellungen wie oben beschrieben vornehmen, doch diese beeinflussen den aktuellen Block nicht. Die Einstellungen legen die Standardattribute fest, die auf alle neu erzeugten Komponenten angewandt werden.
- Wählen Sie **Layout** → **Set Attributes to Default**, um die Standardattribute auf den aktuellen grafischen Block anzuwenden.

Sie können ein Symbol in einer Komponente verwenden, das dann im grafischen Block dieser Komponente dargestellt wird. Symbole von ASCET werden in der Datenbank gespeichert, und auf sie kann mit Hilfe des Komponentenmanagers zugegriffen werden.

## Zuweisen eines Symbols zu einer Klasse:

---

- Wählen Sie im Layout-Editor **Layout** → **Select Icon**, um das Dialogfenster „Select Icon“ zu öffnen.



- Wählen Sie aus der Liste „1 Database“ das Symbol aus, das Sie dem Block zuweisen möchten.
- Klicken Sie auf **OK**, um das Symbol zuzuweisen.  
Das Symbol wird der Komponente im Layout-Editor zugewiesen.

Wie Sie Symbole erzeugen und bearbeiten, ist im Kapitel „Der Symboleditor“ auf Seite 600 beschrieben.

Der Layout-Editor bietet verschiedene Möglichkeiten, die Darstellung einzurichten.

## Ändern der Darstellung des Layout-Editors:

---

- Wählen Sie **View** → **Redraw**, um die Darstellung zu ändern.

- Wählen Sie **View** → **Grid**, um das Dialogfenster „Grid Option“ zu öffnen.  
Dieses Dialogfenster wird im Abschnitt „Betrachten und Drucken von Diagrammen“ auf Seite 280 beschrieben.
- Wählen Sie **View** → **Print**, um das Layout auszudrucken.
- Klicken Sie auf **OK**, um den Layout-Editor zu schließen.

Sie können die öffentliche Schnittstelle ändern, indem Sie den öffentlichen Zugriff auf Methoden und Variable freigeben oder sperren.

#### **Freigeben und Sperren von Methoden:**

---

- Wählen Sie in der Liste „Methods“ des Layout-Editors die Methode aus, die Sie freigeben oder sperren möchten.
- Wählen Sie **Method** → **Disable**, um den öffentlichen Zugriff auf die ausgewählte Methode zu sperren.
- Wählen Sie **Method** → **Enable**, um den öffentlichen Zugriff auf die ausgewählte Methode freizugeben.

Standardmäßig sind alle Methoden freigegeben. Sie sind in der Liste „Methods“ des Layout-Editors rot dargestellt. Eine gesperrte Methode ist im Layout-Feld nicht mehr sichtbar und in der Liste „Methods“ schwarz dargestellt.

#### 4.13.2 Bearbeiten des Layouts anderer Komponenten

---

- *Module*: Das Bearbeiten des Layouts eines Moduls erfolgt ebenso wie bei Klassen, mit dem Unterschied, dass Prozesse anstelle von Methoden freigegeben und gesperrt werden können.
- *Zeitkontinuierliche Blöcke*: Im Layout-Editor können keine Methoden freigegeben oder gesperrt werden. Das Ändern des Aussehens des Blocks erfolgt ebenso wie bei Klassen.
- *Zustandsautomaten* und *Bedingungstabellen*: Ebenso wie bei Klassen.
- *Boolesche Tabellen*: Wie bei Klassen, nur können keine Methoden gesperrt werden.

## 4.14 Arithmetische Dienste

---

ASCET bietet dem Benutzer die Möglichkeit, so genannte *Arithmetische Dienste* (engl. Arithmetic Services) zu definieren und zu nutzen, um elementare Operationen wie z.B. Additionen zu optimieren und mit speziellen Eigenschaften wie Wertebegrenzungen oder Überlaufbehandlungen zu erweitern.

Ein Arithmetischer Dienst ist eine C-Funktion, die vom Benutzer bereitgestellt und anstelle von Standardoperationen aus dem von ASCET generierten Code heraus aufgerufen wird. Beispielsweise wird anstelle der normalen Anweisung für eine Addition,

```
c = a + b;
```

ein Funktionsaufruf

```
c = add(a, b);
```

generiert.

Der Codegenerator kann u.a. folgende Standardoperationen durch benutzerdefinierte Aufrufe von Servicefunktionen ersetzen:

- Addition, Subtraktion, Multiplikation und Division
- Negation und Absolutwertbildung
- Stützstellensuche und Stützstellen-Interpolation
- kombinierte Operationen wie Multiplikation mit darauf folgender Division

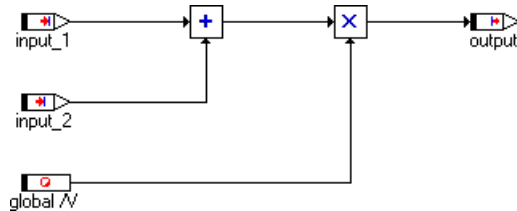
Arithmetische Dienste können für alle physikalischen Spezifikationstypen (ESDL, Blockdiagramme) und alle komplexen Elemente (Klassen, Module, Zustandsautomaten) verwendet werden und werden für alle Target-Typen (Mikrocontroller-Targets, Experimentaltargets (Rapid Prototyping) und PC) sowohl im Offline- als auch im Online-Experiment mit einem der Implementierungscode-Experimente (s. Seite 443) unterstützt.

ASCET übernimmt die Informationen über das Vorhandensein und die Art der Services aus einer Datei `services.ini`, die für jedes Target im entsprechenden Targetverzeichnis (`Ascet5.2\target\<target>`) abgelegt ist. Diese Dateien folgen dem Windows-Standard für `*.ini`-Dateien; sie können mit einem gewöhnlichen Texteditor oder dem mitgelieferten AS-Editor (siehe Kapitel 4.14.5) erstellt und bearbeitet werden.

Die Arithmetischen Dienste selbst werden nicht von ASCET bereitgestellt. Der Benutzer hat hier die Möglichkeit, speziell für seinen Anwendungsfall optimierte Dienstfunktionen als Bibliothek, Objektcode, C-Code oder mit Hilfe der C-Code-Editoren von ASCET bereitzustellen.

#### 4.14.1 Funktionsweise eines Arithmetischen Dienstes

ASCET bietet neben der Beschreibung von Klassen, Modulen etc. in C-Code auch die Möglichkeit, diese grafisch (als Blockdiagramm) oder in einer abstrakten Sprache (ESDL) zu beschreiben. Um aus diesen Beschreibungen ausführbaren Code für ein Steuergerät oder ein Experiment zu generieren, müssen die Beschreibungen zunächst in C-Code umgewandelt werden. Zu diesem Zweck verfügt ASCET über einen Codegenerator. Um zu verstehen, wozu der Codegenerator einen Arithmetischen Dienst benötigt und was dieser bewirkt, soll ein kleines Beispiel herangezogen werden. .



Im obigen Blockdiagramm werden zwei Operationen grafisch beschrieben, eine Addition und eine Multiplikation. Um diese Funktion in C-Code umzuwandeln gibt es nun zwei Möglichkeiten.

1. durch Verwenden der Standard-Operationen + bzw. \* von C
2. durch Verwenden von eigenen Funktionen für die Operationen Addition und Multiplikation

Für die Addition sähe die Standardoperation so aus:

```
output := input_1 + input_2;
```

Als zweite Möglichkeit wäre diese Version denkbar:

```
uint8 add(uint8 summand_1, uint8 summand_2)
{
    return summand_1 + summand_2
}
...
output := add(input_1, input_2);
```

Standardmäßig würde ASCET den normalen Operator + für die einfache Addition verwenden. Über die Angabe eines Arithmetischen Dienstes kann man aber festlegen, dass die zweite Variante verwendet werden soll.

Um dies zu tun, müssen zwei Schritte durchgeführt werden:



1. Bereitstellen des Funktionscodes für die Funktion
 

```
uint8 add(uint8 summand_1, uint8 summand_2).
```

 Dies erfolgt typischerweise außerhalb von ASCET, z.B. in Form einer Funktionsbibliothek, kann aber auch innerhalb des Modells geschehen.
2. Definition des zu verwendenden Dienstaufrufs.

Für die einfache Addition aus dem obigen Beispiel sieht diese Definition so aus:

```
+|u8|u8|u8 = add(%i1%, %i2%)
```

Die Syntax dieser Definition ist in den folgenden Abschnitten erläutert.

Aufgrund dieser Definition erzeugt der Codegenerator für jede Addition von zwei `uint8`-Werten, die zu einem `uint8`-Ergebnis führt, anstelle der Standardoperation „+“ den Funktionsaufruf `add(%i1%, %i2%)`. Die formalen Argumente `%i1%` und `%i2%` werden dabei durch die entsprechenden konkreten Argumente (z.B. `input_1` und `input_2`) ersetzt.

#### 4.14.2 Definition von Arithmetischen Diensten

---

Arithmetische Dienste bestehen aus zwei Teilen: der Definition des Dienstaufrufs und der Bereitstellung des Funktionscodes. Während der Funktionscode keinen festen Regeln folgen muss, außer dass es sich um gültigen C-Code handelt, der ASCET zur Verfügung gestellt werden muss (siehe Kapitel 4.3.3), muss die Definition dieser Dienstaufrufe in einer für ASCET verständlichen Form erfolgen. Eine Definition des Dienstaufrufs für einen Arithmetischen Dienst folgt einer strengen Syntax:

```
operation|opType1|opType2|opType3|resType=function
```

Jede Definition setzt sich aus zwei Teilen zusammen:

- Funktionsschlüssel
  - `operation|opType1|opType2|opType3|resType`
- Funktionsaufruf in C-Syntax
  - `function`

Der Funktionsschlüssel dient der eindeutigen Zuordnung einer Standardoperation zu einem Arithmetischen Dienst. Der Funktionsaufruf stellt exakt den durch den Codegenerator zu verwendenden Funktionsaufruf dar.

Jede theoretisch mögliche Standardoperation wird durch exakt einen Schlüssel beschrieben, und es darf nur einen definierten Dienst pro Schlüssel geben.

##### *Funktionsschlüssel*

---

Ein Funktionsschlüssel wird stets nach folgender Syntax gebildet:

```
operation|opType1|opType2|opType3|resType
```

Die einzelnen Elemente des Schlüssels haben dabei folgende Bedeutung:

- `operation` - eine Zeichenkette, die eindeutig die entsprechende Operation benennt;
- `opType` - eine Zeichenkette, die eindeutig den Typ der benötigten Operanden beschreibt;
- `resType` - eine Zeichenkette, die eindeutig den Typ des Ergebnisses der Operation beschreibt.

### Zulässige Arithmetische Dienste

Für alle in ASCET vorkommenden Berechnungen gibt es einen passenden Arithmetischen Dienst, den der Anwender nach seinen Wünschen anpassen und optimieren kann. Die nachfolgende Tabelle gibt alle von ASCET verstandenen Arithmetischen Dienste, die Anzahl und den Typ der Parameter (O = Operation; 1,2,3 = Operand 1, 2, 3; R = Resultat) sowie die Bedeutung der Funktion an. Zusätzlich ist eine Standardfunktionsdefinition angegeben, wie sie vom AS-Editor (siehe Kapitel 4.14.5) automatisch erzeugt werden kann.

<b>Funktions-schlüssel</b>	<b>Standard-Funktionsdefinition</b>	<b>Parameter</b>	<b>Bedeutung</b>
<code>abs * *</code>	<code>abs_%t1%_tr% (%i1%)</code>	O 1 R	Bildung des Absolutwertes
<code>neg * *</code>	<code>neg_%t1%_tr% (%i1%)</code>	O 1 R	Negation eines Wertes
<code>+ * * *</code>	<code>add_%t1%%t2%_tr% (%i1%, %i2%)</code>	O 1 2 R	Addition zweier Werte
<code>+1 * * *</code>	<code>addl_%t1%%t2%_tr% (%i1%, %i2%)</code>	O 1 2 R	Addition zweier Werte mit Sättigung <sup>a</sup>
<code>- * * *</code>	<code>sub_%t1%%t2%_tr% (%i1%, %i2%)</code>	O 1 2 R	Subtraktion zweier Werte
<code>-1 * * *</code>	<code>subl_%t1%%t2%_tr% (%i1%, %i2%)</code>	O 1 2 R	Subtraktion zweier Werte mit Sättigung <sup>a</sup>
<code>* * * *</code>	<code>mul_%t1%%t2%_tr% (%i1%, %i2%)</code>	O 1 2 R	Multiplikation zweier Werte
<code>*1 * * *</code>	<code>mull_%t1%%t2%_tr% (%i1%, %i2%)</code>	O 1 2 R	Multiplikation zweier Werte mit Sättigung <sup>a</sup>
<code>/ * * *</code>	<code>div_%t1%%t2%_tr% (%i1%, %i2%)</code>	O 1 2 R	Division zweier Werte
<code>/1 * * *</code>	<code>divl_%t1%%t2%_tr% (%i1%, %i2%)</code>	O 1 2 R	Division zweier Werte mit Sättigung <sup>a</sup>
<code>% * * *</code>	<code>mod_%t1%%t2%_tr% (%i1%, %i2%)</code>	O 1 2 R	Modulorechnung zweier Werte

a: Sättigung meint die Begrenzung des Ergebnisses auf die durch den maximal möglichen Wertebereich des Ergebnistyps vorgegebenen Grenzen.

<b>Funktions- schlüssel</b>	<b>Standard-Funktionsdefinition</b>	<b>Parameter</b>	<b>Bedeutung</b>
%1 * * *	modl_%t1%%t2%_%tr% (%i1%, %i2%)	O 1 2 R	Modulorechnung zweier Werte mit Sättigung <sup>a</sup>
*> * * * *	mul_r_%t1%%t2%_%tr% (%i1%, %i2%, %i3%)	O 1 2 3 R	Multiplikation zweier Werte mit anschließender Rechtsverschiebung
*>1 * * * * *	mul_rl_%t1%%t2%_%tr% (%i1%, %i2%, %i3%)	O 1 2 3 R	Multiplikation zweier Werte mit anschließender Rechtsverschiebung und Ergebnisbegrenzung (Limitierung)
/> * * * *	div_r_%t1%%t2%_%tr% (%i1%, %i2%, %i3%)	O 1 2 3 R	Division zweier Werte mit anschließender Rechtsverschiebung
/>1 * * * * *	div_rl_%t1%%t2%_%tr% (%i1%, %i2%, %i3%)	O 1 2 3 R	Division zweier Werte mit anschließender Rechtsverschiebung und Sättigung <sup>a</sup>
*/ * * * *	muldiv_%t1%%t2%%t3%_%tr% (%i1%, %i2%, %i3%)	O 1 2 3 R	Multiplikation zweier Werte mit anschließender Division
*/1 * * * * *	muldivl_%t1%%t2%%t3%_%tr% (%i1%, %i2%, %i3%)	O 1 2 3 R	Multiplikation zweier Werte mit anschließender Division und Sättigung <sup>a</sup>
getAt1 * *	CharTable1_getAt_%t1%_%t r(%ct%, %x%)	O 1 R	Kennlinienzugriff
getAt1R * *	CharTable1_getAtR_%t1%_ tr(%ct%, %x%)	O 1 R	gerundeter Kennlinienzugriff
getAt2 * * *	CharTable2_getAt_%t1%&t 2_%tr(%ct%, %x%, %y%)	O 1 2 R	Kennfeldzugriff
getAt2R * * * *	CharTable2_getAtR_%t1%&t 2_%tr(%ct%, %x%, %y%)	O 1 2 R	gerundeter Kennfeldzugriff
getAtFixed1 * *	CharTable1_getAtFixed_%t 1_%tr(%ct%, %x%)	O 1 R	Zugriff auf Festkennlinie
getAtFixed1R * * *	CharTable1_getAtFixedR_% t1%_%tr(%ct%, %x%)	O 1 R	gerundeter Zugriff auf Festkennlinie

a: Sättigung meint die Begrenzung des Ergebnisses auf die durch den maximal möglichen Wertebereich des Ergebnistyps vorgegebenen Grenzen.

<b>Funktions- schlüssel</b>	<b>Standard-Funktionsdefinition</b>	<b>Parameter</b>	<b>Bedeutung</b>
getAtFixed2 * * *	CharTable2_getAtFixed_ t1%&t2%_tr(%ct%, %x%, %y%)	0 1 2 R	Zugriff auf Festkennfeld
getAtFixed2R * * *	CharTable2_getAtFixedR_ t1%&t2%_tr(%ct%, %x%, %y%)	0 1 R	gerundeter Zugriff auf Festkennfeld
interp1Grou * * *	GroupTable1_getAt_ tr(%ct%, %x%)	0 1 R	Interpolationsroutine für Gruppenkennlinie
interp1Grou * * *	GroupTable1_getAtR_ tr(%ct%, %x%)	0 1 2 R	Interpolationsroutine (gerundet) für Gruppenkennlinie
interp2Grou * * *	GroupTable2_getAt_ 2%_tr(%ct%, %x%, %y%)	0 1 2 R	Interpolationsroutine für Gruppenkennfeld
interp2Grou * * *	GroupTable2_getAtR_ t2%_tr(%ct%, %x%, %y%)	0 1 2 R	Interpolationsroutine (gerundet) für Gruppenkennfeld
searchDis-trib * * *	Distribution_search_ _(%i1%)	0 1	Stützstellensuche
getHighPart	getHighPart__(%i1%)	0	Liefert die obersten Bits einer Größe

**Tab. 4-1** Von ASCET unterstützte Arithmetische Dienste

**Hinweis**

Die Zahl und Reihenfolge der Parameter in Tab. 4-1 sind die Standards. Grundsätzlich ist Art, Zahl und Reihenfolge der Parameter nicht begrenzt, für Spezialanwendungen können Sie weitere Parameter einfügen (s. „Einen Eintrag spezifizieren:“ auf Seite 590).

Zulässige Typen

<b>Zeichenkette</b>	<b>Typ</b>
u8	unsigned integer 8 bit
u16	unsigned integer 16 bit
u32	unsigned integer 32 bit
s8	signed integer 8 bit

Zeichenkette	Typ
s16	signed integer 16 bit
s32	signed integer 32 bit
*	alle der oben genannten Typen sind möglich

**Tab. 4-2** Zulässige Typen für Operanden und Ergebnisse

Von der gewählten Operation hängt das Vorhandensein eines Ergebnistypes und die Anzahl der Operanden ab. Eine Addition besitzt beispielsweise zwei Operanden, eine Negation nur einen, und eine Multiplikation mit Rechtsshift benötigt drei Operanden.

#### *Funktionsdeklaration*

Die Funktionsdeklaration beschreibt den Funktionsaufruf in C-Syntax, der vom Codegenerator anstelle der Standardoperation verwendet werden soll. Ist eine Funktion in C beispielsweise definiert als

```
int Add(int operand1, int operand2){...},
```

so wäre der entsprechende Funktionsaufruf

```
Add(operand1, operand2);
```

Da für die Deklaration der Funktion in der Datei `services.ini` die konkreten Zeichenketten für `operand1` und `operand2` natürlich nicht bekannt sind, werden an dieser Stelle sogenannte formale Argumente verwendet. Diese Platzhalter werden vom Codegenerator erkannt und bei der Codegenerierung durch die entsprechenden Zeichenketten ersetzt. Der obige Funktionsaufruf würde in der Datei `services.ini` also so aussehen:

```
Add(%i1%, %i2%);
```

In Tab. 4-3, Tab. 4-4 und Tab. 4-5 finden Sie eine Auflistung aller vom Codegenerator unterstützten Argumente.

Argument	Bedeutung
%i1%	Ausdruck für 1. Operand
%i2%	Ausdruck für 2. Operand
%i3%	Ausdruck für 3. Operand
%t1%	Kurztyp (z.B. u8) für 1. Operand
%t2%	Kurztyp (z.B. u8) für 2. Operand
%t3%	Kurztyp (z.B. u8) für 3. Operand
%ft1%	Langtyp (z.B. uint8) für 1. Operand
%ft2%	Langtyp (z.B. uint8) für 2. Operand

Argument	Bedeutung
<code>%ft3%</code>	Langtyp (z.B. <code>uint8</code> ) für 3. Operand
<code>%tr%</code>	Kurztyp (z.B. <code>u8</code> ) für Ergebnis
<code>%ftr%</code>	Langtyp (z.B. <code>uint8</code> ) für Ergebnis
<code>%c1%</code>	Automatisch bestimmter Type-Cast für den ersten Operanden
<code>%c2%</code>	Automatisch bestimmter Type-Cast für den zweiten Operanden
<code>%c3%</code>	Automatisch bestimmter Type-Cast für den dritten Operanden
<code>%op%</code>	Name der Operation
<code>%il%</code>	Bitlänge des Eingangs

**Tab. 4-3** Formale Argumente für die Operationen `Mul` und `Div` (mit und ohne Verschiebung und Limitierung), `MulDiv`, `Add`, `Sub` (mit und ohne Limitierung), `Mod`, `Neg`, `Abs`, `getHighPart`.

Argument	Bedeutung
<code>%ct%</code>	Ausdruck für Kennlinie/-feld
<code>%x%</code>	Ausdruck für den x-Wert, an dem interpoliert werden soll
<code>%y%</code>	Ausdruck für den y-Wert, an dem interpoliert werden soll (nur Kennfeld)
<code>%tx%</code>	Kurztyp (z.B. <code>u8</code> ) für den x-Wert
<code>%ty%</code>	Kurztyp (z.B. <code>u8</code> ) für den y-Wert
<code>%tv%</code>	Kurztyp (z.B. <code>u8</code> ) für das Ergebnis
<code>%ftx%</code>	Langtyp (z.B. <code>uint8</code> ) für den x-Wert (nur Festkennlinie/-feld)
<code>%fty%</code>	Langtyp (z.B. <code>uint8</code> ) für den y-Wert (nur Festkennfeld)
<code>%ftv%</code>	Langtyp (z.B. <code>uint8</code> ) für das Ergebnis (nur Festkennlinie/-feld)
<code>%fmst%</code>	Langtyp, ergibt sich aus der maximalen Größe ( <code>maxSize</code> ) der Tabelle (nur Festkennlinie/-feld)
<code>%xd%</code>	Ausdruck für x-Distribution (nur Gruppenkennlinie/-feld)
<code>%yd%</code>	Ausdruck für y-Distribution (nur Gruppenkennfeld)
<code>%xlen%</code>	Anzahl der x-Stützstellen
<code>%ylen%</code>	Anzahl der y-Stützstellen
<code>%xdind%</code>	Index der unteren Stützstelle (x-Richtung; nur Distribution, Gruppenkennlinie/-feld)
<code>%xdoff%</code>	Ausdruck für den Offset zwischen unterer und oberer Stützstelle (x-Richtung; nur Distribution, Gruppenkennlinie/-feld)

Argument	Bedeutung
<code>%xdis%</code>	Ausdruck für den Abstand zwischen der unteren Stützstelle und dem Wert, an dem interpoliert werden soll (x-Richtung; nur Distribution, Gruppenkennlinie/-feld)
<code>%ydis%</code>	Index der unteren Stützstelle (x-Richtung; nur Gruppenkennfeld)
<code>%yoff%</code>	Ausdruck für den Offset zwischen unterer und oberer Stützstelle (y-Richtung; nur Gruppenkennfeld)
<code>%ydis%</code>	Ausdruck für den Abstand zwischen der unteren Stützstelle und dem Wert, an dem interpoliert werden soll (y-Richtung, nur Gruppenkennfeld)

**Tab. 4-4** Formale Argumente für die Operationen `Getat1`, `Getat2`, `Getatfixed1`, `Getatfixed2`, `Interpolgroup1`, `Interpolgroup2` (mit und ohne Rundung).

Argument	Bedeutung
<code>%i1%</code>	Ausdruck für den Eingangswert einer Distribution
<code>%t1%</code>	Kurztyp für den Eingangswert einer Distribution
<code>%ft1%</code>	Langtyp für den Eingangswert einer Distribution
<code>%c1%</code>	Automatisch bestimmter Type-Cast für den Eingangswert einer Distribution
<code>%d%</code>	Distributionsobjekt
<code>%xdind%</code>	Index eines Wertes innerhalb der x-Distribution
<code>%xdooff%</code>	Offset eines Wertes innerhalb der x-Distribution
<code>%xdis%</code>	Distanz zweier Werte innerhalb der x-Distribution

**Tab. 4-5** Formale Argumente für die Operation `Searchdistrib`.

Die Vielzahl der möglichen Argumente geht auf die Syntax von möglichen C-Funktionsaufrufen zurück:

```
(%ftr%)Funktionsname(%i1%, %i2%, %i3%);
```

Zusätzlich kann der Codegenerator auch die konkreten Funktionsnamen erzeugen, sofern diese der Syntax

```
%op%_%t1%t2%t3%_%tr%
```

folgen.

Die Wahl eines Funktionsnamens ist frei, wir empfehlen Ihnen jedoch die obige Syntax zu nutzen.

### 4.14.3 Erstellen und Speichern von Arithmetischen Diensten

---

Die Definition von Arithmetischen Diensten erfolgt außerhalb von ASCET in einer Datei `services.ini`. Diese Datei wird für jedes installierte Target separat im Verzeichnis des entsprechenden Targets abgelegt. Die Datei folgt dem Windows-Standard für `.ini`-Dateien.

Um den Codegenerator von ASCET schnell auf verschiedene Anforderungen innerhalb eines Targets umstellen zu können und so eine möglichst hohe Flexibilität bei der Verwendung von selbstdefinierten arithmetischen Serviceroutinen zu ermöglichen, ist die Definition von mehreren verschiedenen Sets von Arithmetischen Diensten innerhalb einer Datei `services.ini` möglich. ASCET liest bei der Initialisierung alle Informationen aus der Datei für das aktuelle Target ein und ermöglicht dann, vor jedem Lauf des Codegenerators eines der spezifizierten Sets zu verwenden.

Die Einträge in einer Datei unterliegen folgender Syntax (in BNF):

```
SERVICE-FILE ::= [ENTRY]+
ENTRY ::= [FUNCTION] | [COMMENT] | [SET]
COMMENT ::= ";"  $\Sigma^*$ 
SET ::= "["  $\Sigma^+$  "]"
FUNCTION ::= [OPERATION] ("," | "[" [OPERAND] )+ "="  $\Sigma^*$ 
OPERATION ::= "abs" | "neg" | "+" | "-" | "*" | "/"
| "%" | "+1" | "-1" | "*1" | "/1" | ">" | ">"
| ">1" | "/>1" | "*/" | "*/1" | "getAt1"
| "getAt1R" | "getAt2" | "getAt2R" | "getAtFixed1"
| "getAtFixed1R" | "getAtFixed2" | "getAtFixed2R"
| "interpolGroup" | "interpolGroup1"
| "interpolGroup1R" | "interpolGroup2"
| "interpolGroup2R" | "searchDistrib"
| "searchDistribR" | "getHighPart"
OPERAND ::= "*" | "u8" | "u16" | "u32" | "s8" | "s16"
| "s32"
```

Weitere Einzelheiten zu den `services.ini`-Dateien und der Erstellung und Bearbeitung finden Sie in Kapitel 4.14.5 „Der Schnittstelleneditor für Arithmetische Dienste“.

### 4.14.4 Verwendung von Arithmetischen Diensten in ASCET

---

ASCET unterstützt die Verwendung von Arithmetischen Diensten

- für physikalische Spezifikationen in ESDL und im Blockdiagramm;
- in Klassen, Modulen und Zustandsautomaten;



- für Offline- und Online-Experimente;
- für Mikrocontroller-Targets, Experimentaltargets (Rapid Prototyping) und PC.

Die Verwendung von Arithmetischen Diensten ist direkt an die Codegenerierung gekoppelt. Der Codegenerator entnimmt die Informationen über die zu wählende Funktion aus den Datentypen der Elemente, die mit den entsprechenden Eingängen und Ausgängen einer Operation verbunden sind.

Hat eine Addition zwei Eingänge vom Typ „unsigned integer 8 bit“ (`uint8`) und einen Ausgang vom Typ „unsigned integer 16 bit“ (`uint16`), so sucht der Codegenerator im aktuellen Set nach dem Schlüssel `+|u8|u8|u16`. Wenn der Schlüssel vorhanden ist, verwendet der Codegenerator den dort abgelegten Funktionsaufruf, andernfalls wird je nach Art der Operation die Standardoperation verwendet oder eine Fehlermeldung ausgegeben.

### *Auswahl eines Satzes*

Die Auswahl eines Satzes von Arithmetischen Diensten geschieht im Rahmen eines Projektes in ASCET.

#### **Einen Satz Arithmetischer Dienste auswählen:**



- Öffnen Sie das Projekt, in dessen Rahmen Sie die Arithmetischen Dienste verwenden wollen.

- Klicken Sie im Projekteditor auf die Schaltfläche **Project Properties**.

Das Dialogfenster „Project Properties“ wird im Knoten „Build“ geöffnet.

- Wählen Sie im Kombikästchen „Code Generator“ den Eintrag **Implementation Experiment** oder **Object Based Controller Implementation**.

#### **Hinweis**

Die Einstellungen für die „Fixed Point Codegeneration“ sind für **Physical Experiment** und **Quantized Physical Experiment** **nicht** verfügbar.

- Öffnen Sie den Knoten **Integer Arithmetic**.  
Im Kombikästchen „Arithmetic Service set“ sind alle verfügbaren Sätze von Arithmetischen Diensten aufgelistet.

- Wählen Sie im Kombikästchen den Satz aus, den Sie verwenden wollen.

### **Hinweis**

*Wird ein neues Projekt erstellt, so wird automatisch der erste Satz aus der aktuellen Datei `services.ini` für das Projekt verwendet. Enthält die aktuelle Datei keine oder nur leere Sätze oder einen leeren Satz mit der Bezeichnung `[None]` als ersten Satz in der Datei, wird im Kombikästchen „Arithmetic Service set“ automatisch `<None>` gewählt.*

Sie können für jeden Lauf des Codegenerators einen anderen Satz (so vorhanden) verwenden. Es ist erlaubt, kein Set (`<None>`) zu wählen. In diesem Fall verwendet der Codegenerator die Standardoperationen.

ASCET speichert die Information über den zuletzt ausgewählten und verwendeten Satz für jedes einzelne Projekt. Wenn für ein Projekt ein Satz eingestellt ist, der nicht in der aktuellen Datei `services.ini` enthalten ist (z.B. wenn ein Projekt von einem anderen Computer übernommen wurde, der eine andere Datei benutzt), so wird folgende Fehlermeldung generiert:

```
The selected set of arithmetic services <...> is not
available. Please check the services.ini file.
```

Sollten innerhalb eines Satzes Funktionen nicht definiert worden sein, die der Codegenerator zur Codegenerierung benötigt, so wird – in Abhängigkeit von der jeweiligen Operation – eine Fehlermeldung ausgegeben oder die Standardoperation herangezogen und eine Warnung ausgegeben. Einzelheiten dazu finden Sie im Abschnitt „Mögliche Fehlerzustände“.

Ein spezieller Fall bei der Verwendung von Arithmetischen Diensten tritt ein, wenn Konstanten in eine Operation einbezogen sind. Eine in ASCET definierte Konstante oder ein Literal hat im Allgemeinen keinen vorgegebenen Typ. Hat der Anwender die Benutzung von Arithmetischen Diensten aktiviert, werden für diese Werte bei der Codegenerierung automatisch Typen zugewiesen. Der Codegenerator wählt hierbei den ersten, zum Wert der Konstanten passenden Typ aus der Liste:

```
sint8, uint8, sint16, uint16, sint32, uint32
```

Eine Ausnahme zu dieser Regelung tritt ein, wenn ein binärer Operator eine Konstante von unbestimmtem Typ und eine Variable mit bestimmtem Typ als Eingang besitzt.

- Ist der Wert der Konstanten positiv und der Typ der Variablen ist nicht vorzeichenbehaftet (`unsigned`), so erhält die Konstante den selben Typ wie die Variable. Ziel dieser Ausnahmeregelung ist es, in möglichst vielen Fällen eine Typgleichheit der Eingänge einer Operation zu erreichen.

- Ist der Wert der Konstanten negativ, muss für diese ein vorzeichenbehafteter Typ gewählt werden. Es gibt zwei Möglichkeiten:
  - Die Variable ist nicht vorzeichenbehaftet (`unsigned`). Hier ist keine Typgleichheit möglich.
  - Die Variable ist vorzeichenbehaftet (`signed`). Wenn der Wert der Konstanten zum Typ der Variablen passt, wird dieser Typ der Konstanten zugewiesen. Andernfalls ist keine Typgleichheit möglich.

### Mögliche Fehlerzustände

Wenn die Verwendung von Arithmetischen Diensten aktiviert ist, ist der Codegenerator auf die Richtigkeit und Vollständigkeit der Arithmetischen Dienste angewiesen.

Für alle Targets wird eine Fehlermeldung

```
Arithmetic service <name> required but not defined
```

generiert, wenn für eine Operation ein Arithmetischer Dienst benötigt wird, der im aktuell gewählten Satz nicht vorhanden ist. Dieses Verhalten ist auf alle aus reinen arithmetischen Operationen (+, -, \*, /, abs, neg) abgeleiteten Operation beschränkt.

#### Hinweis

*Im Gegensatz zu früheren ASCET Versionen werden im obigen Fall **keine** Standardoperationen verwendet!*

Um sicherzustellen, dass für alle Operationen ein entsprechender Arithmetischer Dienst definiert ist, können im entsprechenden Satz die Standardoperationen durch Einfügen von Wildcard-Funktionen eingebunden werden. Für jede aus einer arithmetischen Operation (+, -, \*, /, abs, neg) abgeleiteten Operation muss dann ein entsprechender Eintrag vorhanden sein.

Für eine Addition sähe dieser Eintrag so aus:

```
+|*|*|*|*=(%i1% + %i2%)
```

für eine kombinierte Multiplikation/Division so:

```
*|/*|*|*|*|*|*=( (%i1% * %i2%) / %i3%)
```

Die Wildcards \*) werden vom Codegenerator durch alle theoretisch möglichen Kombinationen von Typen ersetzt.

#### Hinweis

*Wenn für eine spezielle Typkombination (z.B. +|u8|s8|s8 = `add_u8s8_s8(%i1%, %i2%)`) eine Funktionsdefinition im aktuellen Satz vorhanden ist, so hat diese Vorrang vor der Standardoperation.*

ASCET generiert eine Warnung,

```
Usage of arithmetic service <name> requires extra
code for requantization
```

wenn alle direkt mit einer arithmetischen Operation (+, -, \*, /, abs, neg) verbundenen Elemente Implementation-Casts sind und der Anwender unterschiedliche Quantisierungen für diese festgelegt hat.

#### Hinweis

*Sind für die Services getAt.., searchDistrib, interpolGroup.., getHighPart und modulo im aktuellen Satz keine Funktionen definiert, verwendet der Codegenerator hier die Standardoperationen von ASCET, **ohne** eine Warnmeldung oder einen Hinweis zu generieren.*

#### 4.14.5 Der Schnittstelleneditor für Arithmetische Dienste

Der Schnittstelleneditor für Arithmetische Dienste (AS-Editor, nach engl. *Arithmetic Service Interface Editor*) ist ein zum Lieferumfang von ASCET gehörendes Werkzeug, mit dessen Hilfe Dateien, die Schnittstellendefinitionen für Arithmetische Dienste enthalten, erstellt und bearbeitet werden können. Diese Dateien tragen den Namen `services.ini` und müssen in jedem Targetverzeichnis von ASCET angelegt werden.

Die Dateien für Arithmetische Dienste (AS-Dateien) enthalten Schnittstellendefinitionen, die für die Codegenerierung mit ASCET notwendig sind. AS-Dateien entsprechen in ihrem Aufbau dem Windows-Standard für \*.ini-Dateien und enthalten ausschließlich Definitionen von Sätzen, Schnittstellendefinitionen und Kommentare.

Eine Satzdefinition ist eine Zeichenkette eingeschlossen in eckige Klammern ([ ]), eine Schnittstellendefinition ist eine beliebige, einer vorgegebenen Syntax entsprechende Zeichenkette und ein Kommentar ist eine beliebige Zeichenkette beginnend mit einem Semikolon.

Eine AS-Datei sollte mindestens einen Satz und höchstens 255 Sätze von Arithmetischen Diensten enthalten. Ein Satz beginnt mit der Deklaration seines Namens und beinhaltet dann alle Schnittstellendefinitionen bis zur nächsten Satzdefinition oder dem Dateiende.

Eine AS-Datei sieht beispielhaft so aus:

```
[Setname]
Funktion
Funktion
...
```

```
[Setname]  
;Kommentar  
Funktion  
Funktion  
...
```

Eine detaillierte Syntax für den Aufbau einer AS-Datei finden Sie im Kapitel 4.14.3.

### Funktionsumfang des AS-Editors

Der AS-Editor ist in der Lage, bestehende AS-Dateien zu öffnen, neue AS-Dateien anzulegen und vorgenommenen Änderungen in diesen Dateien zu speichern.

Darüber hinaus bietet der Editor folgende Funktionalitäten:

- Erstellen eines neuen (leeren) Satzes,
- Löschen eines Satzes,
- Duplizieren eines Satzes,
- Umbenennen eines Satzes,
- Erstellen einer neuen (leeren) Schnittstellendefinition,
- Löschen einer Schnittstellendefinition,
- Ändern einer Schnittstellendefinition in allen Bestandteilen,
- Erstellen eines Standard-Funktionsaufrufes für eine Schnittstellendefinition,
- Syntaktische Überprüfung eines Eintrages,
- Umwandeln einer Schnittstellendefinition in einen Kommentar,
- Umwandeln eines Kommentars in eine Schnittstellendefinition,
- Finden von Schnittstellendefinitionen innerhalb eines Satzes.

Alle Änderungen an einer Schnittstellendefinition werden durch Auswahl verschiedener vorgegebener Werte aus Auswahllisten vorgenommen. Veränderungen in einer AS-Datei können im AS-Editor nicht von Hand eingetragen werden.

Der AS-Editor verhindert das Erstellen syntaktisch fehlerhafter Einträge und erzeugt nur AS-Dateien, die vollständig den Anforderungen des Codegenerators von ASCET entsprechen.

## Starten des Editors

---

Der AS-Editor ist ein eigenständiges Tool, das separat oder aus ASCET heraus gestartet werden kann.

### AS-Editor aus ASCET starten:

---

- Öffnen Sie im Komponentenmanager das Untermenü **Tools** → **Arithmetic Service Editor**.

Das Untermenü enthält die auf Ihrem Rechner installierten Targets. Der Eintrag >PC< ist immer vorhanden.

- Wählen Sie das Target, dessen AS-Datei Sie bearbeiten wollen, z.B. **Tools** → **Arithmetic Service Editor** → >PC<.

Der AS-Editor öffnet sich. Die AS-Datei des gewählten Targets wird geöffnet.

Sollte im entsprechenden Targetverzeichnis die Datei `services.ini` nicht vorhanden sein, so wird automatisch eine neue Datei mit entsprechendem Namen erzeugt.

### AS-Editor aus dem Windows Explorer starten:

---

Um den Editor unabhängig von ASCET zu starten, können Sie ihn aus dem Windows Explorer heraus aufrufen.

- Öffnen Sie den Windows Explorer.
- Wechseln Sie in das Unterverzeichnis `ETAS\Ascet5.2\ASEditor` Ihrer ASCET Installation.
- Starten sie die Anwendung `editor.exe`.

Der AS-Editor öffnet sich.

### AS-Editor aus dem ASCET-Startmenü starten:

---

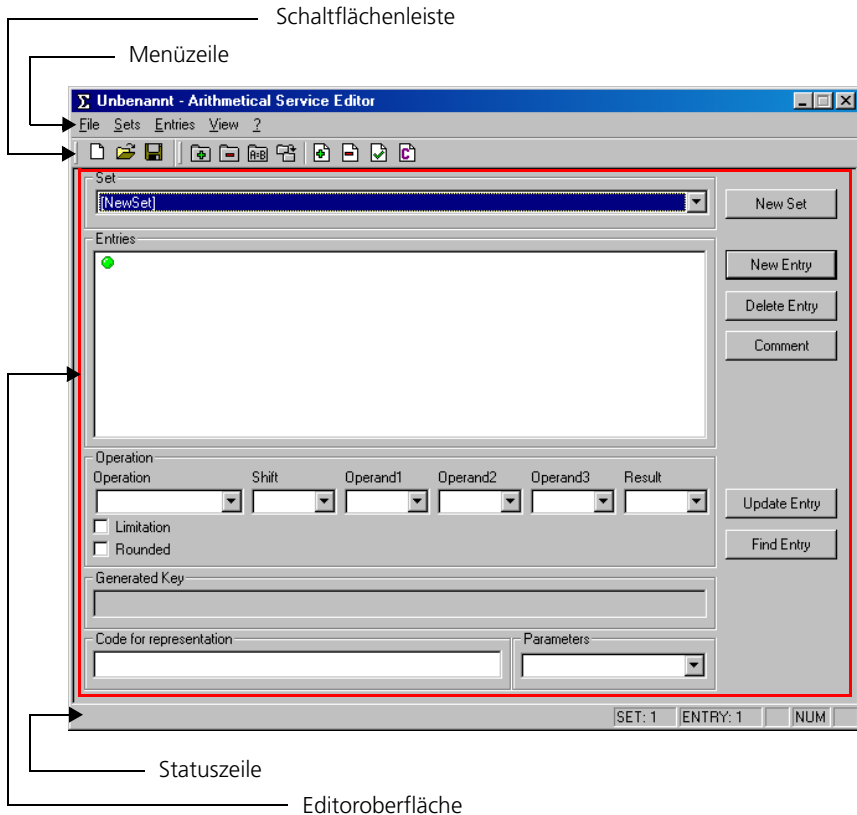
Als dritte Möglichkeit können Sie den AS-Editor auch aus dem ASCET-Startmenü heraus aufrufen.

- Klicken Sie in der Windows-Taskleiste auf die Schaltfläche **Start**.
- Wählen Sie in der Programmgruppe **ETAS** den Eintrag **ASCET5.2** → **AS-Editor**.

Der AS-Editor öffnet sich.

## Die Benutzeroberfläche des AS-Editors

Nach dem Start des AS-Editors öffnet sich das Programmfenster. Die folgende Grafik gibt einen Überblick über seine wesentlichen Bestandteile.



### Beschreibung der Menüfunktionen:

Über die Funktionen der Menüzeile können alle Dateifunktionen (**File**) aufgerufen, Sätze (**Sets**) und Einträge (**Entries**) bearbeitet, das Aussehen des Editors bestimmt (**View**) und Informationen über das Programm abgerufen (**?**) werden.

- **File**
  - *New* (<CTRL> + <N>)  
Erstellt eine neue (leere) AS-Datei.

- *Open* (<CTRL> + <O>)
  - Öffnet eine bestehende AS-Datei.
- *Save* (<CTRL> + <S>)
  - Speichert die aktuelle AS-Datei.
- *Save As*
  - Speichert die aktuelle AS-Datei unter einem beliebigen Namen.
- *1 .. 4*
  - Öffnet eine der letzten 4 geöffneten AS-Dateien
- *Exit*
  - Beendet den AS-Editor.
- **Sets**
  - *New*
    - Erstellt einen neuen (leeren) Satz.
  - *Copy*
    - Erzeugt eine Kopie des aktuellen Satzes.
  - *Rename*
    - Benennt den aktuellen Satz um.
  - *Delete*
    - Löscht den aktuellen Satz.
- **Entries**
  - *New*
    - Erstellt einen neuen (leeren) Eintrag
  - *Update*
    - Übernimmt die Änderungen für den aktuellen Eintrag.
  - *Delete*
    - Löscht den aktuellen Eintrag.
  - *Comment*
    - Wandelt den aktuellen Eintrag in einen Kommentar um.
  - *Find*
    - Sucht einen Eintrag.
- **View**



- *Symbolbar*  
Blendet die Standard-Schaltflächenleiste ein/aus.
- *Statusbar*  
Blendet die Statuszeile ein/aus
- *Toolbar*  
Blendet die spezifische Schaltflächenleiste ein/aus
- ?
  - *About Editor*  
Zeigt Informationen über das Programm an.

### Beschreibung der Schaltflächenleisten:

Der AS-Editor verfügt über zwei Schaltflächenleisten. Zum einen eine normale Standard-Schaltflächenleiste, zum anderen eine mit editor-spezifischen Befehlen.

1 2 3



1. Erstellt eine neue (leere) AS-Datei.
2. Öffnet eine bestehende AS-Datei.
3. Speichert die aktuelle AS-Datei.

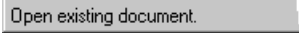
4 5 6 7 8 9 10 11



4. Erstellt einen neuen (leeren) Satz.
5. Löscht den aktuellen Satz.
6. Benennt den aktuellen Satz um.
7. Erzeugt eine Kopie des aktuellen Satzes.
8. Erstellt einen neuen (leeren) Eintrag.
9. Löscht den aktuellen Eintrag.
10. Übernimmt die Änderungen für den aktuellen Eintrag.
11. Wandelt einen Eintrag in einen Kommentar um.

### Die Statuszeile:

Die Statuszeile gibt kurze Hilfestellungen und Informationen über Funktionen von grafischen Bedienelementen und zeigt den Index des aktuellen Satzes und des aktuellen Eintrags an. Zusätzlich können die Zustände der Tasten <NUM>, <CAPSLOCK> und <ROLL> abgelesen werden.



Open existing document.



SET: 1



ENTRY: 1



NUM

### Aussehen des AS-Editors ändern:

---

Über das Menü **View** können Sie die Anzeige einzelner Elemente des AS-Editors ein- oder ausschalten.

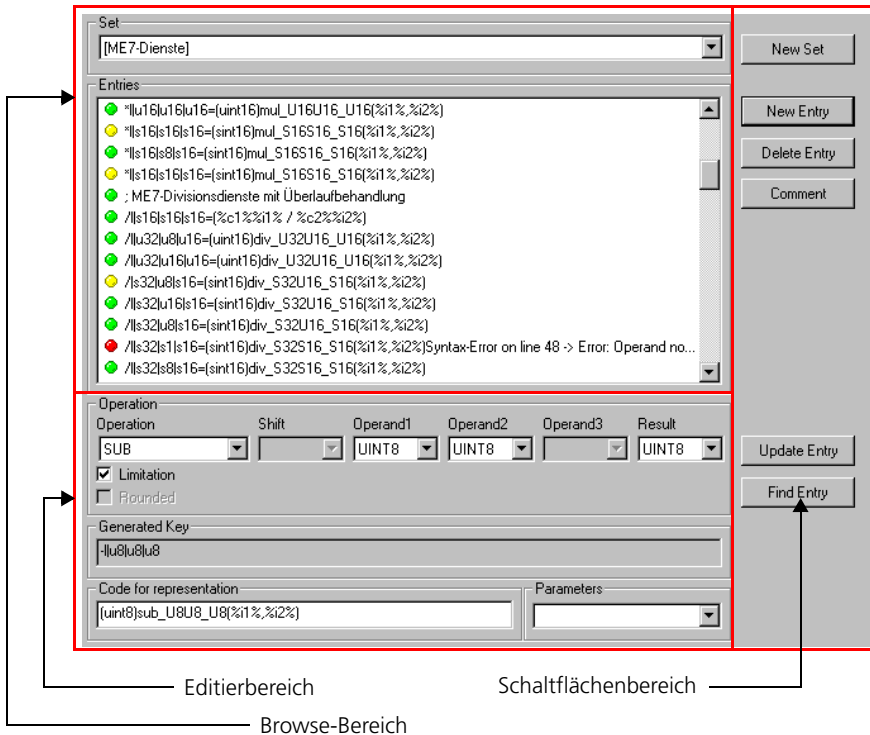
- Wählen Sie die Menüfunktion **View** → **Symbolbar**, um die Standard-Schaltflächenleiste ein- oder auszuschalten.

Wenn die Schaltfläche angezeigt wird, ist die Menüfunktion mit einem Haken markiert.

- Wählen Sie **View** → **Toolbar**, um die spezifische Schaltflächenleiste ein- oder auszuschalten.
- Wählen Sie die Menüfunktion **View** → **Statusbar**, um die Statuszeile ein- oder auszuschalten.

## Die Editoroberfläche

Die Editoroberfläche ist wiederum in drei Bereiche gegliedert.

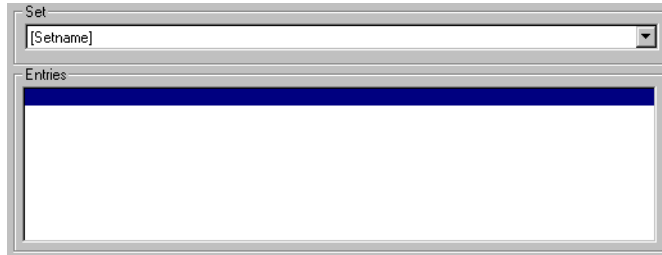


Im *Browse-Bereich* werden alle Informationen und Einträge aus einer AS-Datei dargestellt. Im *Editierbereich* lassen sich Änderungen an diesen Einträgen vornehmen und über die Schaltflächen im *Schaltflächenbereich* können Einträge erstellt, aktualisiert und gelöscht werden.

### Die Elemente des Browse-Bereichs:

Im *Browse-Bereich* werden alle Informationen und Einträge der aktuellen AS-Datei dargestellt. Wurde eine Datei erfolgreich eingelesen, werden alle in der Datei gefundenen Sätze mit ihrem Namen im Kombikästchen „Set“ eingetragen.

gen. Die zum jeweils ausgewählten Satz gehörenden Einträge werden darunter im Feld „Entries“ aufgelistet. Wird ein anderer Satz ausgewählt, ändern sich die Einträge im Feld „Entries“ entsprechend.



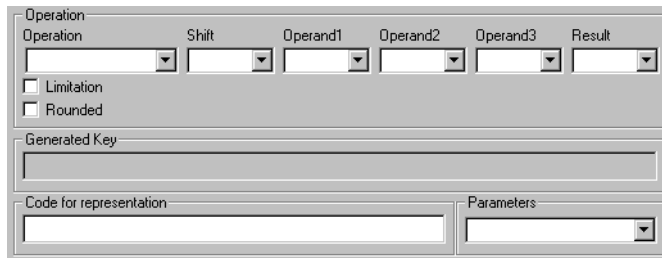
Das Feld „Set“ ist ein Kombikästchen; beim Anklicken öffnet sich eine Liste, aus der exakt ein Element per Mausklick ausgewählt werden kann. Die Einträge in dieser Liste sind nicht direkt editierbar und werden vom Programm automatisch generiert.

Das Feld „Entries“ ist ein Listenfeld. Es kann genau ein Element aus der Liste ausgewählt werden. Die Einträge sind nicht direkt editierbar und werden vom Programm automatisch eingetragen.

Das Symbol vor jedem Eintrag kennzeichnet den momentanen Zustand des Eintrages. Ein grüner Punkt bedeutet, dass der Eintrag syntaktisch richtig ist, ein roter Punkt, dass er syntaktisch falsch ist. Gelbe Punkte symbolisieren Einträge, die zwar syntaktisch richtig sind, aber gegen die Regeln einer AS-Datei verstoßen (z.B. doppelt vorhandene Einträge).

### Die Elemente des Editierbereichs:

Im Editierbereich werden die zu einem Eintrag gehörenden Informationen in aufbereiteter Form dargestellt. Gleichzeitig können hier alle einen Eintrag bildenden Werte geändert werden.



Das Feld „Operation“ ist ein Kombikästchen, aus der exakt ein Eintrag gewählt werden kann. Die Einträge in dieser Liste sind nicht editierbar und stellen alle möglichen und zulässigen Operationen dar. Wird eine Element aus dieser Liste gewählt, werden automatisch alle für diese Operation nicht benötigten Felder deaktiviert. Folgende Operationen stehen zur Verfügung:

ADD, SUB, MUL, DIV, MOD, MULDIV, NEG, ABS, GETAT,  
GETATFIXED, INTERPOLGROUP, SEARCHDISTRIB, GETHIGHPART

Die Felder „Shift“, „Operand1“, „Operand2“, „Operand3“ und „Result“ sind ebenfalls Kombikästchen. Die Einträge für alle 5 Felder sind gleich und geben alle zulässigen Typen an. Dies sind

ALL, UINT8, UINT16, UINT32, SINT8, SINT16, SINT32

Der Typ ALL steht für keine Typfestlegung und ist als Wildcard zu verstehen. Zusätzlich kann auch kein Typ gewählt werden. Dies ist jedoch nur bei solchen Operationen zulässig, die optionale Parameter zulassen.

Die Optionen **Limitation** und **Rounded** sind optionale Angaben, die für einige Operationen gesetzt werden können.

Im Feld „Generated Key“ wird der für ASCET notwendige Schlüssel (siehe Abschnitt „Funktionsschlüssel“) der Funktion dargestellt. Diese Feld ist nicht editierbar, da der Schlüssel aus Sicherheitsgründen nur vom Programm generiert werden kann.

Das Feld „Code for representation“ stellt den für die Codegenerierung in ASCET verwendeten Funktionsaufruf (siehe Abschnitt „Funktionsdeklaration“) dar. Prinzipiell ist hier jeder beliebige Eintrag zulässig, da dieser keiner Syntax unterliegt und vom Programm auch nicht auf semantische Korrektheit überprüft werden kann.

Das Feld „Parameters“ ist ein Kombikästchen, aus dem aus einer Liste ein Parameter (z.B. %i1%) ausgewählt werden kann. Der gewählte Parameter wird automatisch im Feld „Code for representation“ eingefügt.

### Die Elemente des Schaltflächenbereichs:

Dieser Bereich bietet Schaltflächen für die am häufigsten benötigten Funktionen an.



- **New Set** – Erstellt einen neuen (leeren) Satz.
- **New Entry** – Erstellt einen neuen (leeren) Eintrag.
- **Delete Entry** – Löscht den aktuellen Eintrag.
- **Comment** – Wandelt einen Eintrag in einen Kommentar um.
- **Uncomment** (ersetzt **Comment** bei ausgewähltem Kommentar) – Wandelt einen Kommentar in einen Eintrag um.
- **Update Entry** – Übernimmt die Änderungen für den aktuellen Eintrag.
- **Find Entry** - Sucht einen Eintrag.

#### Hinweis

*Abhängig vom aktuellen Status des Programms sind nicht immer alle Schaltflächen aktiviert.*

*Den AS-Editor verwenden*

---

#### Eine Datei laden:



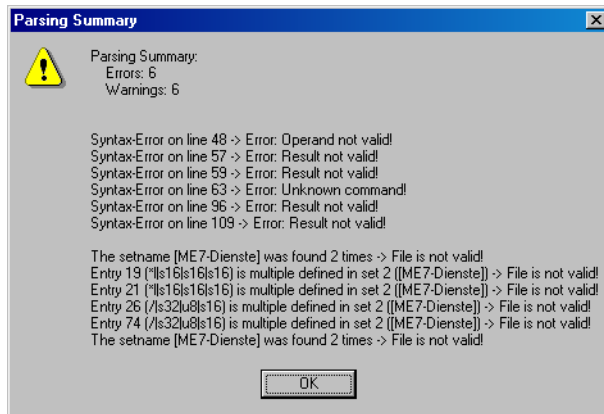
- Betätigen Sie die Schaltfläche **Open**  
*oder*
- wählen Sie die Menüfunktion **File** → **Open**.  
Das Dateiauswahlfenster öffnet sich.
- Wählen Sie die Datei aus, die Sie öffnen wollen.
- Klicken Sie auf die Schaltfläche **Öffnen**, um die Datei zu laden,  
*oder*

- klicken Sie auf **Abbrechen** um den Vorgang abzubrechen.

### Hinweis

Über das Menü **File** können die letzten vier geöffneten Dateien direkt ausgewählt werden.

Wurde eine Datei ausgewählt, wird der Inhalt dieser Datei durch das Programm eingelesen. Stellt das Programm dabei fest, dass Einträge fehlerhaft sind, werden entsprechende Meldungen generiert und nach Ende des Einlesens im Fenster „Parsing Summary“ angezeigt.



Fehlerhafte Einträge werden automatisch in Kommentare umgewandelt und mit einer Fehlerbeschreibung versehen. Sie können anhand des entsprechenden Symbols (roter bzw. gelber Punkt) leicht im Feld „Entries“ gefunden werden.

### Hinweis

Das Programm ignoriert beim Einlesen einer Datei alle Einträge die vor der ersten Satzdefinition (vor der ersten öffnenden Klammer „[“ ) stehen. Ebenso werden alle leeren Zeilen ignoriert und nicht eingelesen.

Nachdem alle Einträge eingelesen wurden, werden die des ersten gefundenen Satzes im Feld „Entries“ dargestellt.

## Eine Datei speichern:

---



- Klicken Sie in der Schaltflächenleiste auf die Schaltfläche **Save**

oder

- wählen Sie **File** → **Save**, um die aktuelle Datei zu speichern.

Wenn es sich um eine neu erstellte Datei handelt, öffnet sich der Standard-Dialog zum Speichern einer Datei.

- Wählen Sie hier einen Dateinamen und einen Speicherort.
- Bestätigen Sie ihre Angaben mit **Speichern**

oder

- brechen Sie den Vorgang mit **Abbrechen** ab.

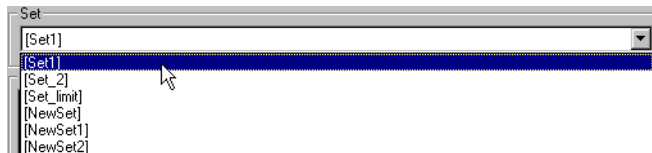
## Sätze verwalten:

Wurde eine Datei eingelesen, sind alle in dieser Datei enthaltenen Sätze über das Kombikästchen „Sets“ wählbar.

## Einen Satz auswählen:

---

- Klicken Sie in das Kombikästchen „Set“ bzw. auf die Schaltfläche am rechten Ende des Feldes.
- Wählen Sie aus der erscheinenden Liste einen Eintrag aus.



Sollte die Anzahl der Sätze die Größe der Liste überschreiten, so befindet sich auf der rechten Seite der Liste eine Bildlaufbalken.

Der ausgewählte Satz wird aktiviert und die dazugehörigen Einträge im Feld „Entries“ dargestellt.



## Einen Satz erstellen:

---

- Wählen Sie die Menüfunktion **Sets** → **New**  
oder



- klicken Sie in der Schaltflächenleiste auf die Schaltfläche **New Set**.

oder

- klicken Sie im Schaltflächenbereich auf **New Set**.

Das Dialogfenster „Please enter name of new set“ öffnet sich.

Please enter name of new set

Enter a name for the new Set. Setnames must be unique and can contain all characters except [ ].

Setname:

OK Cancel

- Geben Sie den Namen für den neuen Satz an.

### Hinweis

*Da die Sätze über den Namen unterschieden werden, dürfen zwei Sätze nicht den selben Namen erhalten. Ein Name muss mindestens 1 und darf höchstens 255 Zeichen lang sein und darf keines der beiden Zeichen [ oder ] enthalten.*

- Bestätigen Sie Ihre Eingabe mit **OK**.

Der neue Satz wird an das Ende der Liste der vorhandenen Sätze angehängt und gleichzeitig als aktiver Satz ausgewählt. Das Feld „Entries“ wird geleert, da der neue Satz noch keine Einträge enthält.

Oder

- brechen Sie den Vorgang über **Cancel** ab.

## Einen Satz löschen:

---

- Wählen Sie den Menüpunkt **Sets** → **Delete**  
oder



- klicken Sie in der Schaltflächenleiste auf die Schaltfläche **Delete Set**.  
Sie werden aufgefordert, das Löschen des Satzes mit allen zugehörigen Einträgen zu bestätigen.
- Bestätigen Sie das Löschen des Satzes mit **OK**  
*oder*
- brechen Sie den Vorgang mit **Cancel** ab.

### Hinweis

*Mit dem Satz werden auch alle Einträge des Satzes unwiderruflich gelöscht.*

### Einen Satz duplizieren:

- Wählen Sie **Sets** → **Copy**

*oder*



- klicken Sie auf die Schaltfläche **Copy Set**.  
Das Dialogfenster „Please enter name of new set“ öffnet sich.
- Geben Sie den Namen des neuen Satzes an.
- Bestätigen Sie den Vorgang mit **OK**.

Die Kopie des aktuellen Satzes wird unter dem neuen Namen am Ende der Liste angehängt und als aktiver Satz markiert.

*Oder*

- brechen Sie den Vorgang mit **Cancel** ab.

### Einen Satz umbenennen:

- Wählen Sie **Sets** → **Rename**

*oder*



- klicken Sie auf die Schaltfläche **Rename Set**.
- Geben Sie im Dialogfenster „Please enter name of new set“ den neuen Namen des Satzes an.
- Bestätigen Sie den Vorgang mit **OK**

*oder*

- brechen Sie den Vorgang mit **Cancel** ab.

**Einträge verwalten:**

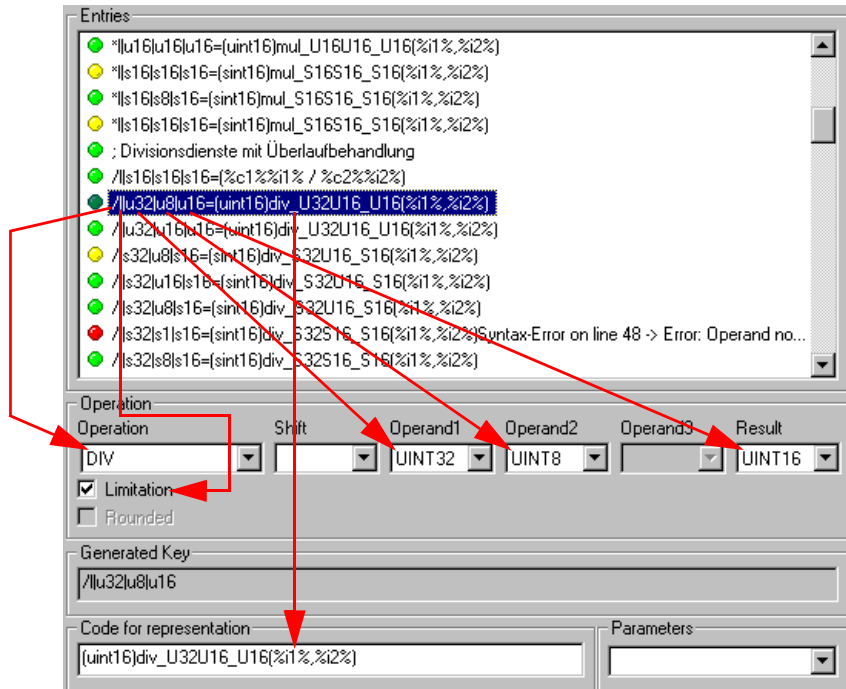
Wurde eine Datei eingelesen oder ein Satz ausgewählt, werden alle zum aktuellen Satz gehörenden Einträge im Feld „Entries“ dargestellt. Sollten mehr Einträge vorhanden sein, als im Listenfeld angezeigt werden können, so erscheint an der rechten Seite des Feldes ein Bildlaufbalken. Durch einen einfachen Mausklick auf einen der angezeigten Einträge wird dieser ausgewählt; die zugehörigen Informationen werden im *Editierbereich* dargestellt.

## Einen Eintrag auswählen:

- Klicken Sie im Feld „Entries“ auf einen Eintrag.

Die zum Eintrag gehörenden Werte werden im Editierbereich dargestellt.

Die Pfeile in der Abbildung zeigen an, welcher Teil des Eintrags in welchem Feld zu finden ist..



Ist der Eintrag fehlerhaft (gelber oder roter Punkt), wird er automatisch wie ein Kommentar behandelt.

Prinzipiell werden zwei Typen von Einträgen unterschieden: Funktionen und Kommentare. Kommentare beginnen immer mit einem „;“. Ist der ausgewählte Eintrag ein Kommentar, so sind alle Felder mit Ausnahme der Felder „Code for representation“ und „Parameters“ deaktiviert. Ist der ausgewählte Eintrag eine Funktion, so werden abhängig von der Operation die entsprechenden Felder aktiviert und die damit verbundenen Werte dargestellt.

## Einen Eintrag erstellen:

- Wählen Sie **Entry** → **New**

oder



- klicken Sie in der Schaltflächenleiste auf die Schaltfläche **New Entry**

oder

- klicken Sie im Schaltflächenbereich auf die Schaltfläche **New Entry**.

Im aktuellen Satz wird unter dem bisher ausgewählten Eintrag ein neuer, leerer Eintrag erstellt und alle Elemente des Editierbereichs werden aktiviert.

The screenshot shows the 'Entries' dialog box. The list of entries includes:

- $(s16)s16=(sint16)mul\_S16S16\_S16(\%1\%,\%2\%)$
- $(s16)s8s16=(sint16)mul\_S16S16\_S16(\%1\%,\%2\%)$
- $(s16)s16s16=(sint16)mul\_S16S16\_S16(\%1\%,\%2\%)$
- ; Divisionsdienste mit Überlaufbehandlung
- $(s16)s16s16=(\%c1\%\%1\% / \%c2\%\%2\%)$
- $(u32)u8u16=(uint16)div\_U32U16\_U16(\%1\%,\%2\%)$
- $(u32)u16u16=(uint16)div\_U32U16\_U16(\%1\%,\%2\%)$
- $(s32)u8s16=(sint16)div\_S32U16\_S16(\%1\%,\%2\%)$
- $(s32)u16s16=(sint16)div\_S32U16\_S16(\%1\%,\%2\%)$
- $(s32)u8s16=(sint16)div\_S32U16\_S16(\%1\%,\%2\%)$
- $(s32)s1s16=(sint16)div\_S32S16\_S16(\%1\%,\%2\%)$  Syntax-Error on line 48 -> Error: Operand no...

The 'Operation' section has dropdown menus for 'Operation', 'Shift', 'Operand1', 'Operand2', 'Operand3', and 'Result'. There are checkboxes for 'Limitation' and 'Rounded'. At the bottom, there are fields for 'Code for representation' and 'Parameters'.

Der neue Eintrag ist eine Funktion. Um einen Kommentar zu erstellen, wandeln Sie den eben erstellten Eintrag direkt in einen Kommentar um (siehe: „Einen Eintrag in einen Kommentar umwandeln:“ auf Seite 594).

Im AS-Editor können Änderungen an Einträgen nicht direkt durch Bearbeiten der Zeichenketten vorgenommen werden. Statt dessen können die jeweiligen Parameter eines Eintrages durch Auswahl verschiedener Vorgaben beeinflusst

werden. Um versehentliches Ändern eines Eintrages zu vermeiden, müssen alle Änderungen durch manuelles Durchführen einer Aktualisierung übernommen werden.

### **Einen Eintrag spezifizieren:**

---

- Markieren Sie in der Liste der Einträge des Feldes „Entries“ den zu ändernden Eintrag.  
Sie können alle Werte in den Kombikästchen des Editierbereichs nach Ihren Wünschen bearbeiten. Wählen Sie dazu in die entsprechenden Kombikästchen die gewünschten Einträge.
- Wählen Sie im Kombikästchen „Operation“ die gewünschte Rechenoperation.

#### **Hinweis**

*Je nach dem gewählten Eintrag stehen nicht alle Elemente im Editierbereich zur Verfügung. Sind bestimmte Einträge für die gewählte Operation nicht zulässig, werden sie automatisch deaktiviert.*

- Wählen Sie in den Kombikästchen „Shift“ den Typ der Verschiebung.  
Wenn Sie keine Verschiebung durchführen wollen, wählen Sie den leeren Eintrag.
- Wählen Sie in den Kombikästchen „Operand\*“ die Typen der Operatoren.  
Wenn Sie einen Operator für den Zugriff auf Kenngrößen (GETAT, GETATFIXED, INTERPOLGROUP) auswählen, werden die Felder „Operand1“ und „Operand2“ aktiviert. Wenn Sie auf ein *Kennfeld* zugreifen wollen, wählen Sie in beiden Feldern einen Wert aus. Wenn Sie auf eine *Kennlinie* zugreifen wollen, lassen Sie „Operand2“ leer.
- Wählen Sie im Kombikästchen „Result“ den Typ des Ergebnisses.
- Aktivieren Sie die Option **Limitation**, wenn Sie die Operation begrenzen wollen.

- Aktivieren Sie die Option **Rounded**, wenn Sie gerundeten Zugriff auf Kennlinien/-felder wünschen.
- Tragen Sie in das Feld „Code for representation“ eine Funktionsdefinition oder den Text eines Kommentars ein

oder

- löschen Sie den eventuell vorhandenen Inhalt des Feldes, um einen Standard-Funktionsdefinition zu erzeugen (siehe Seite 592).
- Wenn Sie weitere Parameter benötigen, können Sie diese nacheinander im Kombikästchen „Parameter“ auswählen.

Der ausgewählte Parameter wird im Feld „Code for representation“ am Ende der schon vorhandenen Zeichenkette eingefügt.

Wenn sie alle Eingaben vorgenommen haben, müssen Sie den Eintrag noch aktualisieren (siehe „Einen Eintrag aktualisieren:“ auf Seite 591).

#### **Hinweis**

*Der Eintrag im Feld „Generated Key“ kann nicht editiert werden. Dieser Wert wird vom Programm automatisch berechnet, sobald die Änderungen übernommen werden.*

#### **Einen Eintrag aktualisieren:**

---

- Wählen Sie **Entry → Update**

oder



- klicken Sie in der Schaltflächenleiste auf die Schaltfläche **Update Entry**

oder

- klicken Sie im Schaltflächenbereich auf die Schaltfläche **Update Entry**.

Die Werte aus dem Editierbereich werden übernommen und einer Syntax-Überprüfung unterzogen. Sind alle Angaben korrekt, werden die Änderungen auf den aktuellen Eintrag übertragen.

Wenn die Änderungen übernommen werden, führt das Programm automatisch eine Überprüfung der Werte durch. Diese umfasst folgende Punkte:

- Vollständigkeit der Angaben,
- syntaktische Richtigkeit der Angaben,
- Test auf Duplikate innerhalb des aktuellen Satzes.

Nur wenn alle Angaben vollständig und syntaktisch richtig sind und der Eintrag für diesen Funktionsschlüssel nicht schon im Satz vorhanden ist, werden die Änderungen übernommen. Andernfalls wird eine Fehlermeldung generiert und der Eintrag bleibt unverändert.

Sollte im Feld „Code for representation“ nichts eingetragen sein, so errechnet das Programm aus den anderen Angaben eine Standard-Funktionsdefinition. Diese ist als Vorschlag zu verstehen und kann jederzeit gegen eine andere Funktionsdefinition ausgetauscht werden.

#### **Hinweis**

*Wenn der Benutzer den Eintrag im Feld „Code for representation“ selbst von Hand einträgt, wird dieser keiner Überprüfung unterzogen. Der Anwender trägt in jedem Fall selbst die Verantwortung, dass die angegebene Funktionsdefinition bei der Codegenerierung mit ASCET auch tatsächlich zur Verfügung steht!*

Eine Liste der durch den Editor generierbaren Codes finden Sie im Abschnitt „Zulässige Arithmetische Dienste“. Die nachfolgenden Beispiele gibt noch einmal einen besseren Einblick in die Funktionsweise:

#### **Funktionsschlüssel: generierte Standard-Funktionsdefinition:**

+l * * *	addl_%t1%%t2%_%tr%(%i1%, %i2%)
+l * u8 *	addl_%t1%u8_%tr%(%i1%, %i2%)
+l u16 u8 *	addl_u16u8_%tr%(%i1%, %i2%)
+l u16 u8 u32	addl_u16u8_u32(%i1%, %i2%)

Die vom Editor generierten Funktionsdefinitionen können vom Anwender beeinflusst werden. Dazu müssen die beiden Dateien `operations.ini` und `type.ini`, die sich im selben Verzeichnis wie der Editor (d.h. standardmäßig `..\ETAS\Ascet5.2\ASEditor`) befinden, editiert werden. In diesen



Dateien kann zu jeder Operation und zu jedem Typ eine entsprechende Zeichenkette angegeben werden, die dann vom Editor anstelle der Standardzeichenketten verwendet wird.

### Hinweis

Achten Sie beim Bearbeiten von `operations.ini` und `type.ini` darauf, dass die Namen nicht geändert werden, da der AS-Editor die Dateien sonst nicht findet und auf die Standardeinstellungen zurückgreift.

**Beispiel:** In der Datei `operations.ini` wurde der Eintrag

```
add = add_
```

gegen

```
add = doAddition_
```

ausgetauscht. Der Code für den Schlüssel

```
+|*|*|*
```

würde nun nicht mehr

```
add_%t1%%t2%_tr%(%i1%, %i2%)
```

sondern

```
doAddition_%t1%%t2%_tr%(%i1%, %i2%)
```

sein.

### Einen Eintrag löschen:

### Hinweis

Das Löschen eines Eintrages erfolgt **ohne** Rückfrage. Ein einmal gelöschter Eintrag kann nicht wieder hergestellt werden. Verwenden Sie diese Funktion daher vorsichtig.

- Wählen Sie den Menüpunkt **Entries** → **Delete**

oder



- klicken Sie auf die Schaltfläche **Delete Entry**.  
Der markierte Eintrag wird ohne weitere Rückfrage sofort aus dem aktuellen Satz gelöscht.

Nach dem Löschen eines Eintrages wird automatisch der nächste Eintrag der Liste markiert. War der gelöschte Eintrag der letzte in der Liste, wird der vorhergehende Eintrag markiert. Dieses Verhalten ermöglicht das schnelle Löschen mehrerer aufeinanderfolgender Einträge.

Der AS-Editor bietet die Möglichkeit, Kommentare in eine AS-Datei einzufügen, um kurze Beschreibungen oder Erklärungen zu den einzelnen Einträgen zu ermöglichen. Diese Funktionalität kann aber auch dazu verwendet werden, bestehende Einträge in einen Kommentar umzuwandeln. Dabei wird dem eigentlichen Eintrag nur ein „;“ vorangestellt, worauf er sowohl von ASCET als auch vom AS-Editor als Kommentar behandelt wird.

#### **Einen Eintrag in einen Kommentar umwandeln:**

---

- Wählen Sie aus der Liste der Einträge im Feld „Entries“ einen Eintrag.

- Wählen Sie **Entries** → **Comment**

oder



- klicken Sie in der Schaltflächenleiste auf die Schaltfläche **Make Comment**

oder

- klicken Sie im Schaltflächenbereich auf die Schaltfläche **Comment**.

Der markierte Eintrag wird in einen Kommentar umgewandelt.

Kommentare können unter bestimmten Umständen in Einträge umgewandelt werden. Dies ist immer dann möglich, wenn der entsprechende Kommentar ohne das ";" einen gültigen Eintrag darstellen würde. Ebenso können auf diese Weise fehlerhafte Einträge korrigiert werden.

Soll ein Kommentar in einen Eintrag umgewandelt werden, so wird der im Feld "Code for representation" enthaltene Text erneut vom Parser untersucht. Wird dabei kein Fehler festgestellt, wird der Kommentar zu einem Eintrag umgewandelt. Sollte der Text im Feld keinem gültigen Eintrag entsprechen, so bleibt der Kommentar in seinem momentanen Zustand (ein fehlerhafter Eintrag bleibt also ein fehlerhafter Eintrag).

#### **Einen Kommentar in einen Eintrag umwandeln:**

---

- Wählen Sie im Feld „Entries“ einen Kommentar.

Die Schaltfläche **Comment** im Schaltflächenbereich heißt nun **Uncomment**.

- Wählen Sie **Entries** → **Comment**

oder

- klicken Sie im Schaltflächenbereich auf **Uncomment**.

Der markierte Kommentar wird in einen Eintrag umgewandelt, *wenn er syntaktisch korrekt ist*. Andernfalls geschieht nichts.

### **Einen Eintrag suchen:**

---

Manchmal ist es notwendig, einen speziellen Eintrag innerhalb eines Sets zu finden. Dazu ist es möglich, einen Eintrag anhand seines eindeutigen Funktionsschlüssels zu suchen.

- Wählen Sie im Editierbereich in den Feldern „Operation“, „Operand1“, „Operand2“, „Operand3“, „Shift“ und „Result“ die Werte für die zu suchende Funktion.
- Markieren Sie die Optionen **Limitation** und **Rounded** wie in der zu suchenden Funktion.

- Wählen Sie **Entries** → **Find**

*oder*

- klicken Sie im Schaltflächenbereich auf die Schaltfläche **Find Entry**.

Sofern ein Eintrag zu den eingestellten Werten passt, wird dieser automatisch ausgewählt und im Feld „Entries“ angezeigt.

Sollte die gesuchte Funktion nicht existieren oder bereits markiert sein, wird eine entsprechende Meldung ausgegeben.



## 5 Signale und Symbole

---

Dieser Abschnitt beschreibt, wie mit Signalen und Symbolen in der ASCET Datenbank gearbeitet wird. Signale und Symbole sind ergänzende Daten für ein Steuersystem, die zusammen mit dem System selbst in der ASCET Datenbank gespeichert werden können.

Sowohl Signale als auch Symbole können in die Datenbank importiert und in das Dateisystem exportiert werden. Signale werden vom System verwendet; Symbole können Elementen und Hierarchien zugeordnet werden. Die folgenden Abschnitte erläutern, wie mit jedem Typ gearbeitet wird.

### 5.1 Der Signalbetrachter

---

Mit dem Signalbetrachter können Signale in die ASCET Datenbank geladen und angesehen werden. Der Betrachter kann eine Vielzahl von Signalformaten lesen und Signale zwischen Formaten konvertieren.

Die folgenden Messdatenformate werden von ASCET unterstützt:

- Tabulatorbegrenzter ASCII. Nahezu jede Tabelle oder Datenbank kann dieses Format lesen und schreiben.
- ETAS Format. Dies ist ein sehr wirksames Binärformat und wird nur intern von ASCET genutzt.
- FAMOS-Channel-Format.
- FAMOS-Record-Format.
- Matlab-Format. Dies ist das ASCII-Format von MATLAB/Simulink.
- MDF-Format.

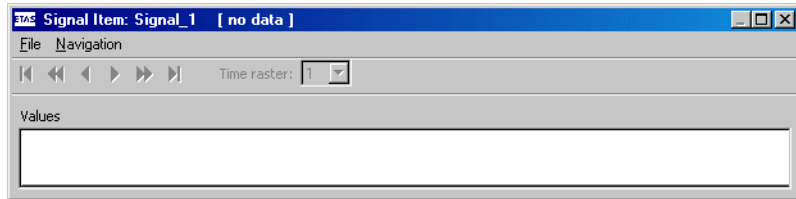
Die letzten vier sind Formate Dritter für Messdaten. Details finden Sie in der betreffenden Dokumentation. Intern kann ASCET die ASCII-, ETAS- und beide FAMOS-Formate verarbeiten. MDF und Matlab können gelesen und geschrieben werden.

#### **Erzeugen und Öffnen eines Signals:**

---

- Wählen Sie im Komponentenmanager **Insert** → **Signal**.  
Ein neues Signal wird erzeugt.
- Doppelklicken Sie auf das Signal  
*oder*
- wählen Sie **Component** → **Edit Item**  
*oder*

- drücken Sie <RETURN>.  
Der Signalbetrachter öffnet sich.



### Importieren von Messdaten in ein Signal:

---

- Öffnen Sie das Signal im Signalbetrachter.
- Wählen Sie im Signalbetrachter **File** → **File In Data** → **<format>**, um die Signaldaten aus dem Dateisystem zu importieren.  
Das Dateiauswahlfenster von Windows öffnet sich.

- Wählen Sie Pfad- und Dateinamen der zu importierenden Datei.

### Hinweis

An dieser Stelle kann auf Messdaten zurückgegriffen werden, die zuvor im Experiment erzeugt und abgespeichert wurden.

Die Daten werden importiert und automatisch in das angegebene Format umgewandelt.  
Nach der Konvertierung werden die Daten im Betrachter angezeigt.

The screenshot shows a window titled 'View Signal Item: Signal [ 132 samples - MDF Format ]'. It features navigation buttons and a 'Time raster' dropdown menu currently set to '1'. Below the menu, it says 'Displaying sample 1 ... 63'. A table displays signal data for various channels.

time	1.00500000	1.01500000	1.02500000	1.03500000	1.04510000	1.05500000
ELM1_n_CANE-target:1	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
ELM1_trq_CANE-target:1	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
ELM2_n_CANE-target:1	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
ELM2_trq_CANE-target:1	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
NMD TLE-target:1	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
DHLE-target:1	0.50000000	1.50000000	1.50000000	0.00000000	0.00000000	0.50000000

- Wenn Sie beim Import das MDF-Format oder das FAMOS Record Format ausgewählt haben können Sie verschiedene Zeitraster zur Ansicht auswählen.

### Ansehen von Messdaten:

- Wählen Sie **Navigation** → **Show Next** oder



- klicken Sie auf die Schaltfläche **Show Next Column**.

Sie gelangen zur nächsten Datenspalte.

- Wählen Sie **Navigation** → **Fast Forward** oder



- klicken Sie auf die Schaltfläche **Fast Forward**.

- Die nächste Bildschirmseite mit Daten wird angezeigt.  
Wenn z.B. der Bildschirm fünf Datenspalten anzeigt, werden die nächsten fünf gezeigt.

- Wählen Sie **Navigation** → **Show Last**

oder



- klicken Sie auf die Schaltfläche **Show Last Column**.

Sie gelangen zur letzten Datenspalte.

Zu jedem dieser drei Befehle gibt es analog einen Befehl und eine Schaltfläche zum Bewegen in die entgegengesetzte Richtung.



## 5.2 Der Symboleditor

Symbole können dem Layout von Komponenten zugewiesen werden. Das Symbol wird in dem grafischen Block gezeigt, wenn die Komponente in anderen Komponenten im Blockdiagrammeditor eingeschlossen ist. ASCET liefert einige Symbole, Sie können aber auch eigene entwerfen.

### **Erzeugen und Öffnen eines Symbols:**

- Legen Sie im Komponentenmanager mit **Insert** → **Icon** ein neues Symbol an.

Oder

- Wählen Sie ein bestehendes Symbol aus der Datenbank aus.

- Doppelklicken Sie auf das gewählte Symbol

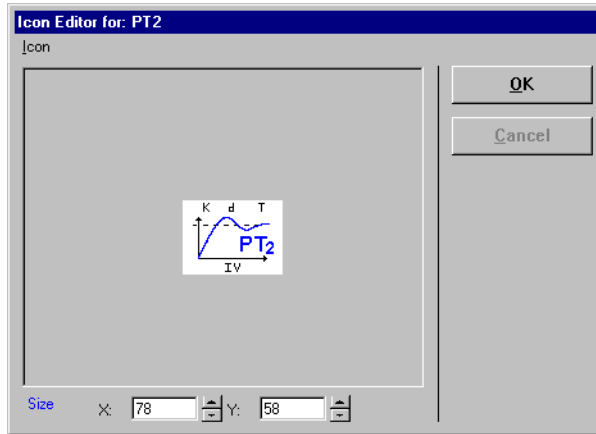
oder

- wählen Sie **Component** → **Edit Item**

oder



- drücken Sie <RETURN>.  
Der Symboleditor öffnet sich.



Haben Sie ein neues Symbol angelegt, ist die Zeichenfläche leer. Wenn Sie ein bestehendes Symbol öffnen, wird es in der Zeichenfläche angezeigt.

Sie können jede Bitmap-Datei laden (BMP, PCX und TIFF) und als Symbol in ASCET verwenden. Mit einem Zeichenprogramm können Sie eigene Symbole entwerfen, oder Sie verwenden die existierenden. Die Bitmap-Dateien, die Sie importieren, sollten von geeigneter Größe sein, d.h. nicht größer als der grafische Block, dem das Symbol zugewiesen wird. Eine Rastereinheit in ASCET beträgt 5 Pixel.

#### **Laden eines Symbols:**

- Wählen Sie im Symboleditor **Icon** → **Load**.  
Das Dialogfenster „Image file“ öffnet sich.
- Wählen Sie den Namen des zu ladenden Symboldatei.
- Klicken Sie auf **Öffnen**.  
Das ausgewählte Symbol wird im Symboleditor angezeigt.

Symbole können skaliert werden, um sie dem Block, in dem sie verwendet werden anzupassen. Eine Skalierung wird jedoch die Bildqualität herabsetzen. Es sollte daher vermieden werden mit einem großen Faktor zu skalieren.

### **Skalieren eines Symbols:**

---

- Öffnen Sie den Symboleditor für das zu skalierende Symbol.
- Passen Sie die vertikale und die horizontale Symbolgröße mittels der „Size“-Kästchen unter der Symbolfläche an.  
Die Symbolgröße ändert sich sofort, wenn ein Wert in einem Size-Kästchen geändert wird.
- Klicken Sie auf **OK**.

ASCET Symbole können als Bitmap-Dateien gespeichert werden. Dies ist zum Beispiel nützlich, wenn Sie die Dokumentation Ihrer Komponenten illustrieren wollen.

### **Speichern eines Symbols in eine Datei:**

---

- Rufen Sie den Symboleditor für das zu speichernden Symbol auf.
- Wählen Sie **Icon** → **File Out** → **<file format>**.  
Es erscheint das Dialogfenster „Image file“.
- Wählen Sie den Dateinamen und Pfad, unter dem Sie das Symbol speichern wollen.
- Klicken Sie auf **OK**.  
Das Symbol wird im entsprechenden Format gespeichert (BMP, PCX oder TIFF).

## 6 Experimentieren

### 6.1 Die Experimentierumgebung

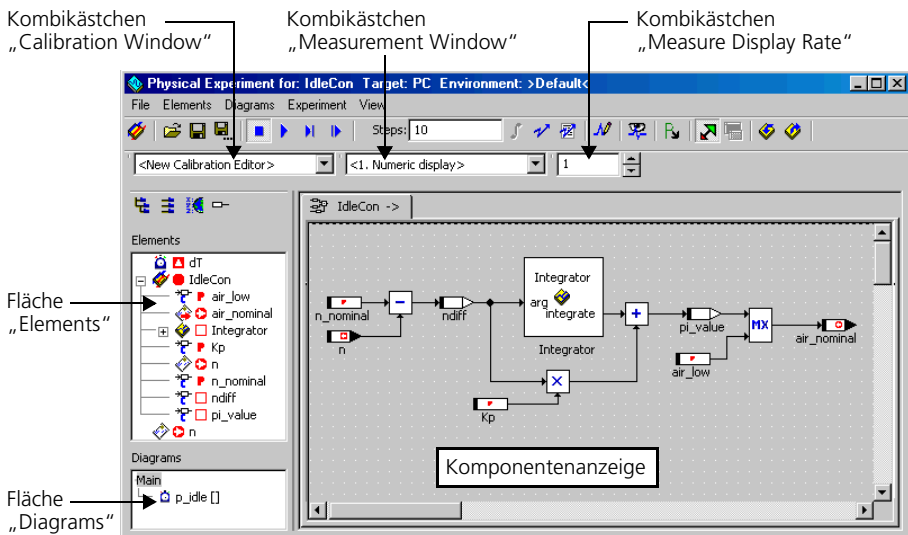
ASCET unterstützt eine hochgradig modulare Verfahrensweise für die Entwicklung von Software. Die einzelnen Komponenten eines Projekts lassen sich unabhängig voneinander entwickeln und nach gründlicher Erprobung auch miteinander kombinieren. ASCET stellt die Experimentierumgebung für das Erproben von Komponenten und Projekten bereit.

Projekte können sowohl in Offline- als auch in Online-Experimenten erprobt werden. Bei Verbindung des Projekts mit einer externen Hardware lässt sich die betreffende Verbindung nur in einem Online-Experiment erproben.

#### Hinweis

*Online-Experimente sind nur möglich, wenn Sie ASCET-RP installiert haben. Daher wird Online-Experimentieren ausführlich im Benutzerhandbuch von ASCET-RP behandelt.*

Mit der Experimentierumgebung ist es möglich, die Prozesse und Methoden sowie die Elemente einer Komponente oder eines Projekts zu stimulieren (im Offline-Experiment). Die Datenwerte der Elemente können in verschiedenen Visualisierungen gemessen, in eine Datei geschrieben und interaktiv verstellt werden. Die Hauptelemente des Experimentierfensters sind in der folgenden Darstellung zu sehen (dargestellt ist ein Offline-Experiment).



Die Elemente der Komponente sind in der Fläche „Elements“ links von der Komponentenanzeige aufgelistet. Die Komponentendarstellung zeigt ein Blockdiagramm, wenn die Komponente als Blockdiagramm spezifiziert wurde. Bei in C-Code oder ESDL spezifizierten Komponenten wird der Code angezeigt. Bei einem Projekt werden die Register des Projekteditors angezeigt.

Unterhalb der Fläche „Elements“ befindet sich die Fläche „Diagrams“, in der alle Diagramme der aktuellen Komponente aufgelistet werden. Das Kombikästchen „Measurement Window“ befindet sich unter der Schaltflächenleiste. Darin werden alle verfügbaren Messfenster aufgelistet. Links vom Kombikästchen „Measurement Window“ befindet sich das Kombikästchen „Calibration Window“, in dem alle verfügbaren Verstellfenster aufgelistet werden. In der Titelleiste sind der Name der aktuellen Komponente oder des aktuellen Projekts sowie das aktuelle Target angegeben.

### 6.1.1 Beschreibung der Menüfunktionen

---

- **File:**
  - *Load Environment*  
Laden einer Umgebung für das Experiment.
  - *Save Environment*  
Speichern der aktuellen Umgebung für das Experiment.
  - *Save Environment As*  
Speichern der aktuellen Umgebung für das Experiment unter beliebigem Namen.
  - *Export Environment*  
Exportieren der Umgebungen der aktuellen Komponente.
  - *Exit*  
Verlassen der Offline-Experimentierumgebung.
- **Elements:**
  - *Show Component*  
Zeigt die ausgewählte referenzierte Komponente an.
  - *Calibrate* (<CTRL> + <C>)  
Weist die ausgewählten Elemente einem Verstellfenster zu.

- *Stimulate* (<CTRL> + <S>)

### **Hinweis**

*Diese Menüfunktion ist nur im Offline-Experiment vorhanden.*

Öffnet das Fenster „Stimulus“ für das ausgewählte Element.

- *Measure* (<CTRL> + <M>)

Weist die ausgewählten Elemente einem Messfenster zu.

- *Log* (<CTRL> + <L>)

Fügt das ausgewählte Element in den Data-Logger ein.

- *Log All* (<CTRL> + <A>)

Fügt alle Elemente in den Data-Logger ein.

- *Reinitialize*

Liest die Werte bestimmter Elemente aus dem aktuellen Datensatz der Komponente;

→ *Variables* – von Variablen,

→ *Parameters* – von Parametern,

→ *Both* – von Variablen und Parametern.

- *Write Back Data*

Schreibt die Werte bestimmter Elemente in den Datensatz der Komponente;

→ *Selected Elements* – ausgewählte Elemente,

→ *Calibrated Elements* – im Experiment verstellte Elemente.

- *Show Element Implementation* (<CTRL> + <I>)

Zeigt die Implementierung des ausgewählten Elements an (nur Lesezugriff).

- *Update Dependent Parameters* (<CTRL> + <U>)

Aktualisiert die Werte abhängiger Parameter.

- *Load Data*

Lädt Daten aus einer Datei. Die Optionen für diesen Vorgang werden in einem eigenen Fenster eingestellt.

- *Save Data*

Speichert die Daten ausgewählter Parameter oder Variablen in eine Datei. Die Optionen für diesen Vorgang werden in einem eigenen Fenster eingestellt.

- **Diagrams:**

- *Show Diagram*  
Zeigt das ausgewählte Diagramm.
- *Stimulate*

**Hinweis**

*Diese Menüfunktion ist nur im Offline-Experiment vorhanden.*

Öffnet das Fenster „Event“ für das ausgewählte Ereignis (Prozess, Methode, Task). Das Ereignis kann eingerichtet werden.

- **Experiment:**

**Hinweis**

*Diese Menüfunktionen sind im **Offline**-Experiment verfügbar.*

- *Event Generator → Open*  
Öffnet den Ereignisgenerator.
- *Event Generator → Step Mode*  
Wählt die Art des Schrittmodus:
  - *Steps* – Einzelschrittmodus (Standardeinstellung),
  - *Timed [s]* – getakteter Zeitmodus,
  - *Break At Condition* – Schrittmodus mit Haltepunkt.
- *Event Generator → Edit Break Condition*  
Öffnet den Editor für die Bedingung des Haltepunkts.
- *Event Tracer*  
Öffnet das Fenster „Event Tracer“ zur Erstellung von Datenprotokollen.
- *Data Generator*  
Öffnet den Datengenerator.
- *Data Logger*  
Öffnet den Data-Logger.
- *Automatic Stop*  
Beendet das Experiment automatisch, wenn das Ende des stimulierenden Signals erreicht ist (s. Seite 624).
- *Stop Experiment*  
Beendet das Experiment.

- *Start Experiment*  
Startet das Experiment.
- *Pause Experiment*  
Unterbricht das Experiment.
- *Step Experiment*  
Das Experiment wird im Schrittverfahren abgearbeitet.
- *Open Target Debugger*  
Öffnet das Debugger-Fenster für C-Code-Komponenten.
- *Update Calibration Windows*  
Aktualisiert die Inhalte der Verstellfenster.
- *Close Calibration Windows*  
Schließt alle geöffneten Verstellfenster.
- *Close Measure Windows*  
Schließt alle geöffneten Messfenster.
- **View:**
  - *Redraw*  
Zeichnet das Blockdiagramm neu.
  - *Hide Seq. Calls*  
Verbirgt alle Sequenzaufrufe des aktuellen Blockdiagramms.
  - *Show Seq. Calls*  
Zeigt alle Sequenzaufrufe des aktuellen Blockdiagramms.
  - *Monitor All*  
Weist allen Elementen Monitore zu.
  - *Delete Monitors*  
Löscht alle Monitore im Diagramm.
  - *Show Parent Component*  
Zeigt die übergeordnete Komponente an.

### 6.1.2 Aufrufen und Einrichten der Experimentierumgebung

---

Sie können die Experimentierumgebung für eine Komponente oder ein Projekt aus dem entsprechenden Komponenteneditor oder Projekteditor aufrufen.

## Experimentierumgebung für Offline-Experiment aufrufen:

---

- Öffnen Sie die gewünschte Komponente oder das Projekt.
- Wenn Sie mit einem Projekt offline experimentieren wollen, wählen Sie in den Build-Optionen das Target PC.

Im Kombikästchen „Experiment Target“ ist nur der Eintrag `Offline (PC)` verfügbar. Für Komponenten ist das Kombikästchen deaktiviert.

- Wählen Sie **Component** → **Open Experiment**

oder



- klicken Sie auf die Schaltfläche **Open Experiment for selected Experiment Target**.

Die Standard-Experimentierumgebung für die Komponente/das Projekt wird geöffnet. Wurden mehrere Umgebungen gespeichert, können Sie auswählen, welche geöffnet wird.

Details dazu siehe Abschnitt „Laden und Speichern von Umgebungen“ auf Seite 638.

Wenn Sie mit Komponenten experimentieren, werden unter manchen Umständen die globalen Elemente (s. „Definieren der globalen Kommunikation“ auf Seite 416) im Standardprojekt nicht korrekt aktualisiert. In diesem Fall wird im ASCET Monitorfenster folgende Fehlermeldung ausgegeben:

```
Error: need export for imported element <name> with  
type <type>
```

Um den Fehler zu korrigieren, gehen Sie wie folgt vor:

### Definieren von globalen Elementen im Standardprojekt:

---

- Wählen Sie im Editor der Komponente **Component** → **Default Project** → **Resolve Globals**, um die globalen Elemente aufzulösen.

Sie können jetzt das Experiment erneut starten.



Nun können Sie das Experiment einrichten. Das Einrichten eines Offline-Experiments umfasst vier Schritte:

- Starten der Experimentierumgebung aus oder dem jeweiligen Komponenteneditor. Die Experimentierumgebung funktioniert für alle Typen von Komponenten gleich.
- Einrichten des Ereignisgenerators. Der Ereignisgenerator bestimmt, welche Methoden oder Prozesse in welchem Modus ausgelöst werden. Beim Offline-Experimentieren mit einem Projekt werden die Tasks stimuliert, keine Methoden oder Prozesse.
- Einrichten des Datengenerators. Hier werden einige Elemente der Komponente mit einem konfigurierbaren Datenfluss stimuliert.
- Einrichten der Mess- und Verstellfenster. Die Werte aller Elemente können in vielfältiger Form dargestellt werden, z.B. numerisch oder in einem Oszilloskop.

### 6.1.3 Der Ereignisgenerator

In einem Online-Experiment erfolgt das Scheduling für die verschiedenen Tasks und Prozesse eines Projekts durch das Echtzeit-Betriebssystem. Bei Offline-Experimenten wird die Ablaufplanung durch den Ereignisgenerator simuliert. Der Ereignisgenerator legt fest, welche Methoden oder Prozesse der Komponente im Rahmen des Experimentierens in welcher Reihenfolge und in welchem Modus aktiviert werden. Für jede(n) zu aktivierende(n) Methode/Prozess ist ein Ereignis festzulegen.

#### **Einrichten des Ereignisgenerators:**

---

- Wählen Sie **Experiment** → **Event Generator** → **Open**

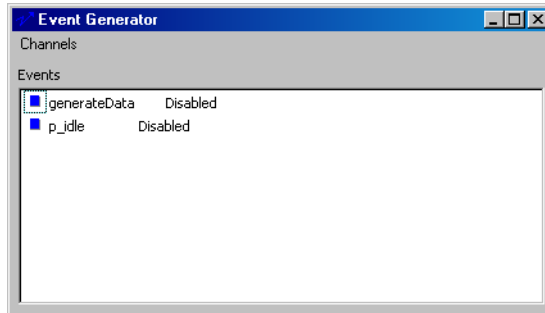
*oder*



- klicken Sie auf die Schaltfläche **Open Event Generator**.

Es wird das Fenster „Event Generator“ geöffnet. Es enthält für jede in der Komponente definierte Methode (bei Klassen) oder darin definierten Prozess (bei Modulen) ein Ereignis.

Standardmäßig sind alle Ereignisse deaktiviert, so dass Sie diejenigen aktivieren müssen, die Sie im Experiment verwenden wollen.



- Wählen Sie ein Ereignis aus.
- Wählen Sie **Channels** → **Enable**  
*oder*
- klicken Sie mit der rechten Maustaste auf das Ereignis und wählen Sie **Enable** aus dem Kontextmenü.
- Wiederholen Sie dies für jedes zu aktivierende Ereignis.
- Wählen Sie **Channels** → **Enable** erneut, um ein Ereignis wieder zu deaktivieren.

Methoden oder Prozesse, für die kein Ereignis aktiv gesetzt wurde, werden nicht aktiviert und haben somit keinen Einfluss auf das Experiment.

Neben den Ereignissen für Methoden und Prozesse wird standardmäßig stets ein Ereignis `generateData` erstellt. Dieses Ereignis löst die Generierung von im Datengenerator festgelegten Daten aus. Erfordert Ihr Experiment keine Generierung von Daten, können Sie dieses Ereignis inaktiv lassen, andernfalls muss es stets aktiv gesetzt werden.

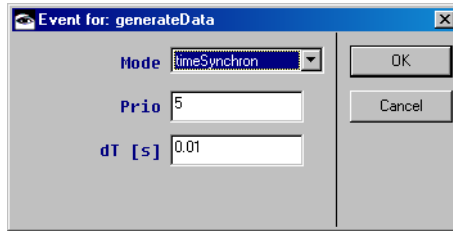
#### **Einrichten eines Ereignisses:**

---

Wenn Sie ein Ereignis erstellen, werden diesem für alle Ereignisoptionen standardmäßige Werte zugewiesen. Eine Bearbeitung dieser Optionen ist u.U. nicht immer erforderlich.

- Wählen Sie das einzurichtende Ereignis aus.

- Wählen Sie **Channels** → **Edit**.  
Es erscheint das Dialogfenster „Event“.



- Stellen Sie die Ereignisoptionen ein.  
Die Bedeutung der verschiedenen Optionen wird auf Seite 612 erläutert.
- Klicken Sie auf **OK**.
- Wiederholen Sie dies für jedes einzurichtende Ereignis.

### Direktes Einrichten eines Ereignisses:

---

- Wählen Sie in der Fläche „Diagrams“ den Prozess oder die Methode aus, für den/die Sie ein Ereignis erzeugen möchten.
- Klicken Sie mit der rechten Maustaste auf den Prozess oder die Methode und wählen Sie **Stimulate** aus dem Kontextmenü.

*Oder*

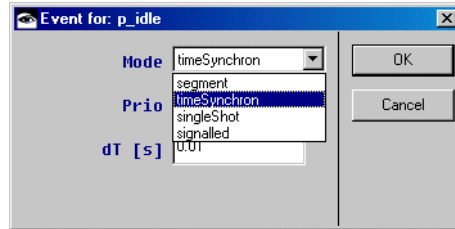
- Wählen Sie **Diagrams** → **Stimulate**.  
Das Ereignis wird aktiv gesetzt und das Dialogfenster „Event“ für das Ereignis geöffnet. Die Verwendung dieses Befehls gleicht einem Vorgang, in dem das Ereignis zunächst im Fenster „Event Generator“ aktiv gesetzt und dann bearbeitet wird.

## Ereignisoptionen

---

Das Dialogfenster „Event“ bietet für jedes Ereignis mehrere Möglichkeiten an.

- Jedes Ereignis verfügt über einen *Modus*. Es gibt vier Modi:



- Ein *zeitsynchrones Ereignis* (*timeSynchron*) wird zu Beginn jedes Intervalls ausgelöst. Die Länge eines Intervalls richtet sich nach dem im Feld „dT [s]“ gesetzten Wert.
- Ein *Segmentereignis* (*segment*) kommt bei Anwendungen im Automobilbau zum Einsatz, wo das Auslösen des Ereignisses von der Drehzahl des Motors abhängig ist; Details dazu siehe Seite 612.
- Ein *Einzelereignis* (*singleShot*) wird nur einmal zu Beginn der Simulation ausgelöst. Dies ist beispielsweise für Initialisierungsmethoden von Nutzen. Sie können das *Einzelereignis* erneut auslösen, indem Sie im Fenster „Event Generator“ **Channels → Reactivate Event** wählen.
- Ein *asynchrones Ereignis* (*signalled*) wird mit den Daten einer tatsächlichen Messung stimuliert. Auf diese Weise können auch im Offline-Experiment echte Daten als Trigger dienen. Das Einrichten eines solchen Ereignisses ist auf Seite 614 beschrieben.
- Die *Priorität* eines Ereignisses bestimmt die Reihenfolge, in der Ereignisse berechnet werden. Ereignisse werden häufig dem gleichen Zeitrahmen zugewiesen, z.B. 10 Millisekunden. In diesem Fall wird das Ereignis mit der höchsten Priorität (d.h. mit der höchsten Nummer) zuerst ausgelöst und die anderen danach.
- Der Wert *dT* eines Ereignisses bestimmt das Intervall, in dem es ausgelöst wird. Beträgt der dT-Wert 0,01 Sekunden, wird das Ereignis aller 10 Millisekunden ausgelöst. Der kleinstmögliche dT-Wert beträgt eine Mikrosekunde ( $10^{-6}$  Sekunden).

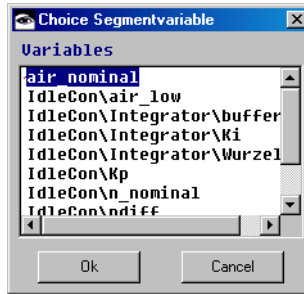
### Einrichten eines Segmentereignisses:

---

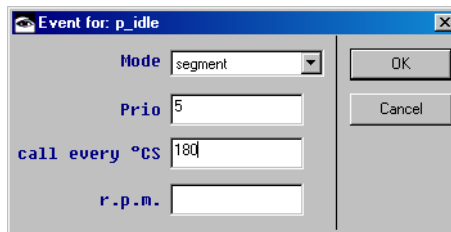
- Öffnen Sie das Dialogfenster „Event“ für das Ereignis.

- Wählen Sie aus dem Kombikästchen „Mode“ den Modus *segment* aus.

Es wird ein Dialogfenster geöffnet, in dem alle Variablen Ihrer Komponente aufgelistet werden.



- Wählen Sie die Variable, die als Segmentvariable dienen soll, und klicken Sie auf **OK**.  
Die Segmentvariable soll die Drehzahl in Umdrehungen pro Minute darstellen.
- Stellen Sie im Feld „call every °CS“ des Dialogfensters „Event“ den Kurbelwinkel ein.
- Klicken Sie auf **OK**.



Das Segmentereignis wird jetzt am Anfang jedes Segmentintervalls  $t_{seg}$  (in Grad) ausgelöst, das sich nach der folgenden Gleichung berechnet:

$$t_{seg}[s] = \frac{CS[deg]}{n \left[ \frac{1}{min} \right] p 6}$$

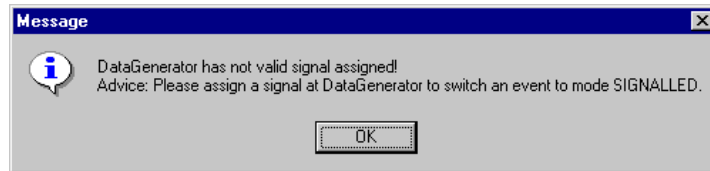
Dabei ist  $n$  die Drehzahl in Umdrehungen pro Minute, und  $cs$  ist der Kurbelwinkel, der die Umdrehung spezifiziert.

## Einrichten eines asynchronen Ereignisses:

---

- Öffnen Sie das Dialogfenster „Event“ für das Ereignis.
- Wählen Sie aus dem Kombikästchen „Mode“ den Modus `signalled` aus.

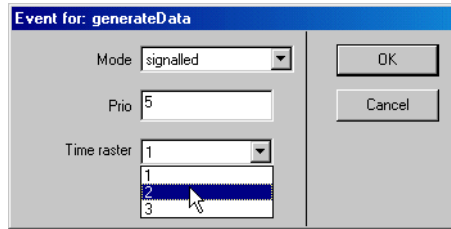
Wenn im Datengenerator (s. „Der Datengenerator“ auf Seite 616) noch kein Signal verfügbar ist, wird folgende Fehlermeldung angezeigt:



- Um den Fehler zu beheben, gehen Sie wie folgt vor:
  - Bestätigen Sie die Fehlermeldung mit **OK**.
  - Wenn Sie das Ereignis direkt einrichten (s. Seite 611), schließen Sie das Dialogfenster „Event“.
  - Klicken Sie im Fenster „Physical Experiment“ auf die Schaltfläche **Open Data Generator**, um den Datengenerator zu öffnen.
  - Definieren Sie ein Signal wie in „Definieren eines Signals:“ auf Seite 621 beschrieben.



- Führen Sie die ersten beiden Schritte der Anweisung erneut aus.

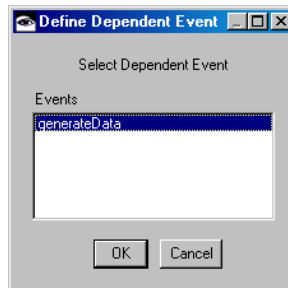


- Wählen Sie das gewünschte Zeitraster aus, falls das Signal mehrere Raster enthält.
- Klicken Sie auf **OK**, um die Einstellung zu übernehmen.

Ein abhängiges Ereignis wird immer dann aktiviert, wenn das Ereignis, von dem es abhängt, aktiviert wird. Dadurch werden die anderen Einstellungen des Ereignisses nicht beeinträchtigt. Es wird immer noch normal aktiviert, zusätzlich zum Aktivieren als abhängiges Ereignis, jedoch erfolgt die Aktivierung synchron mit dem Ereignis, von dem es abhängt.

#### Einrichten eines abhängigen Ereignisses:

- Wählen Sie im Fenster „Event Generator“ ein Ereignis aus.
- Wählen Sie **Channels** → **Dependent Event**. Das Fenster „Define Dependent Event“ geht auf.



- Wählen Sie das Ereignis aus, von dem das erste Ereignis abhängen soll.

- Klicken Sie auf **OK**.

## 6.1.4 Der Datengenerator

Der Datengenerator liefert im Offline-Experiment die Stimuli für die Elemente der Komponente, mit der experimentiert wird. Gewöhnlich handelt es sich bei den angesteuerten Elementen zwar um die Schnittstellenelemente der Komponente, aber mit dem Datengenerator lässt sich jedes beliebige Element ansteuern. Der Datengenerator stellt eine Reihe unterschiedlicher Stimulusmodi wie z.B. Sinuswellen, Impulse usw. bereit.

### **Hinweis**

*Bei Offline-Experimenten stimulieren Sie häufig Variable und Parameter, die tatsächliche Modellwerte benutzen, wenn das Experiment online läuft. Unter Umständen müssen Sie Ihre Experimenteinstellungen entsprechend ändern.*

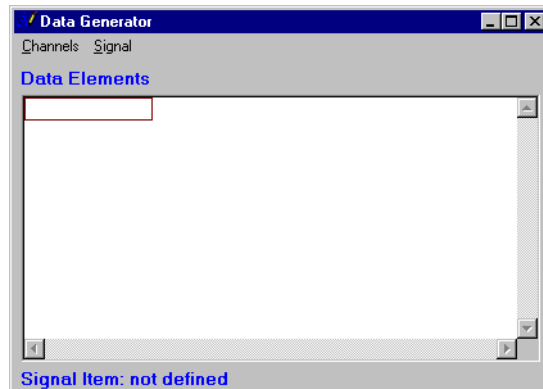
### **Einrichten des Datengenerators:**



- Klicken Sie in der Experimentierumgebung auf die Schaltfläche **Open Data Generator**.

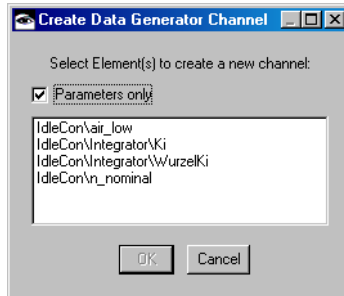
oder

- Wählen Sie **Experiment** → **Data Generator**.  
Das Fenster „Data Generator“ öffnet sich.





- Wählen Sie **Channels** → **Create** zum Öffnen des entsprechenden Dialogfensters.



Standardmäßig enthält die Liste nur die Parameter der aktuellen Komponente. Sie können alle Basiselemente für eine Komponente stimulieren.

- Setzen Sie die Option **Parameters only** inaktiv, um sich alle Basiselemente für die Komponente anzusehen.
- Wählen Sie das Element oder die Elemente, für das/die Sie einen Kanal erstellen möchten. Sie können mehrere Elemente auswählen, indem Sie darauf klicken und dabei die <CTRL>-Taste gedrückt halten.
- Klicken Sie auf **OK**.
- Wiederholen Sie dies für die anderen Elemente, für die Sie Datenkanäle erstellen möchten.

### Einrichten eines Kanals im Datengenerator:

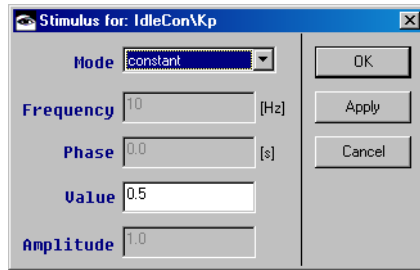
---

- Wählen Sie einen Kanal im Fenster „Data Generator“ aus.
- Wählen Sie **Channels** → **Edit**

oder

- doppelklicken Sie auf den Kanal.

Das Dialogfenster „Stimulus“ wird geöffnet. Es zeigt den Namen des eingerichteten Kanals in seiner Titelleiste an.



- Alternativ wählen Sie in der Experimentierumgebung ein Element in der Fläche „Elements“.
  - Wählen Sie **Stimulate** aus dem Kontextmenü.
- oder
- wählen Sie **Elements** → **Stimulate**.  
Damit wird das Element zum Fenster „Data Generator“ hinzugefügt und das Dialogfenster „Stimulus“ geöffnet.
  - Wählen Sie einen Modus für den Kanal aus.  
Der Modus bestimmt die Art der zu erzeugenden Kurve. Mögliche Modi sind *constant*, *sinus*, *ramp*, *pulse*, *step*, *table*, *matrix*, *random* und *gaussian*.

### Stimulationsmodus einrichten:

---

- Führen Sie die nötigen Schritte für jeden Kanal im Datengenerator aus.

#### Hinweis

Sie können bei laufendem Experiment die Einstellungen für einen Signalkanal ändern. Zum Erproben unterschiedlicher Einstellungen drücken Sie einfach **Apply**, um das Signal ohne Schließen des Dialogs zu ändern.

1. Modus constant einrichten
  - Geben Sie im Dialogfenster „Stimulus“ im Feld „Value“ den Wert der Konstanten ein.  
Alle anderen Einstellmöglichkeiten sind deaktiviert.

- Klicken Sie auf **OK**, um den Wert zuzuweisen und das Dialogfenster zu schließen.
  - Klicken Sie auf **Apply**, um den Wert zuzuweisen, ohne das Dialogfenster zu schließen.
  - Klicken Sie **Cancel**, um die Einstellung zu verwerfen und das Dialogfenster zu schließen.
2. zyklische Modi (`sinus`, `ramp`, `pulse` und `step`) und Zufallsgenerator (`random`; gleichverteilt) einrichten

- Stellen Sie die Frequenz für das Signal ein.
- Stellen Sie die Phase für das Signal ein.

Die Phase bestimmt den Versatz der Zeitachse. Beträgt beispielsweise die Phase eines Sinussignals 0,5 Sekunden, wird das Signal um eine halbe Sekunde nach rechts verschoben.

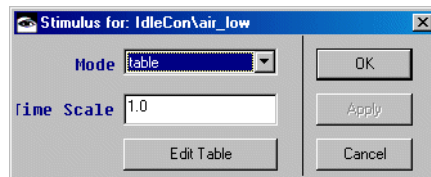
#### **Hinweis**

*Die Felder „Frequency“ und „Phase“ sind für den Modus `random` nicht verfügbar.*

- Stellen Sie den Versatz der y-Achse (`offset`) ein.
- Stellen Sie die Amplitude ein.
- Klicken Sie auf **OK** oder **Apply**.

3. Modus `table` einrichten

Im Modus `table` verwenden Sie eine Tabelle als Stimulus.



- Klicken Sie auf die Schaltfläche **Edit Table**, um die zur Stimulierung verwendete Tabelle zu bearbeiten.

Der Tabelleneditor für eindimensionale Tabellen öffnet sich.

- Geben Sie im Tabelleneditor die gewünschten Werte direkt ein

oder

- wählen Sie **Edit** → **File In Data**, um eine existierende Tabelle zu laden.

Details zum Arbeiten im Tabelleneditor finden Sie in den Abschnitten „Der Editor für zusammengesetzte Typen (Tabelleneditor)“ auf Seite 496 und „Arbeiten mit Verstellfenstern“ auf Seite 665.

- Geben Sie im Feld „Time Scale“ einen Faktor für die Zeitskala ein.

Die Werte der X-Achse der Tabelle werden mit dem eingegebenen Faktor multipliziert und anschließend als Zeitangabe in Sekunden für die Stimulierung verwendet.

- Klicken Sie auf **OK**, um die Einstellungen zu übernehmen.

#### **Hinweis**

*Die Tabelle wird nur einmal durchlaufen. Wenn Sie sie ein weiteres Mal als Stimulus verwenden wollen, müssen Sie das Experiment beenden und neu starten.*

#### 4. Modus `matrix` einrichten

Die Einrichtung des Modus `matrix` ist in den Abschnitten auf Seite 621 und Seite 622 beschrieben.

#### 5. Modus `gaussian` einrichten

Dieser Modus ist nur für numerische Variablen verfügbar. Er stimuliert die Variable mittels eines Zufallszahlengenerators mit Gaußverteilung.

- Geben Sie im Feld „Mean“ den Mittelwert der Gaussverteilung ein.
- Geben Sie im Feld „Variance“ die Varianz der Gaussverteilung ein.
- Klicken Sie auf **OK** oder **Apply**.

## Löschen von Kanälen:

---

- Wählen Sie **Channels** → **Delete** zum Löschen des gewählten Kanals aus dem Datengenerator.
- Wählen Sie **Channels** → **Delete All** zum Löschen aller Kanäle aus dem Datengenerator.

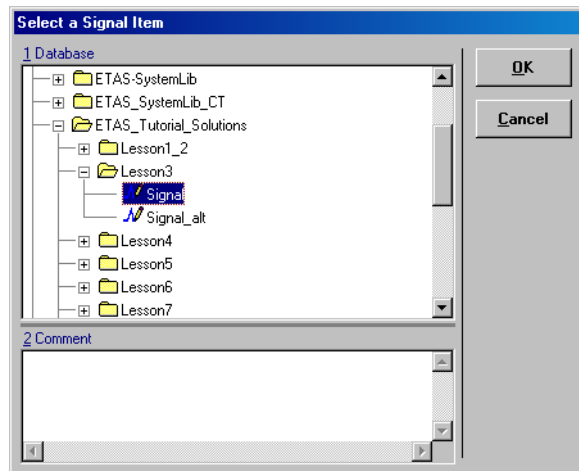
Die Daten tatsächlicher Messungen können als Stimulus im Datengenerator zur Anwendung kommen. Auf diese Weise lassen sich auch beim Offline-Experimentieren Komponenten mit echten Daten erproben. Solche Daten werden in Signaleinträgen in der Datenbank gespeichert und können in vielerlei Formaten gelesen werden. Das Importieren von Signalen wird im Kapitel „Signale und Symbole“ auf Seite 597 beschrieben. Einzelne Kanäle eines Signals können im Datengenerator als Kanäle zugewiesen werden.

## Definieren eines Signals:

---

Ein Signal kann nur definiert werden, wenn das Experiment beendet ist. Gehen Sie wie folgt vor:

- Wählen Sie im Datengenerator **Signal** → **Define Signal**.  
Das Fenster „Select a Signal Item“ wird geöffnet.



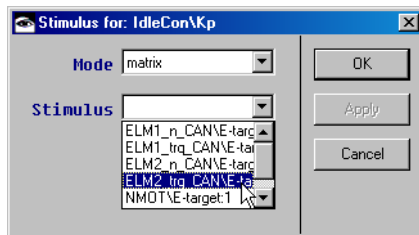
- Wählen Sie in der Fläche „1 Database“ ein Signal aus.

- Klicken Sie auf **OK**.  
Damit wird das Signal festgelegt. Sie können nur einen Signaleintrag pro Experiment verwenden.

### Definieren von Datengeneratorkanälen als Signalkanäle:

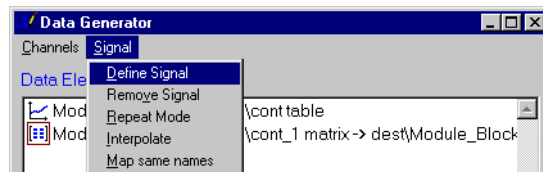
- Wählen Sie im Fenster „Data Generator“ einen Kanal aus.
- Wählen Sie **Channels** → **Edit**  
*oder*
- doppelklicken Sie auf den Kanal.
- Wählen Sie im Dialogfenster „Stimulus“ aus dem Kombikästchen „Mode“ den Modus *matrix*.

Das Kombikästchen „Stimulus“ erscheint unter dem Kombikästchen „Mode“. Es enthält alle im vorher festgelegten Signal enthaltenen Kanäle.



- Wählen Sie den Signalkanal aus, den Sie dem Datengeneratorkanal zuweisen möchten.
- Klicken Sie auf **OK**.

Wenn Sie ein Signal definiert haben (siehe Seite 621), werden im Menü **Signals** des Datengenerators neue Funktionen sichtbar, die allerdings nur bei beendetem Experiment verfügbar sind.

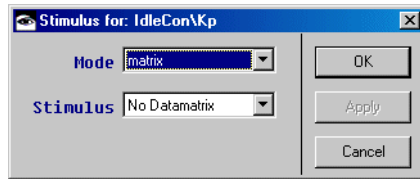


## Signal entfernen:

---

- Wählen Sie im Datengenerator **Signal** → **Remove Signal**.

Das festgelegte Signal wird entfernt. Ein Kanal, der mit dem Signal stimuliert wird, behält den Modus `matrix`, der Eintrag im Kombikästchen „Stimulus“ wird zurückgesetzt.



Wenn die Signalkanäle und die Kanäle im Datengenerator die gleichen Namen haben, brauchen Sie nicht jeden Kanal einzeln im Kombikästchen „Stimulus“ auszuwählen (s. Seite 622). Sie können gleichnamige Kanäle automatisch zuweisen.

## Gleichnamige Kanäle zuweisen:

---

- Wählen Sie für jeden gewünschten Kanal im Dialogfenster „Stimulus“ den Modus `matrix`.

- Wählen Sie im Datengenerator **Signal** → **Map same names**.

Signalkanäle werden gleichnamigen Datengeneratorkanälen zugewiesen.

Jedes Mal, wenn im Experiment Daten generiert werden (s. Seite 610), wird das Signal ausgewertet. Liegt der aktuelle Zeitwert zwischen zwei Signalpunkten, wird dem Kanal standardmäßig der untere Signalwert zugewiesen. Alternativ können Sie lineare Interpolation auswählen.

## Signal interpolieren:

---

- Wählen Sie im Datengenerator **Signal** → **Interpolate**.

- Starten Sie das Experiment.  
Der zugewiesene Wert wird zwischen den beiden den aktuellen Zeitpunkt umgebenden Signalwerten linear interpoliert.

Wenn nichts anderes angegeben wird, läuft das Experiment weiter, wenn das Ende des Signals erreicht ist. Der stimulierte Kanal behält den letzten Signalwert. Sie können einen Wiederholmodus für das Signal oder ein automatisches Beenden der Messung einschalten.

### Hinweis

*Das automatische Beenden hat Vorrang; wenn beide Modi eingeschaltet sind, hält das Experiment am Signalende an.*

### Signalwiederholung einschalten:

---

- Wählen Sie im Datengenerator **Signal** → **Repeat Mode**.
- Starten Sie das Experiment erneut.  
Wenn das Signalende erreicht ist, beginnt die Stimulierung von vorn.

### Automatisches Beenden einstellen:

---

Dieser Befehl ist nur verfügbar, wenn mindestens ein Kanal des Datengenerators mit einem Signal stimuliert wird.

- Wählen Sie im Fenster „Physical Experiment“ **Experiment** → **Automatic Stop**.  
Die Menüfunktion wird mit einem Häkchen markiert. Wenn Sie nun das Experiment starten, hält es automatisch an, wenn das Ende des Signals erreicht ist.
- Wählen Sie erneut **Experiment** → **Automatic Stop**, um das automatische Beenden zu deaktivieren.

Nach Beenden des Experiments und Verlassen der Experimentierumgebung wird die Einstellung von **Experiment** → **Automatic Stop** gespeichert. Wenn Sie das nächste Mal mit der Komponente experimentieren, wird die alte Einstellung übernommen.



## 6.1.5 Das Mess-System

---

Das Mess-System ist für Online- und Offline-Experimente gleich. Es bietet mehrere Möglichkeiten zum Anzeigen der Werte von Elementen wie z.B. ein Oszilloskop, Balkenanzeigen und numerische Anzeigen. Jedes Messfenster ist mit den anzuzeigenden Werten einzurichten. Die Anzeigen bieten mehrere Möglichkeiten der spezifischen Anpassung, die während des Einrichtens des Experiments und auch bei laufendem Experiment genutzt werden können. Alle Messfenster werden geöffnet, indem ihnen Elemente aus der Experimentierumgebung zugewiesen werden. Die Messfenster werden im Kapitel „Messfenster“ auf Seite 697 beschrieben.

### **Zuweisen eines Elements zu einem neuen Messfenster:**

---

- Wählen Sie aus dem Kombikästchen „Measurement Window“ den Typ des Messfensters, den Sie dem Element zuweisen möchten.

Die verfügbaren Typen sind in Klammern aufgelistet, z.B. <1. Oscilloscope>.

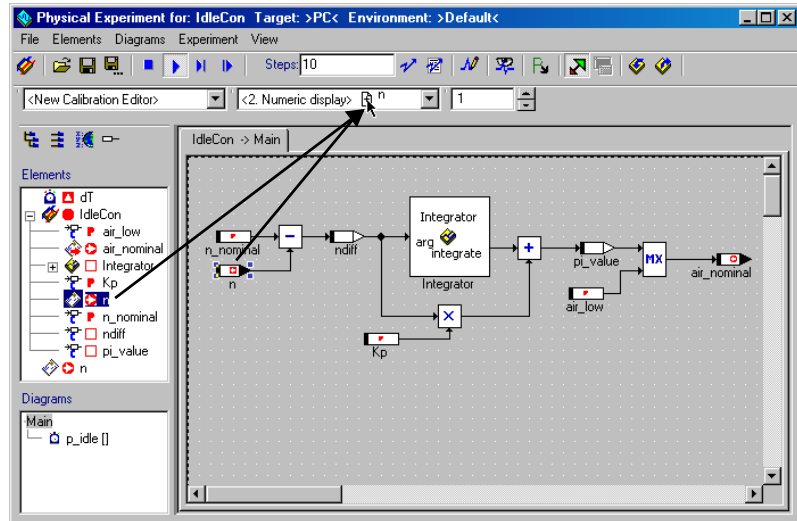
- Wählen Sie in der Fläche „Elements“ das Element, das einem Messfenster zugewiesen werden soll.
- Wählen Sie **Elements** → **Measure**.

*Oder*

- Ziehen Sie das Element aus der Fläche „Elements“ zum Kombikästchen „Measurement Window“.

*Oder*

- Ziehen Sie ein Vorkommen des Elements aus der Blockdiagramm-Anzeige zum Kombikästchen „Measurement Window“.



Ein neues Messfenster wird geöffnet, und das gewählte Element wird zum Messfenster hinzugefügt. Sie können mehrere Elemente auswählen, indem Sie darauf klicken und dabei die <CTRL>-Taste gedrückt halten. In diesem Fall werden alle ausgewählten Elemente dem Fenster zugewiesen.

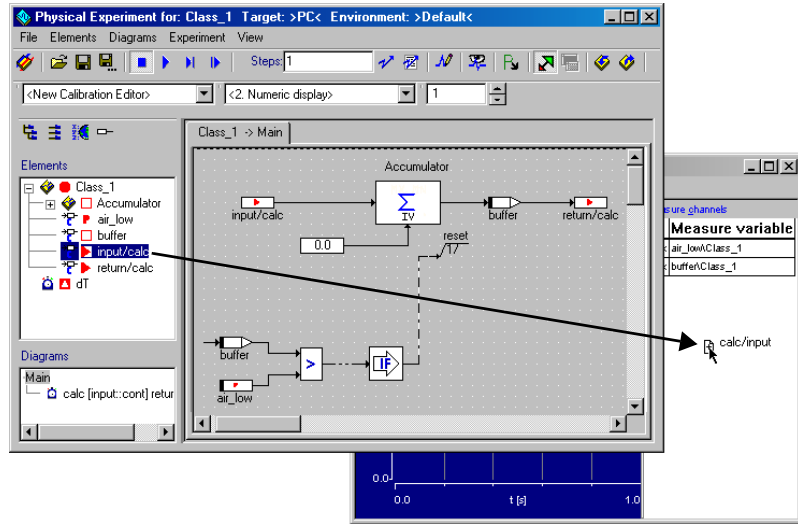
Das Kombikästchen „Measurement Window“ zeigt auch die Titel aller schon geöffneten Messfenster an. Die Einträge für offene Fenster werden ohne Klammern angezeigt, z.B. `oscilloscope; 1`. Haben Sie den Titel eines Messfensters geändert, wird der betreffende Titel im Kombikästchen angezeigt.

#### Zuweisen eines Elements zu einem bestehenden Messfenster:

- Wählen Sie aus dem Kombikästchen „Measurement Window“ den Eintrag für das bestehende Messfenster.
- Ziehen Sie ein oder mehrere Elemente zum Kombikästchen „Measurement Window“.

*Oder*

- Ziehen Sie die Elemente direkt zum Messfenster.



Der Monitor zeigt den aktuellen Wert des Elements über dem ausgewählten Vorkommen im Zeichenbereich an. Details dazu siehe Abschnitt „Monitor“ auf Seite 729.

### Zuweisen eines Monitors zu einem Element:

- Klicken Sie mit der rechten Maustaste auf ein Element im Zeichenbereich.
- Wählen Sie **Monitor** aus dem Kontextmenü.  
Der aktuelle Wert des Elements wird in der Komponentenanzeige angezeigt.

### Schließen aller Messfenster:

- Wählen Sie **Experiment** → **Close Measure Windows** zum Schließen aller aktuell geöffneten Messfenster.

## 6.1.6 Das Verstellsystem

Das Verstellsystem ist für Online- und Offline-Experimente gleich. Im Kombikästchen „Calibration Window“ werden alle aktuell geöffneten Verstellfenster aufgelistet, und es besteht die Möglichkeit zum Öffnen eines neuen Fensters. Im gleichen Fenster können sich mehrere Elemente befinden, jedoch nur unter

der Bedingung, dass sie vom gleichen Typ sind, d.h. mehrere Tabellen können sich in einem Tabelleneditor, mehrere skalare Elemente in einem numerischen Editor befinden usw. Das Zuweisen von Elementen zu Verstellfenstern funktioniert wie das Zuweisen von Elementen zu Messfenstern. Die Verstellfenster sind in Kapitel 6.2 auf Seite 660 beschrieben.

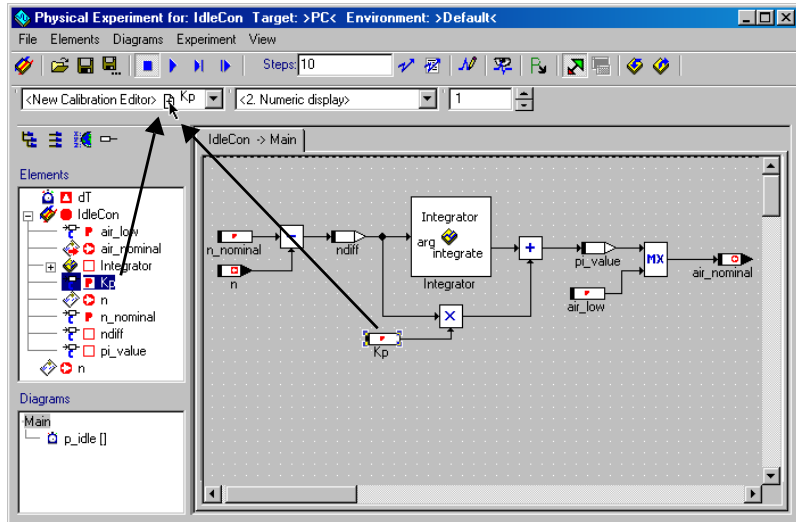
#### **Verstellen eines Elements:**

---

- Wählen Sie in der Fläche „Elements“ der Experimentierumgebung das zu verstellende Element.
- Wählen Sie **Elements** → **Calibrate**  
*oder*
- doppelklicken Sie auf das Element.  
Es wird ein Dateneditor für den gewählten Elementtyp geöffnet.

*Oder*

- Ziehen Sie das Element aus der Fläche „Elements“ oder aus dem Zeichenbereich zum Kombikästchen „Calibration Window“.



Beim Ablegen des Elements auf dem Eintrag <New Calibration Editor> wird ein neues Fenster geöffnet, andernfalls wird das Element zum ausgewählten Fenster hinzugefügt.

Sie können ein Element auch direkt in ein bestehendes Verstellfenster ziehen.

- Wählen Sie **Experiment** → **Update Calibration Windows**, um den Inhalt der Verstellfenster zu aktualisieren.
- Wählen Sie **Experiment** → **Close Calibration Windows** zum Schließen aller Verstellfenster.

### 6.1.7 Durchführen von Offline-Experimenten

Nach dem Einrichten des Experiments können Sie dieses starten. Während das Offline-Experiment läuft, können Sie die Anzeigeeoptionen an allen Messfenstern verändern, Messfenster öffnen und schließen, die Einstellungen im Datengenerator und im Ereignisgenerator verändern, und Sie können auch Datenwerte mit dem Verstellsystem verändern.

## Starten des Offline-Experiments:

---

- Öffnen Sie die Experimentierumgebung für die Komponente (oder das Projekt), mit der Sie experimentieren möchten.
- Stellen Sie im Feld „Measure Display Rate“ die Messhäufigkeit ein.

Bei Einstellen der Messhäufigkeit auf 1 werden alle Messwerte angezeigt. Erfolgt eine Einstellung auf einen höheren Wert, werden nur Mehrfache des betreffenden Wertes angezeigt; wird beispielweise die Messhäufigkeit auf 10 eingestellt, erfolgt nur die Anzeige jedes zehnten Messwerts.



- Klicken Sie auf die Schaltfläche **Expand / Collapse Window**, wenn die Komponentenanzzeige der Experimentierumgebung nicht erscheinen soll.

Nur die Menü- und die Schaltflächenleiste der Experimentierumgebung bleiben sichtbar. Durch erneutes Klicken auf die Schaltfläche wird die Komponente wieder sichtbar.



Darüber hinaus wird die Schaltfläche **Always on top** aktiviert.

- Klicken Sie auf die Schaltfläche **Always on top**, damit das Fenster „Physical Experiment“ immer im Vordergrund bleibt.
- Wählen Sie **Experiment** → **Start Experiment**.

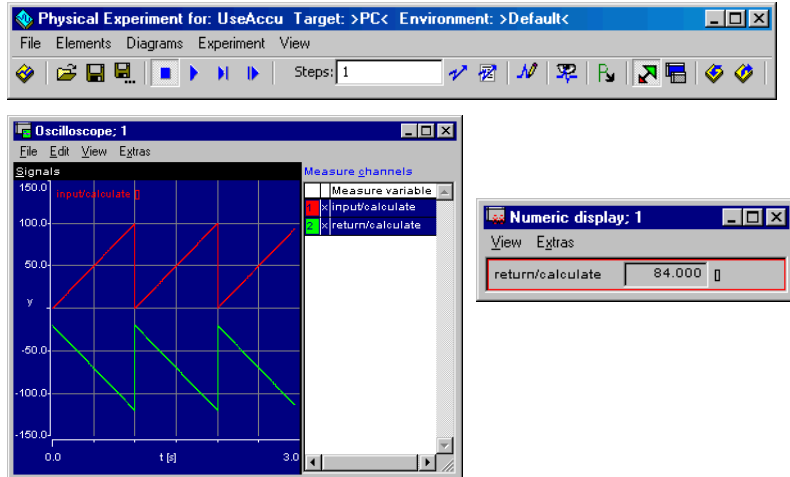
*Oder*



- Klicken Sie auf die Schaltfläche **Start Offline Experiment**.

Wenn Sie in den ASCET-Optionen, Knoten „Experiment“ (s. „Experiment-Optionen“ auf Seite 60), die entsprechenden Optionen gesetzt haben, werden die Variablen und Parameter bei jedem Aufruf dieses Befehls initialisiert.

Das Experiment wird gestartet, und es werden alle Ihrerseits im Mess-System eingerichteten Werte in den entsprechenden Fenstern angezeigt.



- Nehmen Sie alle im Datengenerator, Ereignisgenerator, Mess-System oder Verstellsystem notwendigen Einstellungen vor.

### Beenden des Offline-Experiments:

- Wählen Sie **Experiment** → **Stop Experiment**.

oder



- Klicken Sie auf die Schaltfläche **Stop Offline Experiment**.

Das Experiment wird beendet, aber alle Einstellungen bleiben aktiv. Die Messdaten bleiben im Oszilloskopfenster erhalten, und Sie können nun die Daten analysieren. (Siehe Abschnitt „Analysieren von Messdaten:“ auf Seite 716). Beim Neustarten des Experiments wird die Zeitachse wieder auf 0 gesetzt, ggf. werden Variablen und Parameter initialisiert (s. Seite 60).

## Experimentierumgebung verlassen:

---



- Klicken Sie auf die Schaltfläche **Exit to Component**  
*oder*
- wählen Sie **File** → **Exit**, um die Experimentierumgebung zu verlassen und zurück zum entsprechenden Editor für die von Ihnen simulierte Komponente zu gelangen.  
Sie werden gefragt, ob Sie die aktuelle Experimentierumgebung sichern wollen.
- Aktivieren Sie die Option **Remember my Decicion**, wenn Sie jede Abfrage dieser Art auf dieselbe Weise beantworten wollen.
- Das Fenster „Save Experiment“ wird dann nicht mehr angezeigt. Sie können die Einstellung im ASCET-Optionsfenster, Knoten „Confirmation Dialogs“, wieder rückgängig machen (s. „Optionen für Bestätigungsfenster“ auf Seite 47).
- Bestätigen Sie die Sicherung mit **Yes**.  
Die Experimentierumgebung wird unter dem vorgeschlagenen Namen gesichert. Sie gelangen zurück in den Editor der Komponente.
- Verwerfen Sie die Sicherung mit **No**.  
Sie gelangen zurück in den Editor der Komponente, ohne dass die Experimentierumgebung gesichert wird.
- Klicken Sie **Cancel**, um das Schließen der Experimentierumgebung abzubrechen.

Es ist auch möglich, das Experiment in Einzelschritten durchzuführen. Bei jedem Schritt wird ein Ereignis je nach Priorität aktiviert.

## Schrittweises Abarbeiten eines Experiments:

---

- Wählen Sie **Experiment** → **Pause Experiment**

*oder*





- klicken Sie auf die Schaltfläche **Pause Offline Experiment**.

Das Experiment wird zeitweilig unterbrochen, d.h. es werden keine Berechnungen ange- stellt, aber alle Einstellungen bleiben aktiv. Beim Wiederaufnehmen des Experiments blei- ben alle Datenwerte intakt.

- Wählen Sie **Experiment** → **Step Experi- ment**

oder



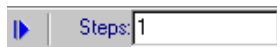
- klicken Sie auf die Schaltfläche **Step Offline Experiment**.

Das Experiment wird im Einzelschrittverfahren abgearbeitet.

Wiederholen Sie nach Bedarf. Sie können auch mit dem schrittweisen Abarbeiten eines Experiments beginnen, wenn dieses nicht läuft. In diesem Fall ändert sich sein Zustand automatisch von *beendet* in *zeitweilig unterbrochen*.

- Vor dem schrittweisen Abarbeiten des Experi- ments können Sie im Feld „Steps“ die Anzahl der auszuführenden Schritte einstellen.

Wird beispielsweise die Anzahl der Schritte auf 5 gestellt, werden fünf Berechnungs- schritte ausgeführt.

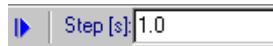


In der aktuellen Version von ASCET können Sie zwischen drei verschiedenen Schrittmodi wählen. Ein Experiment läuft entweder über eine Anzahl von Schritten, eine Anzahl von Sekunden oder bis zum Erreichen einer Haltebedin- gung.

## Umschalten auf getakteten Zeitmodus:

---

- Wählen Sie in der Experimentierumgebung **Experiment** → **Event Generator** → **Step Mode** → **Timed [s]** zum Aktivieren des getakteten Zeitmodus.



Die Experimentierumgebung zeigt ein Textfeld zum Eingeben der Anzahl von Sekunden, über die das Experiment laufen soll. Sie können die Anzahl von Sekunden spezifizieren und danach das Experiment wie oben beschrieben schrittweise abarbeiten.

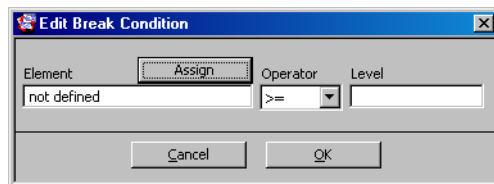
Bei der dritten Möglichkeit zum schrittweisen Abarbeiten eines Experiments unter Verwendung einer Haltebedingung müssen Sie für ein Element in Ihrer Komponente eine Bedingung spezifizieren (z.B. Anzahl der Umdrehungen ist größer als 5000). Das Experiment läuft, bis die Bedingung als wahr (true) bewertet wird.

## Einrichten einer Haltebedingung:

---

- Wählen Sie in der Experimentierumgebung **Experiment** → **Event Generator** → **Step Mode** → **Break At Condition** zum Aktivieren des Schrittmodus mit Haltepunkt.

Das Fenster „Edit Break Condition“ wird angezeigt. Hier können Sie die Bedingung spezifizieren, bis zu der das Experiment laufen soll.



Dieser Editor kann später mit **Experiment** → **Event Generator** → **Edit Break Condition** geöffnet werden.

- Klicken Sie auf die Schaltfläche **Assign**, um den Namen des Elements zu wählen, das in der Bedingung verwendet werden soll.  
Ein Dialogfenster zur Auswahlanforderung zeigt eine Liste von Elementen an.
- Wählen Sie das gewünschte Element und klicken Sie auf **OK**, um den Elementnamen zu benutzen.
- Bearbeiten Sie den Rest der Bedingung durch Auswahl eines Operators aus der Liste und Eingeben des entsprechenden Wertes.
- Klicken Sie zum Bestätigen auf **OK**.



Die Experimentierumgebung zeigt ein Textfeld mit einer Haltebedingung an. Sie können das Experiment wie oben beschrieben schrittweise abarbeiten.

Sie können aus der Experimentierumgebung heraus jederzeit die Implementierung der Komponente ansehen, mit der Sie experimentieren. Ebenso können Sie die Implementierung jedes darin enthaltenen Elements ansehen. Dabei spielt es keine Rolle, ob das Experiment läuft oder angehalten wurde.

### **Implementierung ansehen:**

- Wählen Sie in der Liste „Elements“ das Element oder die Komponente, deren Implementierung Sie ansehen wollen.
- Wählen Sie **Elements** → **Show Implementation**.

### **Hinweis**

*Wenn Sie mehr als ein Element ausgewählt haben, ist der Befehl deaktiviert.*

Der Implementierungseditor für das ausgewählte Objekt öffnet sich. Die Bedienung ist in Kapitel 4.12 „Bearbeiten von Implementierungen“ beschrieben; beachten Sie bitte, dass die Schaltfläche **OK** deaktiviert ist und Sie keine Änderungen vornehmen können.

In C-Code spezifizierte Komponenten bieten zusätzliche Möglichkeiten zur Anzeige von Debugger-Informationen oder Fehlermitteilungen während des Experimentierens. Sie können Debugger- oder Fehlermitteilungen in Ihrem C-Code einbetten. Debugger-Informationen werden im Debugger-Fenster angezeigt, das sich während des Experimentierens öffnen lässt. Fehlermitteilungen werden im Monitorfenster von ASCET angezeigt.

Beim Bearbeiten von C-Code sind die Funktionen `asdWriteUserDebug()` und `asdWriteUserError()` zum Spezifizieren anzuzeigender Informationen zu verwenden. Beide Funktionen verwenden als Argument eine Zeichenkette (String), die die anzuzeigende Mitteilung enthält. Eine typische Aussage könnte wie folgt aussehen:

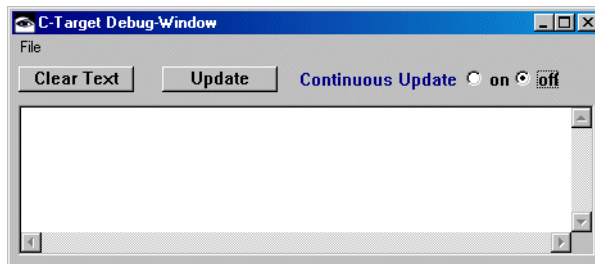
```
asdWriteUserError("Overflow: \n Upper Limit exceeded.")
```

Das String-Argument folgt den standardmäßigen Regeln von ANSI-C; das Beispiel wird in zwei Zeilen ausgedruckt bzw. angezeigt.

### Ansehen von Debugger-Informationen:

---

- Wählen Sie **Experiment** → **Open Target Debugger** zum Öffnen des Fensters „C-Target Debug-Window“.



#### 1. automatische Anzeige

- Aktivieren Sie im Fenster „C-Target Debug-Window“ die Option **on**
- oder
- aktivieren Sie die Menüfunktion **File** → **Continuous Update**.

Die Debugger-Informationen werden sofort bei Erzeugung angezeigt und laufend aktualisiert.

## 2. manuelle Anzeige

- Aktivieren Sie die Option **off**  
*oder*
- deaktivieren Sie die Menüfunktion **File** → **Continuous Update**, um ein ständiges Aktualisieren zu deaktivieren.  
Bei deaktiviertem ständigem Aktualisieren werden neue Informationen auf Wunsch angezeigt.
- Klicken Sie dazu auf die Schaltfläche **Update**  
*oder*
- wählen Sie **File** → **Update Text**.

## 3. Textfenster löschen

- Klicken Sie auf die Schaltfläche **Clear Text**  
*oder*
- wählen Sie **File** → **Clear Text**, um den Inhalt des Fensters „C-Target Debug-Window“ zu löschen.

## 4. Textfenster speichern

- Wählen Sie **File** → **Save as**.  
Das Dateiauswahlfenster öffnet sich.
- Wählen Sie einen Pfad und einen Dateinamen.
- Klicken Sie auf **Speichern**.  
Der Inhalt des Fensters „C-Target Debug-Window“ wird als Textdatei unter dem angegebenen Namen und Pfad gespeichert.

Sie können mit dem Ereignisprotokoll überwachen, welche Ereignisse in welcher Reihenfolge ausgelöst werden. Das Ereignisprotokoll zeigt für jeden Zeitpunkt, zu dem Ereignisse ausgelöst werden, die Zeit in Sekunden an. Für jeden Punkt werden die Ereignisse in der Reihenfolge aufgelistet, in der sie ausgelöst wurden.

### **Überwachen von Ereignissen mit dem Ereignisprotokoll:**

---

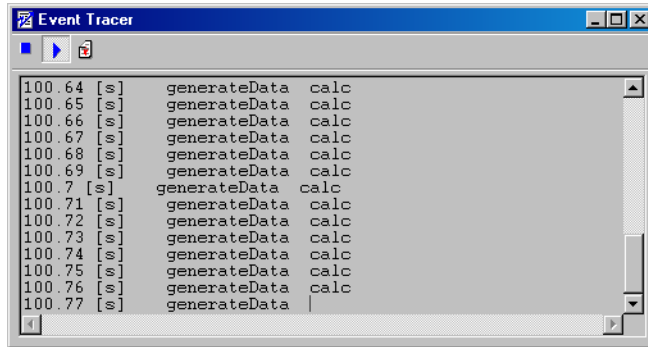
- Richten Sie die Experimentierumgebung für die Komponente ein, mit der Sie experimentieren möchten.



- Klicken Sie auf die Schaltfläche **Open Event Tracer**.

Oder

- Wählen Sie **Experiment** → **Event Tracer** zum Öffnen des Fensters „Event Tracer“.



- Klicken Sie im Fenster "Event Tracer" auf die Schaltfläche **Start Tracer**.

Das Ereignisprotokoll wird gestartet.

- Starten Sie in der Experimentierumgebung das Experiment.



- Zum Beenden des Protokollierens von Ereignissen klicken Sie auf die Schaltfläche **Stop Tracer**.

Sie können das Protokollieren von Ereignissen bei laufendem Experiment starten und beenden.



- Klicken Sie auf **Reset Tracer**, um den Inhalt des Fensters „Event Tracer“ zu löschen.

## 6.1.8 Laden und Speichern von Umgebungen

Experimente für komplexe Modelle können aus zahlreichen Messfenstern und Messkanälen bestehen. Folglich ist es von Nutzen, wenn die Experimentierumgebung gespeichert werden kann, so dass eine Wiederverwendung der Einstellungen bei anderen Experimentiersitzungen möglich ist. In ASCET können Sie mehrere solche Umgebungen für jede Komponente speichern.

Beim Speichern einer Umgebung werden alle Einstellungen im Datengenerator und im Ereignisgenerator sowie im Data-Logger gespeichert. Des Weiteren werden alle geöffneten Verstelleditoren und Messfenster gespeichert, wobei ihre Einstellungen nicht beeinträchtigt werden. Sie können die Umgebung später laden, und die Experimentierumgebung wird wiederhergestellt. Es gibt immer ein Standardexperiment, das geladen wird, wenn keine andere Umgebung definiert wurde.

### Speichern einer Umgebung:

---

- Öffnen Sie die Experimentierumgebung und nehmen Sie alle notwendigen Einstellungen vor.



- Klicken Sie auf die Schaltfläche **Save Environment**

oder

- wählen Sie **File** → **Save Environment** zum Speichern der aktuellen Umgebung.

### Speichern einer Umgebung unter einem anderen Namen:

---

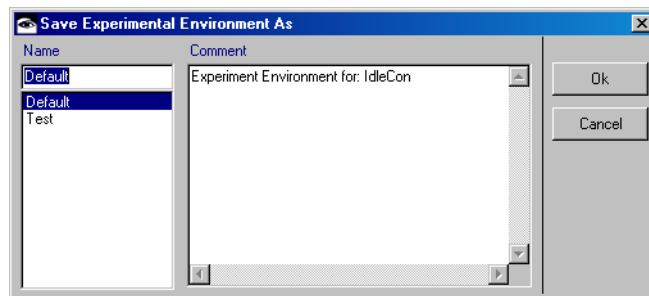


- Klicken Sie auf die Schaltfläche **Save Environment As**

oder

- wählen Sie **File** → **Save Environment As**.

Das Dialogfenster „Save Experimental Environment As“ wird geöffnet.

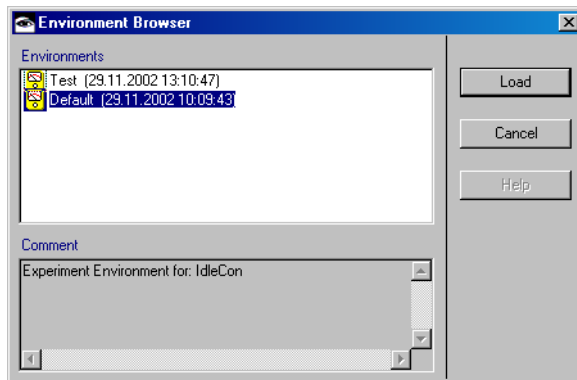


- Geben Sie einen Namen für das Experiment ein.

- Geben Sie in der Fläche „Comment“ einen Kommentar ein, der das Experiment beschreibt.
- Klicken sie auf **OK**.  
Das Experiment wird unter dem eingegebenen Namen gespeichert.

### Laden einer Umgebung:

- Öffnen Sie die Experimentierumgebung.  
Gibt es nur die standardmäßige Umgebung, wird diese beim Öffnen der Experimentierumgebung automatisch geladen. Haben Sie für die Komponente mehrere Umgebungen gespeichert, wird der Umgebungs-Browser mit einer Liste aller verfügbaren Umgebungen geöffnet.



- Wählen Sie die zu ladende Umgebung aus.
- Klicken Sie auf **Load**.  
Die Experimentierumgebung wird mit der gewählten Umgebung geöffnet.

Es können Umgebungen bei schon geöffneter Experimentierumgebung geladen werden. In diesem Fall bleiben die aktuell geöffneten Mess- und Verstellfenster offen, es werden aber alle anderen Einstellungen aus der geladenen Umgebung aktiv, und alle darin definierten Fenster werden geöffnet.



## Umschalten von Umgebungen:

---



- Klicken Sie auf die Schaltfläche **Load Environment**

oder

- wählen Sie **File** → **Load Environment**.  
Das Fenster „Environment Browser“ öffnet sich.
- Wählen Sie die zu öffnende Umgebung.
- Klicken Sie auf **Load**.

## Exportieren von Umgebungen:

---

- Wählen Sie **File** → **Export Experiment**.  
Es wird das Dialogfenster „Export File“ geöffnet.
- Geben Sie für die Exportdatei einen Pfad- und Dateinamen ein.
- Klicken sie auf **Speichern**.  
Die aktuell in der Komponente definierten Umgebungen werden exportiert. Verwenden Sie den Importmechanismus des Komponentenmanagers für das Importieren von Umgebungen. Die Anweisungen für den Import von Umgebungen gelten analog für den Import anderer Daten.

### 6.1.9 Der Data-Logger

---

Mit dem Data-Logger können Sie die Werte von Variablen innerhalb einer Komponente oder eines Projekts während des Offline-Experimentierens bzw. innerhalb eines Projekts während des Online-Experimentierens aufzeichnen. Die Werte werden in eine Datei geschrieben und können später mit einer Anwendung zur Messwertanalyse (z.B. der MDA) analysiert werden. Es gibt drei Aufzeichnungsarten:

- **Transient Sampling**

Für *Transient Sampling* müssen die Einstellungen der Codegenerierung angepasst werden. Dann wird der Code so erzeugt, dass bei jedem Ändern des Werts einer protokollierten Variablen die betreffende Änderung automatisch aufgezeichnet wird. Auf diese Weise lassen sich alle Änderungen eines Werts aufzeichnen. Da der Code verändert wird,

beeinflusst das Transient Sampling das Laufzeitverhalten des Modells. Der Umfang vom Data-Logger erzeugter Daten nimmt zu, da für jeden aufgezeichneten Wert ein *Zeitstempel* erzeugt werden muss.

- **Periodic Sampling**

*Periodic Sampling* erfordert keine Codeänderungen, und folglich wird dadurch das Laufzeitverhalten des Modells nur minimal beeinflusst. Hier wird die Aufzeichnung durch eine spezielle Task ausgelöst, d.h. bei jedem Auslösen einer gewählten Task wird der aktuelle Wert aller registrierten Kanäle aufgezeichnet. Dadurch wird das Laufzeitverhalten der gewählten Task nicht beeinflusst, da die Aufzeichnung erst nach Abschluss der Task erfolgt. Ändert sich der Wert einer protokollierten Variablen mehrfach zwischen aufeinanderfolgenden Aufzeichnungsvorgängen, wird nur die letzte Änderung aufgezeichnet. Es brauchen keine Zeitstempel erzeugt zu werden, da die Aufzeichnung in festgelegten Zeitabständen erfolgt.

Die aufgezeichneten Daten werden in einem Ringpuffer im Target gespeichert und nach dem Ende der Aufzeichnung an den PC-Host übertragen, wo sie ebenfalls in einem Ringpuffer gespeichert werden. Der Ringpuffer speichert eine festgelegte Anzahl von Werten, und bei Überschreiten dieser Anzahl werden die vorher gespeicherten Werte nach dem FIFO-Prinzip überschrieben. Daher können nur über einen begrenzten Zeitraum Daten aufgezeichnet werden. Sie können das Problem umgehen, indem Sie *Continuous Polling* aktivieren. Dann werden die Daten während des Aufzeichnungsvorgangs in regelmäßigen Abständen an den Host übertragen. Allerdings kann die Kommunikation mit dem Host den Targetprozessor beeinflussen. Die Vor- und Nachteile sind für jede Anwendung abzuwägen.

- **Periodic to File**

Wird eine längere Aufzeichnungsdauer gewünscht, kann das Target in der Regel nicht genug RAM für die Aufzeichnung zur Verfügung stellen. Deshalb wählen Sie in diesem Fall *Periodic to File*. Auch hier wird das Aufzeichnen durch eine ausgewählte Task ausgelöst, aber anders als beim *Periodic Sampling* werden die Daten in regelmäßigen Abständen an den PC-Host übertragen. Auf diese Weise können Daten über lange Zeiträume erfasst werden.

Für jede der drei Arten werden die Daten nach Abschluss der Datenregistrierung (s. Seite 653) vom PC-Host in eine Datei geschrieben.

Bezüglich der Anzahl der protokollierbaren Kanäle sowie der Geschwindigkeit, mit der die Datenregistrierung erfolgen kann, gibt es drei begrenzende Faktoren:

- Ein Teil des *RAM des Targets* ist für die Speicherung der aufgezeichneten Daten vorgesehen. Je größer der freie Bereich des RAM, desto mehr Werte können aufgezeichnet werden.
- Die Aufzeichnung der Daten wird durch die *Datenübertragung* zwischen Host und Target beeinflusst. Verfügt beispielsweise das Target nur über einen kleinen RAM, aber die Datenübertragung an den Host kann sehr schnell erfolgen, dann lassen sich mehr Werte registrieren.
- Die aufgezeichneten Werte werden auf dem Host-RAM (physisch und virtuell) gespeichert; je mehr freien RAM der PC hat, desto mehr Daten lassen sich speichern.

Ehe Sie Daten in dem Modus *Transient Sampling* aufzeichnen können, müssen Sie die Einstellungen für Codegenerierung verändern.

### **Hinweis**

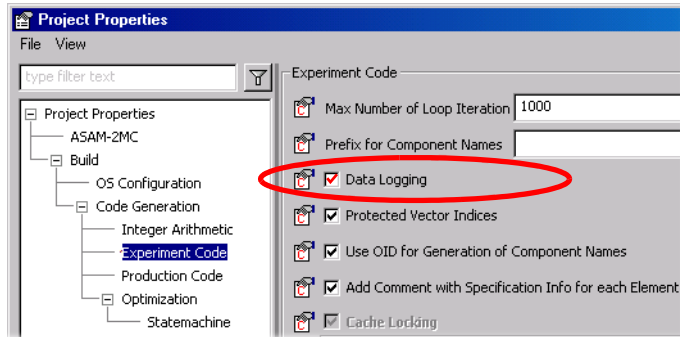
Der Modus ***Transient Sampling*** kann nur aus einem Projekt heraus aktiviert werden.

### **Vorbereiten zum Transient Sampling:**



- Öffnen Sie das Projekt, mit dem Sie experimentieren wollen.
- Klicken Sie im Projekteditor auf die Schaltfläche **Project Properties**.  
Das Dialogfenster „Project Properties“ wird im Knoten „Build“ geöffnet.
- Wechseln Sie in den Knoten „Experiment Code“.

- Aktivieren Sie die Option **Data Logging**, um die Datenaufzeichnung einzuschalten..



- Klicken sie auf **OK**.
- Erzeugen Sie den Code für das Projekt und öffnen Sie die Experimentierumgebung, die Sie verwenden wollen.

Möchten Sie Daten im Modus *Periodic Sampling* oder *Periodic to File* aufzeichnen, brauchen sie die Einstellungen für die Codegenerierung nicht zu verändern. Code lässt sich mit oder ohne aktivierte Datenaufzeichnung generieren. Generieren Sie Code mit aktivierter Datenaufzeichnung, läuft der Code ungeachtet laufender oder nicht laufender Datenaufzeichnung langsamer.

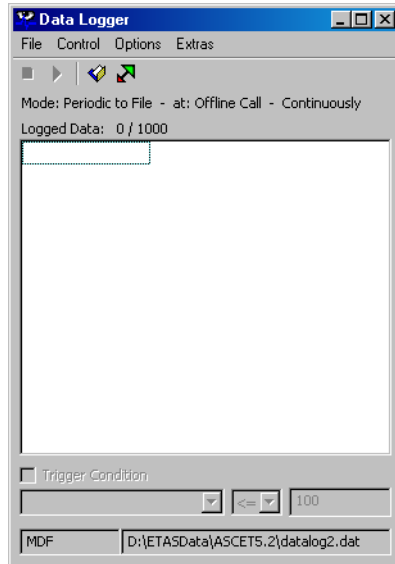
#### Den Data-Logger öffnen:

- Öffnen Sie die Experimentierumgebung für die Komponente oder das Projekt.
- Klicken Sie auf die Schaltfläche **Open Data Logger**



oder

- wählen Sie **Experiment** → **Data Logger**.  
Das Fenster „Data Logger“ öffnet sich.



Das Fenster enthält folgende Elemente:

- Menü **File**
  - *Open* (<CTRL>+<O>)
 





Öffnet eine Liste (im INCA-Format \* .1ab) mit aufgezeichneten Elementen, siehe Seite 647.
  - *Save* (<CTRL>+<S>)
 

Speichert eine Liste (im INCA-Format \* .1ab) der derzeit im Data-Logger erfassten Elemente.
  - *Save As*

Speichert eine Liste (im INCA-Format \* .1ab) der derzeit im Data-Logger erfassten Elemente unter einem beliebigen Namen.
  - *Import*

Importiert eine Liste (Format \* .csv oder SelectX – \* .cfq) mit aufgezeichneten Elementen.
  - *Exit*

Schließt den Data-Logger.

- Menü **Control**
  - *Enable Logging*  
Startet die Aufzeichnung.
  - *Disable Logging*  
Beendet die Aufzeichnung.
- Menü **Options**
  - *Logging Options*  
Öffnet das Fenster „Logging Options“, siehe Seite 648.
  - *Change Filename*  
Ändert den Standardnamen für die Aufzeichnungsdatei.
- Menü **Extras**
  - *Convert MDF to FAMOS*  
Wandelt MDF-Daten in FAMOS-Daten, siehe Seite 655.
- Schaltflächenleiste
  -  Beendet die Aufzeichnung.
  -  Startet die Aufzeichnung.
  -  Ändert den Standardnamen für die Aufzeichnungsdatei.
  -  Zeigt/verbirgt den Data-Logger unterhalb von Menüzeile und Schaltflächenleiste.
- Feld „Logged Data“  
Liste der aufgezeichneten Elemente.
- Option **Trigger Condition**  
Schaltet die Verwendung einer Triggerbedingung ein/aus (s. Seite 652).
- Triggerdefinition  
Zwei Kombikästchen und ein Eingabefeld für die Definition der Triggerbedingung (siehe Seite 652).
- Statuszeile  
Statusinformationen für den Data-Logger.

#### **Einrichten der zu registrierenden Kanäle:**

---

- Wählen Sie in Experimentierumgebung in der Fläche „Elements“ die gewünschten Elemente.

- Wählen Sie **Elements** → **Log**, um die Aufzeichnung der ausgewählten Elemente zu aktivieren.  
Wenn Sie den Data-Logger noch nicht geöffnet haben, öffnet er sich jetzt. Die ausgewählten Elemente sind im Feld „Logged Data“ aufgelistet.
- Sie können auch Elemente direkt in den Data-Logger ziehen.
- Wählen Sie **Elements** → **Log All**, um die Aufzeichnung für alle Elemente zu aktivieren.

### **Hinweis**

*Die Anzahl der aufgezeichneten Elemente wird durch verschiedene Faktoren begrenzt, siehe Seite 642.*

### **Label-Liste verwenden:**

Wenn Sie den Data-Logger schließen, geht die Einrichtung der aufzuzeichnenden Elemente verloren. Um die Einrichtung bei einem weiteren Experiment leicht wiederherzustellen, können Sie eine Liste (Format \* .1a.b) der Elemente anlegen, die im Data-Logger enthalten sind. Diese Liste kann auch für den Austausch mit INCA verwendet werden, wenn etwa sichergestellt werden soll, dass dort dieselben Elemente verwendet werden wie in ASCET.

- Öffnen Sie den Data-Logger (Seite 644).
- Richten Sie die Kanäle ein, die Sie registrieren wollen (Seite 646).
- Wählen Sie **File** → **Save**, um eine Liste der Elemente zu speichern.  
Wenn Sie das erste Mal eine Liste speichern, werden sie nach Pfad und Name für die Datei gefragt.
- Wählen Sie **File** → **Save As**, um eine Liste der Elemente unter einem beliebigen Namen zu speichern.

- Wählen Sie **File** → **Open**, um eine Liste der Elemente im Data-Logger zu öffnen.  
Wenn die Liste Elemente enthält, die im aktuellen Experiment nicht vorhanden sind, werden im Monitorfenster entsprechende Meldungen angezeigt.
- Wählen Sie **File** → **Import**, um eine Liste im Format \* .csv (comma-separated values) oder \* .ctf (SelectX-Format) zu importieren.  
Sie können diese Formate nutzen, um außerhalb von ASCET Label-Listen zu erstellen und verwalten.

### Einstellen der Aufzeichnungsoptionen:

#### 1. Logging Mode

- Wählen Sie im Data-Logger **Options** → **Logging Options**, um das Dialogfenster „Logging Options“ zu öffnen.



- Aktivieren Sie eine der Optionen **Transient Sampling**, **Periodic Sampling** oder **Periodic to File**.

Sie können **Transient Sampling** nur wählen, wenn in den Codegenerierungseinstellungen die Datenaufzeichnung aktiviert wurde (s. Seite 643).

- Wählen Sie eine Task aus dem Kombikästchen „Log at“.

### **Hinweis**

Wenn **Transient Sampling** gewählt wurde, ist das Kombikästchen „Log at“ nicht sichtbar.

Bei *Online*-Experimenten zeigt das Kombikästchen alle im Projekt definierten Tasks an. Das Aufzeichnen erfolgt dann stets nach dem Ausführen der gewählten Task.

Bei *Offline*-Experimenten findet das Aufzeichnen stets nach jedem im Ereignisgenerator definierten Ereignis statt. Sie müssen in diesem Fall nichts auswählen.

- Stellen Sie im Feld „Host Logging Buffer“ die Kanallänge für den Aufzeichnungspuffer des Hosts ein.

### **Hinweis**

Wenn **Periodic to File** gewählt wurde, ist das Feld „Host Logging Buffer“ deaktiviert.

Mit dieser Einstellung wird festgelegt, wieviele Werte im Ringpuffer auf dem Host-PC für jeden Kanal gespeichert werden.

## 2. Datenübertragung zum Host

- Setzen Sie die Option **Continuous Polling** je nach Wunsch aktiv oder inaktiv.

### Hinweis

Wenn **Periodic to File** gewählt wurde, ist **Continuous Polling** automatisch aktiviert..

Bei deaktiviertem **Continuous Polling** werden die Daten im Puffer des Targets gespeichert und erst nach Abschluss eines Registriervorgangs an den Host übertragen.

- Stellen Sie im Feld „Cycle Time (Cont. Polling)“ die Zykluszeit für **Continuous Polling** ein.

### Hinweis

Wenn **Continuous Polling** deaktiviert ist, ist das Feld „Cycle Time (Cont. Polling)“ nicht verfügbar.

Bei aktiviertem **Continuous Polling** werden die im Ringpuffer des Targets gespeicherten Daten in den im Feld „Cycle Time (Cont. Polling)“ festgelegten Abständen an den Host-PC übertragen.

Kann die Zykluszeit aufgrund einer zu langen Transportdauer nicht eingehalten werden, wird in der Titelzeile des Data-Logger die Überschreitung angezeigt. Die Optimierung muss von Hand erfolgen, da Rechnergeschwindigkeit, Targetauslastung und Kommunikationsart großen Einfluss haben.

- Stellen Sie im Feld „Data Rate per Channel and Cycle“ eine Datenrate ein.

Durch die Datenrate wird festgelegt, wieviele Werte pro Abrufintervall für jeden Kanal gelesen werden.

Bei deaktiviertem **Continuous Polling** entspricht das Abrufintervall der gesamten Aufzeichnungszeit, bei aktiviertem **Continuous Polling** der in „Cycle Time (Cont. Polling)“ eingegebenen Zykluszeit.

### 3. Aufzeichnungspuffer des Targets

- Nehmen Sie in den Feldern „Target Logging Buffer“ und „Total Amount“ die Einstellungen für den Aufzeichnungspuffer des Targets vor.

Das Target verfügt über eine maximale RAM-Größe zur Verwendung als Puffer für registrierte Daten. Möchten Sie weniger RAM nutzen, können Sie dies entsprechend einstellen. Durch die Einstellung der Kanallänge wird festgelegt, wieviele Werte für jeden Kanal im Ringpuffer des Targets gespeichert werden.

### 4. Ausgabeformat

- Wählen Sie ein Speicherformat für die Protokolldatei, indem Sie entweder auf die Option **FAMOS** oder **MDF** klicken.
- Klicken Sie auf **OK**.

Mit den Einstellungen in diesem Dialogfenster wird die Anzahl der gleichzeitig protokollierbaren Kanäle festgelegt. Die Anzahl wird unten im Dialogfenster angezeigt und bei jedem Ändern von Einstellungen aktualisiert. Auch der aktuelle beschränkende Faktor wird hier angezeigt. Bei Eingabe einer Einstellung, die durch andere Einstellungen gesetzte Grenzwerte oder die verfügbaren Möglichkeiten überschreitet, erfolgt die Anzeige einer Fehlermeldung und die Einstellung wird nicht verändert.

### Aufzeichnen von Daten:



- Starten sie das Experiment.
- Klicken Sie im Data-Logger auf die Schaltfläche **Enable Logging**

*oder*

- wählen Sie **Control** → **Enable Logging**.  
Die Aufzeichnung der ausgewählten Elemente wird gestartet.  
Wenn Sie diesen Befehl vor dem Start des Experiments in der Experimentierumgebung geben, wird die Aufzeichnung eingeleitet. Erst mit dem Start des Experiments werden tatsächlich Daten aufgezeichnet.

Auf diese Weise beginnt die Aufzeichnung bei laufendem Experiment sofort. Sie haben die Möglichkeit, über die Option **Trigger Condition** im Data-Logger eine Bedingung für den Start der Aufzeichnung zu definieren. Gehen Sie dazu wie folgt vor:

#### **Triggerbedingung definieren:**

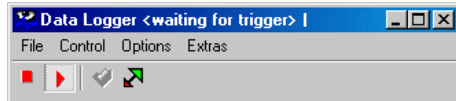
---

- Aktivieren Sie im Data-Logger die Option **Trigger Condition**.  
Diese Option ist nur bei angehaltener Aufzeichnung verfügbar, wenn mindestens ein Element im Feld „Logged Data“ eingetragen ist.

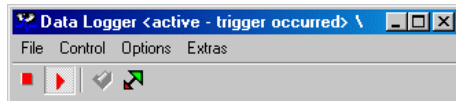
The screenshot shows a dialog box titled "Trigger Condition". It has a checked checkbox "Trigger Condition". Below it, there are three dropdown menus: the first contains "air\_nominal", the second contains ">=", and the third contains "380.0". At the bottom, there are two text input fields: the left one contains "MDF" and the right one contains "D:\ETASData\ASCET5.2\data\log".

- Wählen Sie aus dem linken Kombikästchen eine der registrierten Variablen (z.B. `air_nominal`).
- Wählen Sie im mittleren Kombikästchen einen Vergleichsoperator.  
Sie haben die Wahl zwischen  $\geq$  (größer oder gleich) und  $\leq$  (kleiner oder gleich).
- Geben Sie im rechten Feld einen Schwellenwert ein (z.B. 380).

Wenn Sie nun die Aufzeichnung starten, wartet der Data-Logger, bis die Bedingung erfüllt wird. Der Status wird in der Titelzeile des Data-Loggers angezeigt.



Wenn die Bedingung erfüllt ist, beginnt die Aufzeichnung. Dies wird wiederum in der Titelzeile angezeigt. Die Aufzeichnung läuft so lange weiter, bis sie abgeschaltet wird, auch wenn die Bedingung zwischenzeitlich nicht mehr erfüllt ist.



### Aufzeichnung beenden:



- Klicken Sie auf die Schaltfläche **Disable Logging**

*oder*

- wählen Sie **Control** → **Disable Logging**.

Die Aufzeichnung von Daten wird beendet und das Dialogfenster „Logging Information“ öffnet sich.

A screenshot of a dialog box titled "Logging Information for: <datalog1.dat>". The dialog has a grey background and a blue title bar. It contains several input fields: "User" with the text "kemolth", "Company" with "ETAS GmbH", "Project" with "ControllerTest", and "Tool" with "Ascet V5.0.0". Below these fields is a "Comment" section with a text area containing "Date: (29.11.2002 13:51:06)". On the right side of the dialog, there are two buttons: "Save" and "Discard".

- Füllen Sie die Informationsfelder nach Bedarf aus.

Alle Angaben in diesem Dialogfenster sind optional.

- Klicken Sie auf die Schaltfläche **Save**, um die Datei zu speichern,

oder

- klicken Sie auf **Discard**, um die Datei zu verwerfen.

Die Daten werden bei Beendigung jedes Aufzeichnungsvorgangs in der Protokolldatei gespeichert. Dabei wird standardmäßig der Dateiname `data-log<n>.dat` vergeben. `<n>` steht dabei für eine ganze Zahl, die bei jedem Abspeichern hochgezählt wird, wenn in den ASCET-Optionen, Knoten „Data-logger“ (s. Seite 60), die Option **Auto Increment DataLogger File Name** aktiviert ist. Ist die Option deaktiviert, wird die Datei bei der nächsten Aufzeichnung überschrieben. Diese Datei wird im Produktdatenverzeichnis Ihrer ASCET Installation, z.B. `ETASData\Ascet5.2`, abgelegt. Sie können Standardnamen und -pfad der Protokolldatei ändern.

#### Ändern der Protokolldatei:

---



- Klicken Sie auf die Schaltfläche **Change File-name**

oder

- wählen Sie **Options** → **Change Filename**.

Es erscheint das Dateiauswahlfenster.

- Stellen Sie den neuen Pfad und Dateinamen ein.
- Klicken Sie auf **Öffnen**, um die neuen Einstellungen zu übernehmen.

Der Dateiname im unteren Bereich des Fensters „Data Logger“ wird geändert, und die Daten werden nun in der neuen Datei gespeichert.

Wenn Sie die entsprechende Option im ASCET-Optionsfenster aktiviert haben (s. Seite 60), wird der von Ihnen gewählte Dateiname um eine ganze Zahl erweitert, die bei jedem Abspeichern hochgezählt wird.

## Umwandeln von MDF-Daten in FAMOS-Format:

---

- Wählen Sie **Extras** → **Convert MDF to FAMOS**.  
Es wird das Dateiauswahlfenster für die Eingabedatei geöffnet.
- Wählen Sie die umzuwandelnde MDF-Datei.  
Es wird das Dateiauswahlfenster für die FAMOS- Ausgabedatei geöffnet.
- Wählen Sie einen Pfad- und Dateinamen für die FAMOS-Datei.  
Die MDF-Eingabedatei wird gelesen, in FAMOS umgewandelt und in die gewählte Ausgabedatei geschrieben.

### 6.1.10 Experimentierumgebungs-Optionen

---

#### *Navigieren zwischen Komponenten*

---

Enthält ein Projekt oder eine Komponente andere Komponenten, kann man sich diese Komponenten ohne Verlassen der Experimentierumgebung ansehen.

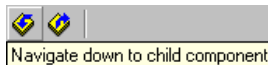
#### **Navigieren zwischen Blockdiagrammen:**

---

##### 1. Abwärts navigieren

- Wählen Sie in der Fläche „Elements“ die referenzierte Komponente, deren Blockdiagramm Sie sich ansehen möchten.

• Wählen Sie **Elements** → **Show Component**  
*oder*



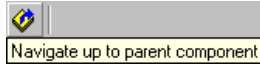
- klicken Sie auf die Schaltfläche **Navigate down to child component**

*oder*

- doppelklicken Sie in der Komponentenanzeige auf das Vorkommen einer aufgenommenen Komponente.

Die gewählte Komponente wird in der Experimentierumgebung dargestellt.

##### 2. Aufwärts navigieren



- Wählen Sie **View** → **Show Parent Component**

oder

- klicken Sie auf die Schaltfläche **Navigate up to parent component**

oder

- doppelklicken Sie auf eine leere Stelle in der Komponentenanzeige.

#### Hinweis

*Aufwärts und abwärts navigieren mit Doppelklick funktioniert nur in Blockdiagrammen und Zustandsautomaten.*

Die übergeordnete Komponente wird angezeigt.

### Schalten zwischen den Diagrammen einer Komponente:

- Wählen Sie in der Fläche „Diagrams“ der Experimentierumgebung das Diagramm, zu dem Sie umschalten möchten.
- Wählen Sie **Diagrams** → **Show Diagram**.  
Es wird das gewählte Diagramm angezeigt.

### Anzeigeoptionen

#### Ändern der Blockdiagrammanzeige:

- Wählen Sie **View** → **Show Seq. Calls** zum Anzeigen der Sequenzaufrufe des aktuellen Blockdiagramms.  
Standardmäßig werden die Sequenzaufrufe nicht angezeigt.
- Wählen Sie **View** → **Hide Seq. Calls** zum Verbergen der Sequenzaufrufe.
- Wählen Sie **View** → **Redraw** um das Blockdiagramm neu zu zeichnen.

Mit den vier Schaltflächen über der Fläche „Elements“ können Sie mehrere verschiedene Ansichten der Elemente der Komponente wählen.



## Ändern der Elementdarstellung:

---



- Klicken Sie auf die Schaltfläche **Elements Hierarchical** zum Umschalten von einer flachen Listenanzeige zur standardmäßigen hierarchischen Ansicht von Elementen.



- Klicken Sie auf die Schaltfläche **Elements Flat** zum Umschalten auf eine flache Listenanzeige.



- Klicken Sie auf die Schaltfläche **Exported Elements**, um sich nur exportierte Elemente (d.h. Messages und globale Elemente) anzeigen zu lassen.



- Klicken Sie auf die Schaltfläche **Parameter Elements**, um sich nur Parameterelemente anzeigen zu lassen.

### 6.1.11 Manipulieren von Daten

---

Während eines Experiments werden die Werte der Elemente gewöhnlich verändert, um beispielsweise die richtige Parametereinstellung für eine bestimmte Funktion zu finden. Wurde die richtigen Werte gefunden, können sie im aktuellen Datensatz der Komponente gespeichert werden. Details dazu siehe Kapitel „Datensätze“ auf Seite 503.

#### Lesen oder Schreiben von Daten aus dem aktuellen Datensatz:

---

- Wählen Sie in der Fläche „Elements“ das Element, dessen Wert Sie wegschreiben möchten.
- Wählen Sie **Elements** → **Write Back Data** → **Selected Elements** oder **Calibrated Elements**.

Es wird eine Liste von Variablen angezeigt. Wurde der Befehl **Selected Elements** gewählt, werden alle aktuell in der Fläche „Elements“ gewählten Elemente in der Liste angezeigt. Wurde **Calibrated Elements** gewählt, werden alle im Verlauf des Experiments verstellten Elemente angezeigt. Zunächst sind alle Elemente in der Liste ausgewählt.

- Wählen Sie die Variablen ab, die Sie nicht wegschreiben möchten, und klicken Sie auf **OK**.

Die Werte der gewählten Elemente werden in den aktuellen Datensatz der Komponente geschrieben. Die alten Werte des betreffenden Datensatzes werden überschrieben. Der aktuelle Datensatz ist der beim Starten des Experiments gewählte Datensatz.

- Wählen Sie **Elements** → **Reinitialize** → **Variables** oder **Parameters** oder **Both** zum Lesen der Daten aus dem aktuellen Datensatz.

Allen Variablen und/oder Parameter werden die Werte aus dem aktuellen Datensatz zugewiesen; die Werte im Experiment werden überschrieben. Dieser Befehl steht bei laufendem Experiment nicht zur Verfügung.

Sie können die Werte von Parametern auch in externe Dateien speichern bzw. aus externen Dateien einlesen.

#### **Hinweis**

*Wenn Sie mit dem Simulationsrechner ES1135 experimentieren, können Sie auch die Daten von nicht-flüchtigen Variablen in eine Datei schreiben.*

#### **Einstellen der Datenaustauschoptionen:**

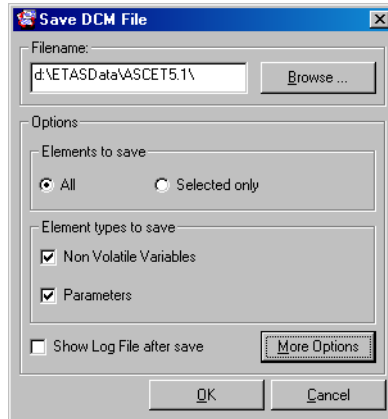
- Öffnen Sie im Komponentenmanager das ASCET-Optionsfenster.
- Stellen Sie im Knoten „Data Exchange“ die gewünschten Optionen ein.

Die Bedeutung der Optionen ist im Abschnitt „Optionen für den Datenaustausch“ auf Seite 68 beschrieben.

#### **Schreiben von Daten in externe Dateien:**

- Wählen Sie in der Fläche „Elements“ die Elemente, deren Werte Sie in eine Datei schreiben möchten.

- Wählen Sie **Elements** → **Save Data**.  
Das Fenster „Save DCM File“ öffnet sich.



- Machen Sie die erforderlichen Angaben.
- Klicken Sie **OK**.

Die Werte der Elemente werden in die angegebene Datei geschrieben. Gegebenenfalls wird anschließend die Log-Datei für Schreibvorgänge angezeigt.

Das Fenster „Save DCM File“ bietet folgende Einstellmöglichkeiten:

### Hinweis

Der Dateiname und mindestens eine der Optionen **Non Volatile Variables** und **Parameters** **müssen** explizit angegeben werden, damit der Vorgang beendet werden kann.

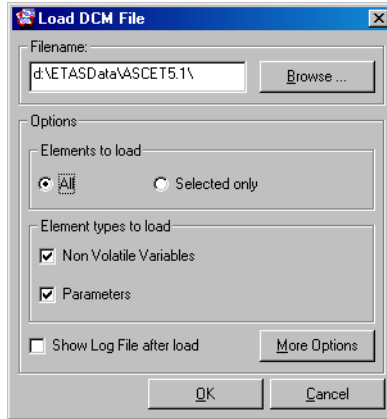
- Im Feld „Filename“ geben Sie Pfad und Name der Datei direkt oder über die Schaltfläche **Browse** ein.  
Als Dateiformat wird die Auswahl im Knoten „Data Exchange“ des Optionsfensters (s. Seite 68) verwendet.
- Mit den Optionen **All** und **Selected only** legen Sie fest, ob die Daten aller oder ausgewählter Elemente gesichert werden.
- Mit den Optionen **Non Volatile Variables** und **Parameters** wählen Sie die Elementtypen, deren Daten gespeichert werden.
- Mit der Option **Show Log File after save** legen Sie fest, ob nach dem Schreibvorgang die Log-Datei angezeigt wird.

- Mit der Schaltfläche **More Options** öffnen Sie das ASCET-Optionsfenster im Knoten „Data Exchange“; die Optionen sind im Abschnitt „Optionen für den Datenaustausch“ auf Seite 68 beschrieben.

### Lesen von Daten aus einer externen Dateien:

---

- Wählen Sie **Elements** → **Load Data**.  
Das Fenster „Load DCM File“ öffnet sich.



Die Einstellmöglichkeiten sind dieselben wie im Fenster „Save DCM File“ (s. Seite 659). Die Optionen **All** und **Selected only** haben beim Einlesen jedoch keine Bedeutung.

- Machen Sie die erforderlichen Angaben.
- Klicken Sie auf **OK**.

Die Werte aus der angegebenen Datei werden gelesen. Gegebenenfalls wird anschließend die Log-Datei für Lesevorgänge angezeigt.

## 6.2 Verstellfenster

---

Mit dem Verstellsystem können Sie die Werte der Basiselemente der Komponente, mit der Sie experimentieren, verändern. Sie können die Werte beim Einrichten eines Experiments oder bei laufendem Experiment ändern. Es ist auch möglich, Elementen während der Spezifikation Anfangswerte zuzuweisen, jedoch haben im Verstellsystem geänderte Werte Vorrang gegenüber solchen standardmäßigen Einstellungen.

Weisen Sie einem Element mit dem Verstellsystem einen Wert zu, bleibt dieser Wert erhalten, bis er durch eine Berechnung innerhalb der Komponente geändert oder durch einen Wert aus dem Datengenerator überschrieben wird. Die Dateneditoren sind die gleichen, die auch beim Spezifizieren von Komponenten zum Einsatz kommen. Sie sind im Kapitel „Bearbeiten von Daten“ auf Seite 493 beschrieben.

Beim Spezifizieren einer Komponente können Sie jedem Element in Ihrer Spezifikation einen Ausgangswert zuordnen. Diese Werte lassen sich – außer für Konstanten – alle später verändern. Sie können verschiedene Datensätze spezifizieren, d.h. Sätze von Initialisierungswerten, zwischen denen Sie hin- und herschalten können, oder Sie können einzelne Werte während der Experimente ändern. Dieser Abschnitt beschreibt die verschiedenen Editoren für die verschiedenen Arten von Elementen.

Gewöhnlich wird ein Dateneditor zuerst aus der Umgebung der Komponentenentwicklung heraus aufgerufen, wie z.B. dem Blockdiagrammeditor, um einem Element einen Standardwert zuzuordnen. Danach kann der Editor erneut aus der Experimentierumgebung heraus geöffnet werden, um den Wert des Elements während eines Experiments zu verstellen. Dateneditoren können ebenfalls zum Festlegen von Datensätzen für Komponenten oder Projekte verwendet werden. Dateneditoren funktionieren immer gleich, unabhängig davon, aus welchem ASCET Teil sie geöffnet wurden.

Wie Sie aus der Offline-Experimentierumgebung einen Dateneditor öffnen, ist im Abschnitt „Das Verstellsystem“ auf Seite 627 beschrieben.

### 6.2.1 Allgemeine Beschreibung der Menüfunktionen

Nicht alle Menüfunktionen sind in jedem Editor verfügbar.

- **Edit:**

#### Hinweis

Das Menü **Edit** ist im 3D-Grafikeditor nicht vorhanden.

- *Undo Last Change* (<CTRL> + <U>)  
Letzte Änderung rückgängig machen.
- *Redo Last Change* (<CTRL> + <D>)  
Letzte Änderung wiederherstellen.
- *Copy* (<CTRL> + <C>)  
Verstellgröße in Zwischenablage kopieren.
- *Paste* (<CTRL> + <V>)  
Verstellgröße aus Zwischenablage in Fenster einfügen.

- *Copy Entire Data Into Clipboard*  
Die im Tabelleneditor aktive Verstellgröße (Kennlinie bzw. -feld) wird in die Zwischenablage kopiert und kann dann von dort in andere Programme (z.B. Excel) eingefügt und weiterverarbeitet werden.
- *Select All Values(<CTRL> + <A>)*  
Alle Werte auswählen.
- *Block Selection (<CTRL> + <B>)*  
Ermöglicht die Auswahl mehrerer Werte auf einer Kennlinie.
- *Decrement bzw. Decrement Value (<CTRL> + <N>)*  
Verkleinern um den voreingestellten Wert.
- *Increment bzw. Increment Value (<CTRL> + <M>)*  
Vergrößern um den voreingestellten Wert.
- *Add Offset*  
Addieren eines oder mehrerer Werte.
- *Multiply By Factor*  
Multiplizieren eines oder mehrerer Werte.
- *Fill With Values*  
Ersetzen eines oder mehrerer Werte durch einen anderen.
- *Decrement X Axis Point (<CTRL> + <J>)*  
x-Achsenpunkt nach links rücken.
- *Decrement Arg (<CTRL> + <J>)*  
Achsenpunkt im 2D-Grafikeditor nach links rücken.
- *Increment X Axis Point (<CTRL> + <K>)*  
x-Achsenpunkt nach rechts rücken.
- *Increment Arg (<CTRL> + <K>)*  
Achsenpunkt im 2D-Grafikeditor nach rechts rücken.
- *Add X Axis Point*  
x-Achsenpunkt hinzufügen.
- *Remove X Axis Point*  
x-Achsenpunkt entfernen.
- *File In Data*  
Lesen der Daten eines Array oder einer Tabelle aus einer Datei.
- *File Out Data*  
Schreiben der Daten eines Array oder einer Tabelle in eine Datei.

- **Axis:**

### Hinweis

Das Menü **Axis** ist nur vorhanden, wenn Kennlinien und -felder im Tabelleneditor bearbeitet werden.

- *X Supporting Points Setup*  
x-Stützstellen mit Werten konstanten Abstands belegen.
  - *Decrement X Axis Point (<CTRL> + <J>)*  
x-Achsenpunkt verkleinern.
  - *Increment X Axis Point (<CTRL> + <K>)*  
x-Achsenpunkt vergrößern.
  - *Edit X Axis Point (<CTRL> + <X>)*  
x-Achsenpunkt einen konkreten Wert zuweisen.
  - *Add X Axis Point*  
x-Achsenpunkt hinzufügen.
  - *Remove X Axis Point*  
x-Achsenpunkt entfernen.
  - *Y Supporting Points Setup*  
y-Stützstellen mit Werten konstanten Abstands belegen.
  - *Decrement Y Axis Point (<CTRL> + <R>)*  
y-Achsenpunkt verkleinern.
  - *Increment Y Axis Point (<CTRL> + <T>)*  
y-Achsenpunkt vergrößern.
  - *Edit Y Axis Point (<CTRL> + <Y>)*  
y-Achsenpunkt einen konkreten Wert zuweisen.
  - *Add Y Axis Point*  
y-Achsenpunkt hinzufügen.
  - *Remove Y Axis Point*  
y-Achsenpunkt entfernen.
- **View:**
    - *Larger Font*  
Anzeigen des verstellten Werts in größerem Schriftgrad. Wenn der Befehl erneut angewählt wird, gelangt man zur ursprünglichen Größe zurück.
    - *Reset Change Marks*  
Änderungsmarkierungen zurücksetzen.

- *Show Process Point*  
Arbeitspunkt markieren.
  - *Set Editor on Process Point* (<CTRL> + <W>)  
Wählt den dem aktuellen Wert nächsten Abtastpunkt einer Tabelle zur Bearbeitung aus.
  - *Grid*  
Anzeigegitter ein- / ausblenden.
  - *xz-Viewpoint*  
xz-Darstellung
  - *yz-Viewpoint*  
yz-Darstellung
  - *Show Key Help*  
Blendet die wichtigsten Tastaturbefehle in der Fußzeile ein bzw. aus.
- **Extras:**
    - *Change Title*  
Verändern des Namens für das markierte Verstellfenster.
    - *Optimize Size*  
Optimiert die Größe des Dialogfensters.
    - *Display Setup* (<CTRL> + <S>)  
Öffnet das Setup-Fenster (das Setup-Dialogfenster ist abhängig vom gewählten Editortyp).
    - *Colors*  
Verändert die Farbeinstellung für den 1D-Grafikeditor:  
*Black & white* → Schwarzweiß-Darstellung,  
*Default colors* → Standardfarben,  
*Invert colors* → invertiert die aktuellen Farben.
    - *Physical Representation* (<CTRL> + <P>)  
Stellt die markierte(n) Verstellgröße(n) als physikalischen Wert dar.
    - *Hexadec. Representation* (<CTRL> + <H>)  
Stellt die markierte(n) Verstellgröße(n) hexadezimal dar.
    - *Decimal Representation* (<CTRL> + <Z>)  
Stellt die markierte(n) Verstellgröße(n) dezimal dar.
    - *Binary Representation* (<CTRL> + <R>)  
Stellt die markierte(n) Verstellgröße(n) binär dar.



- *Move Variable Into Window...*  
Verschiebt die markierte Verstellgröße in einen anderen Dateneditor. Jede Variable kann sich gleichzeitig nur einmal in einem Dateneditor befinden.
- *Remove Variable*  
Entfernt die markierte Verstellgröße aus dem Dateneditor.
- *About Variable* (<CTRL> + <I>)  
Öffnet ein Fenster mit Informationen zu der markierten Verstellgröße.
- *Move* (nur bei mehreren Verstellgrößen in einem Fenster)  
*Up* → verschiebt die markierte Verstellgröße eine Position nach oben  
*Down* → verschiebt die markierte Verstellgröße eine Position nach unten.

### 6.2.2 Dateneditoren für Verstellgrößen

---

Für alle Editoren gilt gleichermaßen, dass die Verstellung direkt in der Darstellung erfolgt. Bei den numerischen Editoren und den Tabelleneditoren wird die Verstellung durch Ändern des Zahlenwertes vorgenommen. Kennlinien und -felder können auch grafisch durch Verschieben der Stützpunkte intuitiv verändert werden.

Wenn Sie eine Verstellung vorgenommen haben, erscheint neben dem verstellten Wert ein roter Pfeil, der nach oben oder nach unten zeigt. Dies bedeutet, dass der Wert der jeweiligen Verstellgröße nach oben bzw. nach unten verändert wurde. Dies gilt für alle veränderten Werte, unabhängig davon, mit welchem Editor Sie arbeiten.

### 6.2.3 Arbeiten mit Verstellfenstern

---

Jedes Verstellfenster kann verschiedene Variablen anzeigen, wobei es möglich ist, die Variablen zwischen den Fenstern zu verschieben. Beim Verschieben einer Variablen wird diese in einem Fenster gelöscht und einem anderen Fenster hinzugefügt. Jede Variable kann nur in einem Editor auftauchen.

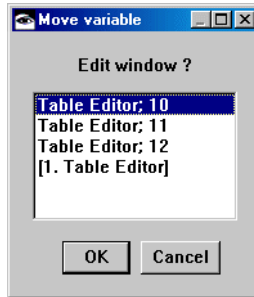
#### **Verschieben von Variablen zwischen Verstellfenstern:**

---

- Wählen Sie in einem Verstellfenster eine oder mehrere Variablen aus, die Sie verschieben wollen.

- Wählen Sie **Extras** → **Move Variable Into Window**.

Das Dialogfenster „Move variable“ wird geöffnet. Es enthält eine Liste aller zum Datentyp passenden Verstellfenster.



- Wählen Sie das Fenster aus, in das Sie die markierten Variablen verschieben wollen.
- Klicken Sie auf **OK**.

Die markierten Variablen werden in das ausgewählte Fenster (neu oder vorhanden) verschoben und aus dem ursprünglichen Fenster entfernt. Wenn das alte Fenster dadurch leer wird, wird es geschlossen.

### **Variablen aus einem Verstellfenster löschen:**

---

Um Variablen aus einem Verstellfenster zu löschen, gehen Sie wie folgt vor:

- Wählen Sie eine oder mehrere Variablen aus, die Sie löschen wollen, indem Sie auf das entsprechende Feld klicken.
- Wählen Sie **Extras** → **Remove Variable**.

Die markierten Variablen werden aus dem Fenster entfernt. Wenn das Fenster dadurch leer wird, wird es geschlossen.

### **Titel eines Verstellfensters ändern:**

---

Um den Titel eines Verstellfensters zu ändern, gehen Sie wie folgt vor:

- Wählen Sie in dem Verstellfenster, dessen Titel Sie ändern wollen, **Extras** → **Change Title**.

- Geben Sie im Eingabefenster den neuen Titel ein.
- Klicken Sie **OK**.  
Der neue Titel erscheint in der Kopfzeile des Verstellfensters.

### Informationen über Variable anzeigen:

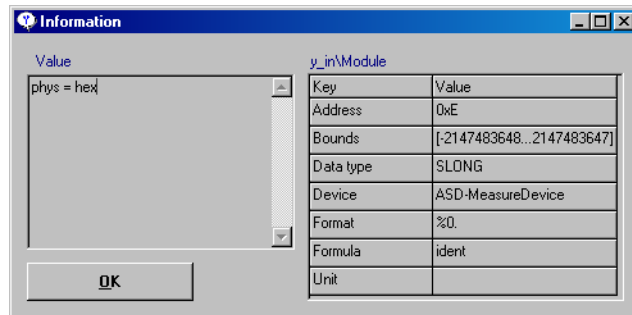
Um Informationen über eine oder mehrere ausgewählte Variablen anzuzeigen, gehen Sie wie folgt vor:

- Wählen Sie in einem Verstellfenster eine oder mehrere Variablen aus.
- Wählen Sie in diesem Fenster **Extras** → **About Variable**

oder

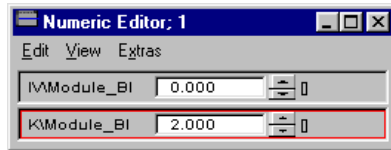
- drücken Sie <CTRL> + <I>.

Für jede ausgewählte Variable wird ein Fenster „Information“ geöffnet, das Informationen über die jeweilige Variable enthält.



- Klicken Sie auf **OK**, um das Fenster „Information“ zu schließen.

Im folgenden werden die verschiedenen Verstellfenster beschrieben.



**Einrichten eines numerischen Editors:**

---

- Wählen Sie **Extras** → **Physical Representation**

*oder*

- drücken Sie <CTRL> + <P> , um sich die Werte als physikalische Größen anzeigen zu lassen.
- Wählen Sie **Extras** → **Hexadec. Representation**

*oder*

- drücken Sie <CTRL> + <H>, um sich die Werte im Hexadezimalformat anzeigen zu lassen.
- Wählen Sie **Extras** → **Binary Representation**

*oder*

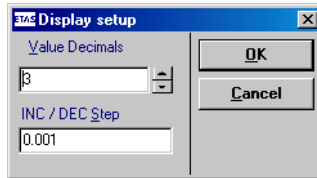
- drücken Sie <CTRL> + <R>, um sich die Werte im Binärformat anzeigen zu lassen.
- Wählen Sie **Extras** → **Decimal Representation**

*oder*

- drücken Sie <CTRL> + <Z>, um sich die Werte ganzzahlig anzeigen zu lassen.
- Wählen Sie **Extras** → **Display Setup**

*oder*

- drücken Sie <CTRL> + <S>.  
Es erscheint das Dialogfenster „Display setup“.



- Passen Sie die Anzahl der Dezimalstellen an.  
Damit wird die Anzahl der Dezimalstellen festgelegt, die für den Wert angezeigt werden sollen.
- Passen Sie den Wert im Feld „INC/DEC Step“ an.  
Dieser Wert bestimmt die Schrittgröße beim Inkrementieren bzw. Dekrementieren.
- Klicken Sie auf **OK**.

### **Bearbeiten eines numerischen Wertes:**

---

- Klicken Sie auf die numerische Anzeige im Fenster.
- Bearbeiten Sie den Wert und drücken Sie <ENTER>, um Ihre Änderung zu bestätigen.  
Sie können den Wert auch mit den Pfeiltasten rechts von der Anzeige des Wertes anpassen. Damit wird der Wert um die im Setup eingestellte Schrittgröße inkrementiert oder dekrementiert. Der neue Wert erscheint sofort.
- Wählen Sie **Edit** → **Undo Last Change**  
oder
- drücken Sie <CTRL> + <U>, um zum Wert vor der letzten Änderung zurückzukehren.
- Wählen Sie **Edit** → **Redo Last Change**  
oder

- drücken Sie <CTRL> + <D>, um das Rückgängigmachen aufzuheben.

### Hinweis

*Die letzten 10 Änderungen jeder Größe werden gespeichert.*

## Bearbeiten mehrerer numerischer Werte:

Wenn Sie mehrere Größen in einem numerischen Editorfenster bearbeiten, können Sie diese auch gemeinsam verändern.

- Markieren Sie eine oder mehrere Größen, die Sie verändern wollen.
- Wählen Sie **Edit** → **Increment**

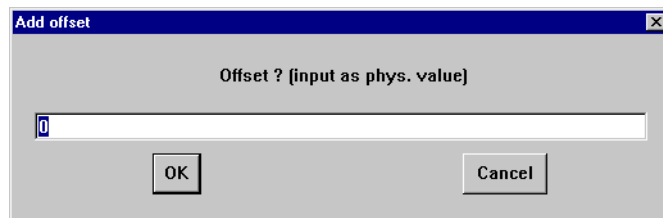
*oder*

- drücken Sie <CTRL> + <M>, um alle markierten Werte um die im Setup eingestellte Schrittgröße zu inkrementieren.
- Wählen Sie **Edit** → **Decrement**

*oder*

- drücken Sie <CTRL> + <N>, um alle markierten Werte um die im Setup eingestellte Schrittgröße zu dekrementieren.
- Wählen Sie **Edit** → **Add Offset**, um allen markierten Werten eine Zahl hinzuzufügen.

Sie werden zuerst nach einem Offsetwert gefragt, der danach zum Wert in der Anzeige addiert wird.



- Wählen Sie **Edit** → **Multiply By Factor**, um alle markierten Werte mit einem Faktor zu multiplizieren.  
Sie werden zuerst nach einem Faktor gefragt, worauf die Multiplikation mit diesem Faktor erfolgt.
- Wählen Sie **Edit** → **Fill With Values**, um alle markierten Werte zu überschreiben.  
Sie werden zuerst nach einer Konstanten gefragt; danach wird der Wert durch diese Konstante ersetzt.

#### Hinweis

*Wenn Sie mehr als einen Wert ausgewählt haben, können Sie die Änderungen **nicht** gemeinsam rückgängig machen.*

Es ist möglich, mit anderen Anwendungen über die Zwischenablage von Windows Daten auszutauschen. Das können andere Verstellfenster, aber auch andere Programme wie Tabellenkalkulationsprogramme oder Datenbanken sein.

#### **Austauschen von Daten mit anderen Anwendungen:**

---

- Markieren Sie die Variable, deren Wert Sie in die Zwischenablage kopieren wollen.

#### Hinweis

*Auch wenn ein numerischer Editor mehrere Variablen enthält, kann nur eine Variable zur Zeit kopiert werden.*

- Wählen Sie **Edit** → **Copy**  
oder
- drücken Sie <CTRL> + <C>, um den ausgewählten Wert in die Zwischenablage zu kopieren.
- Um den Wert in eine andere Variable in einem numerischen Editor zu kopieren, markieren Sie diese Variable.

- Um den Wert in eine Zelle einer Tabelle einzufügen, markieren Sie die Zellen, in die der Wert eingefügt werden soll.

#### **Hinweis**

*Auch wenn Sie mehrere Zellen markieren, wird der Wert nur in die erste kopiert.*

- Wählen Sie dann **Edit** → **Paste**

*oder*

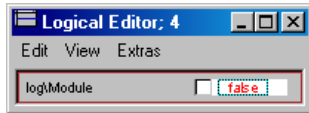
- drücken Sie <CTRL> + <V>.

Die Daten aus der Zwischenablage werden in die Variable bzw. die markierte Zelle geschrieben.

- Sie können die Daten auch in jede andere Anwendung einfügen.

### *Logischer Editor*

---



Die Menüfunktionen entsprechen den gleichnamigen Menüfunktionen des numerischen Editors.

#### **Bearbeiten eines logischen Wertes:**

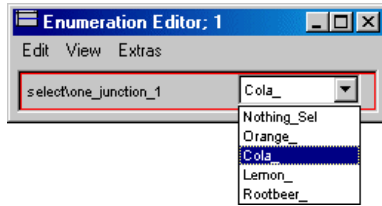
---

- Aktivieren Sie die Option, um den Wert auf `true` zu setzen.
- Deaktivieren Sie die Option, um den Wert auf `false` zu setzen.



## Enumeration-Editor

---



Die Menüfunktionen entsprechen den gleichnamigen Menüfunktionen des numerischen Editors.

### Bearbeiten einer Enumeration:

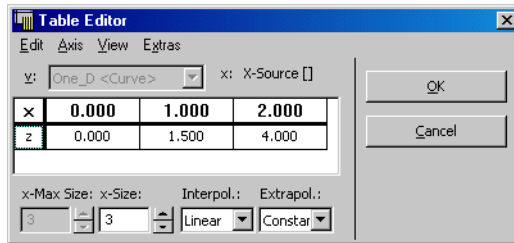
---

Das Kombikästchen enthält alle zur Verfügung stehenden Enumeratoren.

- Wählen Sie aus dem Kombikästchen einen Enumerator.

## Array-Editor

---



Sie können gleichzeitig mehrere Arrays in einem einzigen Tabelleneditor-Fenster öffnen, während Sie mit einer Komponente experimentieren. In diesem Fall werden die aktuell geöffneten Arrays im Kombikästchen „v:“ angezeigt, und Sie können zwischen ihnen durch Auswahl der entsprechenden Arrays aus diesem Kombikästchen umschalten. Tabellen unterschiedlicher Art sind ebenfalls möglich, wie z.B. im gleichen Editorfenster geöffnete Arrays und eindimensionale Tabellen.

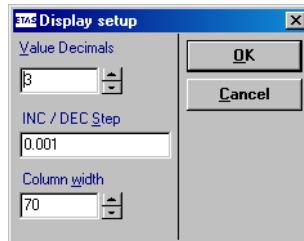
### Hinweis

Zweidimensionale Arrays sind in der Form von Matrizen verfügbar.

## Array-Editor einrichten:

---

- Wählen Sie **Extras** → **Display Setup**  
oder
- drücken Sie <CTRL> + <S>.  
Das Dialogfenster „Display setup“ wird geöffnet.



- Passen Sie die Anzahl der Dezimalstellen an, mit denen Zahlen im Array-Editor angezeigt werden sollen.
- Passen Sie die Schrittgröße zum Inkrementieren und Dekrementieren an.
- Passen Sie die Spaltenbreite des Array-Editors an.
- Klicken Sie auf **OK**, um das Dialogfenster „Display setup numeric“ zu schließen.

## Bearbeiten von einzelnen Ausgabewerten:

---

- Klicken Sie auf den zu bearbeitenden Ausgabewert.  
Der Ausgabewert wird markiert und kann direkt bearbeitet werden.
- Geben Sie den neuen Ausgabewert in die Zelle der Tabelle ein.
- Drücken Sie <ENTER>.  
Der eingegebene Wert wird übernommen und mit einem Änderungspfeil markiert.
- Wählen Sie **Edit** → **Undo Last Change**, um die letzte Änderung rückgängig zu machen.

- Wählen Sie **Edit** → **Redo Last Change**, um das Rückgängigmachen aufzuheben.
- Wählen Sie **View** → **Reset Change Marks**, um die Pfeile auszublenden.

### **Bearbeiten von mehreren Ausgabewerten:**

---

- Markieren Sie die zu bearbeitenden Ausgabewerte der z-Achse, indem Sie den Cursor über den Ausgabewert ziehen.

- Wählen Sie **Edit** → **Select All Values**

*oder*

- drücken Sie <CTRL> + <A>, um alle Ausgabewerte auf der z-Achse zu markieren.

- Wählen Sie **Edit** → **Increment**

*oder*

- drücken Sie <CTRL> + <M>, um alle markierten Werte um die voreingestellte Schrittgröße zu inkrementieren.

- Wählen Sie **Edit** → **Decrement**

*oder*

- drücken Sie <CTRL> + <N>, um alle markierten Werte um die voreingestellte Schrittgröße zu dekrementieren.

- Wählen Sie **Edit** → **Add Offset**, um zu allen markierten Werten eine Zahl zu addieren.

Wenn Sie diesen Befehl auswählen, werden Sie nach einem Wert gefragt, der dann zu allen markierten Ausgabewerten addiert wird.

- Wählen Sie **Edit** → **Multiply By Factor**, um alle markierten Werte mit einem Faktor zu multiplizieren.

Wenn Sie diesen Befehl auswählen, werden Sie nach einem Faktor gefragt, worauf die Multiplikation mit diesem Faktor erfolgt.

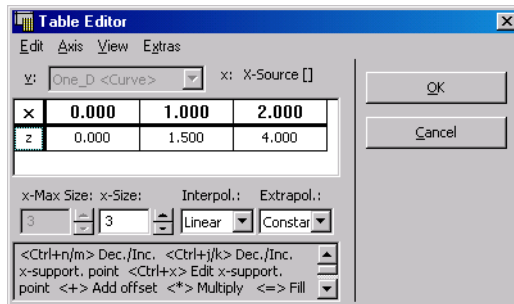
- Wählen Sie **Edit** → **Fill With Value**, um alle markierten Werte zu überschreiben.

Wenn Sie diesen Befehl auswählen, werden Sie nach einer Konstanten gefragt; dann werden alle markierten Ausgabewerte durch diese Konstante ersetzt.

## Ändern der Anzeige:

---

- Wählen Sie **View** → **Show Key Help**, um sich die Tastenkurbefehle für den Array-Editor unten im Fenster anzeigen zu lassen.



- Wählen Sie **Extras** → **Optimize Size**, um automatisch die Fenstergröße beim Umschalten zwischen den Tabellen anzupassen.

Dieser Befehl wird standardmäßig aktiviert.

Es ist möglich, mit anderen Anwendungen über die Zwischenablage von Windows Daten auszutauschen. Das können andere Fenster des Array- oder Tabelleneditors, aber auch andere Programme wie Tabellenkalkulationsprogramme oder Datenbanken sein.

## Austauschen von Daten mit anderen Anwendungen:

---

- Markieren Sie die Werte, die Sie in die Zwischenablage kopieren wollen.

Oder

- Wählen Sie **Edit** → **Select All Values**, um alle Werte zu markieren.

- Wählen Sie **Edit** → **Copy** aus, um die ausgewählten Werte in die Zwischenablage zu kopieren.

### Hinweis

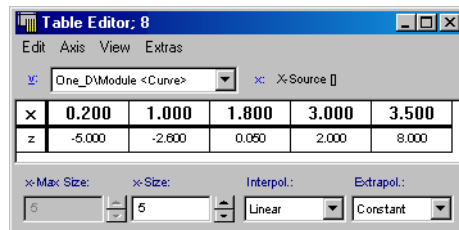
*Nicht die Abtastpunkte, sondern nur die tatsächlichen Datenwerte werden kopiert.*

- Wählen Sie **Edit** → **Copy Entire Data Into Clipboard**, um den gesamten Array in die Zwischenablage zu kopieren.
- Um die Werte in einen anderen Tabelleneditor einzufügen, markieren Sie den Bereich von Zellen, in die sie eingefügt werden sollen (in der Tabelle muss die entsprechende Anzahl von Zellen vorhanden sein).
- Wählen Sie dann **Edit** → **Paste**.

Die Daten aus der Zwischenablage werden in die markierten Zellen geschrieben.

Sie können auch die Daten in jede andere Anwendung einfügen. Die Daten werden in einem tab-abgegrenzten Format in der Standardzwischenablage von Windows gespeichert.

### *1D-Tabelleneditor*



Verschiedene Tabellen können im gleichen Fenster des Tabelleneditors bearbeitet werden, wenn Sie zum Beispiel in der Experimentierumgebung verschiedene Tabellen zum Verstellen ausgewählt haben. In diesem Fall werden alle verfügbaren Tabellen im Kombikästchen „v:“ angezeigt; Sie können zwischen ihnen umschalten, indem Sie sie aus diesem Kombikästchen auswählen.

Der 1D-Tabelleneditor funktioniert auf die gleiche Weise wie der Array-Editor. Der einzige Unterschied zwischen den Editoren besteht darin, dass im Tabelleneditor die Werte auf beiden Achsen bearbeitet werden können. Dieser Abschnitt beschreibt nur die Unterschiede zwischen den beiden Editoren; weitere Informationen sind im vorangegangenen Abschnitt enthalten.

### **Tabelleneditor einrichten:**

---

- Sie können den Tabelleneditor auf die gleiche Weise einrichten wie den Array-Editor; siehe „Array-Editor einrichten:“ auf Seite 674

### **Verwenden der Abtastpunktanzeige**

---

Bei Tabelleneditoren für Kennlinien und -felder haben Sie die Möglichkeit, sich während eines Messvorgangs den Abtastpunkt anzeigen zu lassen.

- Wählen Sie **View** → **Show Process Point**, um die Abtastpunktanzeige zu aktivieren.  
Bei der nächsten Messung werden die den aktuellen Wert einschließenden Abtastpunkte in der Tabelle hervorgehoben (rote Umrandung).
- Wählen Sie **View** → **Set Editor on Process Point**  
*oder*  
• drücken Sie <CTRL> + <W>, um den dem aktuellen Wert nächsten Abtastpunkt zur Bearbeitung auszuwählen.

### **Bearbeiten vorhandener Abtastpunkte:**

---

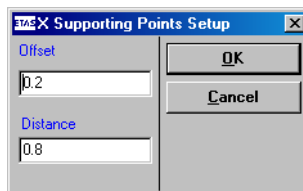
- Klicken Sie auf den zu ändernden Abtastpunkt.  
Es erscheint ein Eingabefeld, das den aktuellen x-Wert des Abtastpunktes enthält.
- Geben Sie den neuen Wert ein.  
Der Wert muss zwischen dem vorangegangenen und dem nachfolgenden Abtastpunkt liegen. Ist dies nicht der Fall, wird die Änderung nicht durchgeführt.

- Klicken Sie auf **OK**.  
Wenn Sie einen Wert geändert haben, erscheint links davon ein roter Pfeil.
- Wählen Sie **Axis** → **Increment X Axis Point**  
*oder*
- drücken Sie <CTRL> + <K>, um den Wert des ausgewählten Achsenpunkts zu inkrementieren.
- Wählen Sie **Axis** → **Decrement X Axis Point**  
*oder*
- drücken Sie <CTRL> + <J>, um den Wert des ausgewählten Achsenpunkts zu dekrementieren.
- Wählen Sie **View** → **Reset Change Marks**, um die Pfeile auszublenden.
- Wählen Sie **Edit** → **Undo Last Change**, um die Änderung der letzten von Ihnen vorgenommenen Operation rückgängig zu machen.
- Wählen Sie **Edit** → **Redo Last Change**, um das Rückgängigmachen aufzuheben.

### **Gleichzeitiges Bearbeiten aller Abtastpunkte:**

---

- Wählen Sie **Axis** → **X Supporting Points Setup**.  
Das Dialogfenster „X Supporting Points Setup“ wird geöffnet.



- Geben Sie einen Offsetwert und einen Abstand in die entsprechenden Felder ein.

- Klicken Sie auf **OK** zum Bestätigen Ihrer Änderungen.

Die Punkte auf der x-Achse der gesamten Tabelle werden wie festgelegt geändert. Der erste Punkt erhält den als Offset festgelegten Wert, alle anderen Werte werden um den festgelegten Abstand erhöht.

### **Einfügen/Entfernen von neuen Abtastpunkten:**

---

- Wählen Sie **Axis** → **Add X Axis Point**, um einen neuen Abtastpunkt hinzuzufügen.

#### **Hinweis**

*Wenn die Tabelle bereits die maximale Größe hat, können Sie keine weiteren Abtastpunkte hinzufügen.*

Es erscheint ein Eingabefeld.

- Geben Sie den x-Wert für den Abtastpunkt ein.
- Klicken Sie auf **OK**.

Der neue Abtastpunkt wird an der richtigen Stelle innerhalb der Tabelle eingefügt. Er nimmt den Wert des Abtastpunktes zu seiner Linken an.

- Wählen Sie **Axis** → **Remove X Axis Point**, um den ausgewählten Achsenpunkt zu entfernen.

### **Bearbeiten von einzelnen Ausgabewerten:**

---

- Klicken Sie auf den zu bearbeitenden Wert.  
Der Wert wird markiert und kann direkt bearbeitet werden.
- Geben Sie den neuen Wert in die Zelle der Tabelle ein.
- Drücken Sie <ENTER>.

### **Bearbeiten von mehreren Ausgabewerten:**

---

- Markieren Sie die zu bearbeitenden Werte, indem Sie den Cursor darüber ziehen.

*Oder*



- Wählen Sie **Edit** → **Select All Values**, um alle Ausgabewerte auf der z-Achse zu markieren.

- Wählen Sie **Edit** → **Increment**

oder

- drücken Sie <CTRL> + <M>, um alle markierten Werte zu inkrementieren.

- Wählen Sie **Edit** → **Decrement**

oder

- drücken Sie <CTRL> + <N>, um alle markierten Werte zu dekrementieren.

- Wählen Sie **Edit** → **Add Offset**, um zu allen markierten Werte eine Zahl zu addieren.

Wenn Sie diesen Befehl auswählen, werden Sie aufgefordert, einen Offsetwert einzugeben, der dann zu allen markierten Ausgabewerten addiert wird.

- Wählen Sie **Edit** → **Multiply By Factor**, um alle markierten Werte mit einem Faktor zu multiplizieren.

Wenn Sie diesen Befehl auswählen, werden Sie aufgefordert, einen Faktor einzugeben, worauf die Multiplikation mit diesem Faktor erfolgt.

- Wählen Sie **Edit** → **Fill With Values**, um alle markierten Werte zu überschreiben.

Wenn Sie diesen Befehl auswählen, werden Sie aufgefordert, einen Wert einzugeben, dann werden alle markierten Ausgabewerte durch diese Konstante ersetzt.

## Ändern der Anzeige:

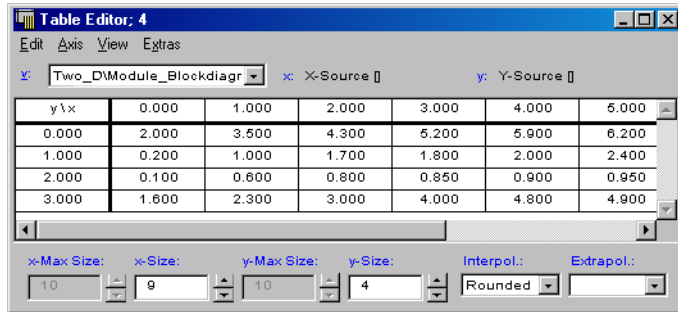
---

- Wählen Sie **View** → **Show Key Help**, um sich die Tastaturkurzbefehle für den 1D-Tabelleneditor unten im Fenster anzeigen zu lassen.

- Wählen Sie **Extras** → **Optimize Size**, um automatisch die Fenstergröße beim Umschalten zwischen den Tabellen anzupassen.

Dieser Befehl wird standardmäßig aktiviert.

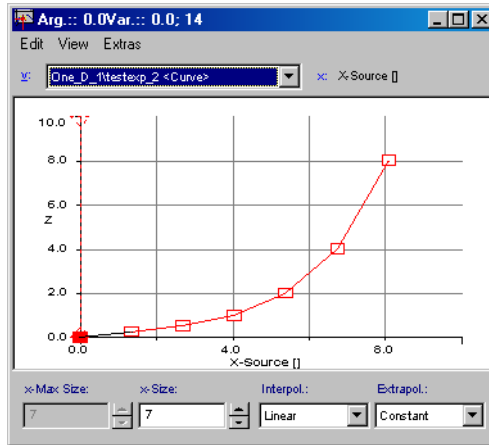
## 2D-Tabelleneditor



Der 2D-Tabelleneditor funktioniert wie der 1D-Editor, der im vorangegangenen Abschnitt beschrieben wurde. Der einzige Unterschied besteht darin, dass die Abtastpunkte zwei Dimensionen anstatt nur einer Dimension haben.

Die Bearbeitung der Datenpunkte und Ausgabewerte funktioniert wie im vorangegangenen Abschnitt beschrieben mit dem Unterschied, dass das Menü **Axis** gesonderte Befehle für die Datenpunkte der x- und y-Achse enthält. Darüber hinaus enthält die 2D-Tabelle paarweise Eingabeflächen für die entsprechenden x- und y-Werte.

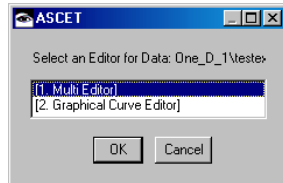
Feste und Gruppentabellen sind auch als 2D-Tabellen verfügbar und funktionieren wie ihre eindimensionalen Varianten.



Kennlinien und -felder können auch grafisch dargestellt und bearbeitet werden. Dieser Abschnitt erläutert die Ansicht und grafische Bearbeitung einer eindimensionalen Tabelle.

Nicht alle Menübefehle des Grafikeditors der eindimensionalen Tabelle werden hier erklärt. Die hier nicht erläuterten Befehle sind mit denen im 1D-Tabelleneditor in der Tabellenansicht identisch. Der 1D-Grafikeditor steht nur in der Experimentierumgebung zur Verfügung.

### Starten des 1D-Grafikeditors:



- Klicken Sie in der Offline-Experimentierumgebung mit der rechten Maustaste auf die Tabelle, die Sie bearbeiten wollen, und wählen Sie **Calibrate** aus dem Kontextmenü.

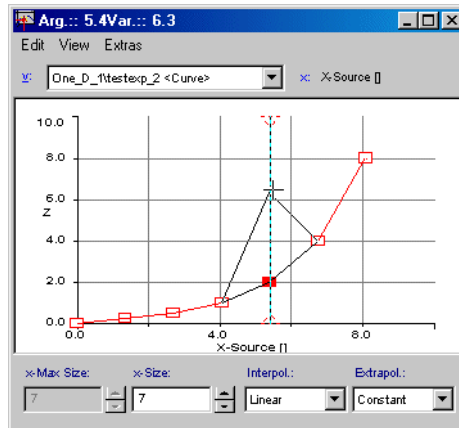
Es erscheint eine Eingabeaufforderung mit einer Liste aller Editoren, die für die ausgewählte Tabelle zur Verfügung stehen.

- Wählen Sie den Eintrag **Graphical Curve Editor** aus.
- Klicken Sie auf **OK**, um den Editor zu öffnen. Der Grafikeditor zeigt die Punkte auf der x-Achse als kleine Quadrate an, die mit einer roten Linie verbunden sind.

## Bearbeiten einer Tabelle im 1D-Grafikeditor:

- Klicken Sie auf das Quadrat, das den Punkt auf der x-Achse darstellt, den Sie bearbeiten wollen.
- Ziehen Sie das Quadrat nach oben oder unten, um den Wert zu verändern.

Der aktuelle Wert für den Punkt wird in der Titelzeile des Editorfensters angezeigt.



- Um den Wert des Punktes auf der x-Achse zu verändern, verschieben Sie den vertikalen Liniencursor für den ausgewählten x-Achsenpunkt.

Der x-Achsenwert wird entsprechend angepasst. Sie können den Cursor bis hin zu einem der benachbarten Abtastpunkte verschieben.

*Oder*

- Wählen Sie **Edit** → **Decrement X Axis Point**, um den ausgewählten x-Achsenpunkt zu dekrementieren.

*Oder*

- Wählen Sie **Edit** → **Increment X Axis Point**, um den ausgewählten x-Achsenpunkt zu inkrementieren.

- Wählen Sie **Edit** → **Block Selection**

oder

- drücken Sie <CTRL> + <B>, um mehrere Abtastpunkte anzupassen.

Sie können den Mauscursor um mehrere Abtastpunkte herum ziehen, um sie auszuwählen. Die Werte dieser Punkte können Sie dann gemeinsam anpassen, indem Sie sie mit der Maus verschieben.

Die Anpassung von mehreren Werten funktioniert nur entlang der z-Achse.

- Wählen Sie erneut **Edit** → **Block Selection**, um wieder die Anpassung für einen Wert vorzunehmen.

### **Entfernen/Hinzufügen von Abtastpunkten:**

---

- Wählen Sie **Edit** → **Remove X Axis Point**, um den gewählten Achsenpunkt zu entfernen.
- Wählen Sie **Edit** → **Add X Axis Point**, um einen neuen Abtastpunkt hinzuzufügen.

#### **Hinweis**

*Wenn die Tabelle bereits die maximale Größe hat, können Sie keine weiteren Abtastpunkte hinzufügen.*

Es erscheint ein Eingabefeld.

- Geben Sie den x-Wert für den Abtastpunkt ein.
- Klicken Sie auf **OK**.

Der neue Abtastpunkt wird an der richtigen Stelle innerhalb der Grafik eingefügt. Er nimmt den Wert des Abtastpunktes zu seiner Linken an.

## Ändern der Anzeige des 1D- Grafikeditors:

---

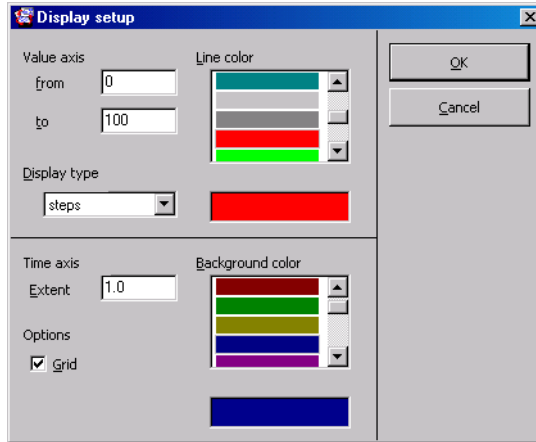
- Wählen Sie **View** → **Grid**, um die Gitternetzlinien aus der grafischen Darstellung auszublenzen.  
Diese Menüauswahl ist umschaltbar, so dass Sie sich bei erneuter Auswahl die Gitternetzlinien wieder anzeigen lassen können.
- Wählen Sie **Extras** → **Colors** → **Black & white**, um auf die Schwarzweißdarstellung des Editors umzuschalten.
- Wählen Sie **Extras** → **Colors** → **Default colors**, um auf die Standardfarben zurückzuschalten.
- Wählen Sie **Extras** → **Colors** → **Invert colors**, um die aktuellen Farben zu invertieren.

### Hinweis

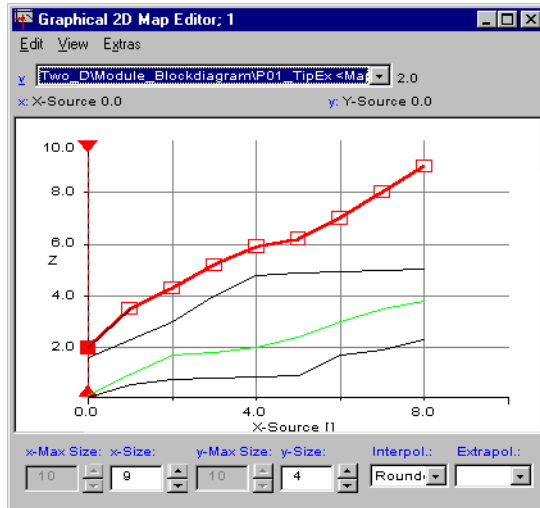
*Koordinatensystem und Beschriftung kann keine andere Farbe zugewiesen werden. Die Darstellung erfolgt grundsätzlich in Schwarz oder, bei Wahl von **Black & white** bzw. **Invert colors** in Weiß.*

## Einrichten der Anzeige des 1D-Grafikeditors:

- Wählen Sie **Extras** → **Display Setup**, um das Dialogfenster „Display setup“ zu öffnen.



- Passen Sie in den Feldern „Value Axis“ und „X-Axis“ die oberen und unteren Begrenzungen für den Wert und die x-Achse an.
- Wählen Sie im Feld „Line color“ eine Farbe für die Kennlinie aus.
- Wählen Sie im Feld „Background color“ eine Hintergrundfarbe aus.
- Passen Sie im Feld „INC/DEC Step“ die Schrittlänge zum Inkrementieren und Dekrementieren an.
- Aktivieren Sie die Option **Grid**, wenn ein Gitter eingeblendet werden soll.
- Klicken Sie auf **OK**, um Ihre Änderungen zu bestätigen.



Mit dem 2D-Grafikeditor ist es möglich, zweidimensionale Tabellen grafisch zu bearbeiten. Dieser Editor ist dem 1D-Grafikeditor ähnlich und stellt die zweidimensionale Tabelle als Sammlung von eindimensionalen Tabellen dar, die einzeln bearbeitet werden können.

### Starten des 2D-Grafikeditors:

- Klicken Sie in der Offline-Experimentierung mit der rechten Maustaste auf die zu bearbeitende Tabelle und wählen Sie **Calibrate** aus dem Kontextmenü.

Es erscheint eine Eingabeaufforderung mit einer Liste der für die ausgewählte Tabelle zur Verfügung stehenden Editoren.

- Wählen Sie den Eintrag **Graphical 2D Map Editor** aus.
- Klicken Sie auf **OK**, um den Editor zu öffnen.



## Bearbeiten einer Tabelle im 2D-Grafikeditor:

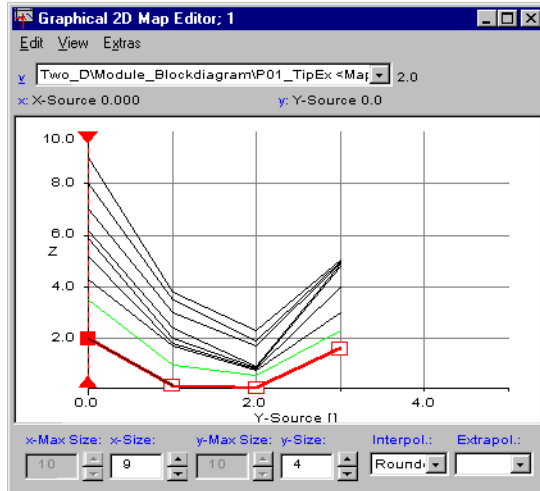
---

- Wählen Sie die Linie aus, die Sie bearbeiten möchten, indem Sie auf sie klicken.  
Die ausgewählte Linie zeigt ein Rechteck für jeden Abtastpunkt an.
- Verschieben Sie die Werte der Abtastpunkte der ausgewählten Linie durch Ziehen wie im 1D-Grafikeditor (siehe Seite 684).  
Die Menüfunktionen zum Inkrementieren/ Dekrementieren der Abtastpunkte heißen hier **Edit** → **Increment Arg** bzw. **Edit** → **Decrement Arg**. Sie wirken entsprechend der gewählten Perspektive (s. Seite 690) auf die X- oder Y-Achse.
- Wechseln Sie zur vorherigen oder nächsten Linie, indem Sie <PFEIL OBEN> oder <PFEIL UNTEN> drücken oder auf eine andere Linie klicken.

Die Menüfunktionen zum Entfernen oder Hinzufügen von Abtastpunkten sind im 2D-Grafikeditor nicht vorhanden.

## Umschalten zwischen den Perspektiven des 2D-Grafikeditors:

- Wählen Sie **View** → **yz-Viewpoint**, um die Achsen auszutauschen und sich die Tabelle aus einer anderen Perspektive anzuschauen.

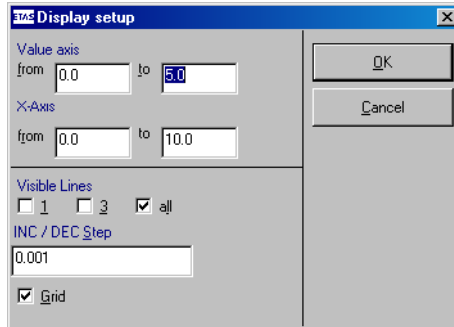


- Sie können nun die Werte und Abtastpunkte der gewählten Achse bearbeiten (s. Seite 689).
- Wählen Sie **View** → **xz-Viewpoint**, um zur Standardperspektive zurückzukehren.

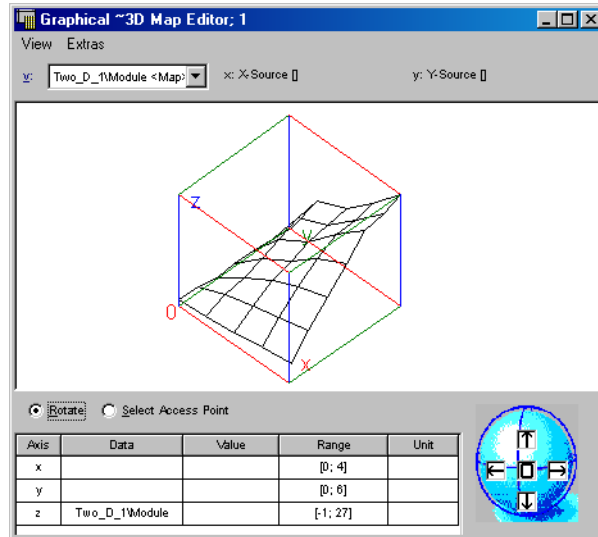
## Einrichten der Anzeige des 2D-Grafikeditors:

---

- Wählen Sie **Extras** → **Display Setup**, um das Dialogfenster „Display setup“ zu öffnen.



- Passen Sie die Bereiche der Werte-Achse und der x-Achse in den entsprechenden Feldern an.
- Aktivieren Sie die gewünschte Option für die Sichtbarkeit der Linie.  
Wenn Sie 1 eingeben, wird nur die aktuell ausgewählte Linie angezeigt. Sie können jedoch noch mit <PFEIL OBEN> und <PFEIL UNTEN> zwischen den Linien wechseln.
- Deaktivieren Sie die Option **Grid**, wenn Sie das Gitternetz ausblenden möchten.
- Passen Sie die Schrittlänge zum Inkrementieren und Dekrementieren an.
- Klicken Sie auf **OK** zum Bestätigen Ihrer Änderungen.



Der 3D-Grafikeditor für 2D-Tabellen stellt eine 2D-Tabelle als dreidimensionales Bild dar. Es ist möglich, die Kurve in alle Richtungen zu drehen und die Werte der Kurve zu messen und bearbeiten. Der 3D-Grafikeditor ist nur in der Experimentierumgebung verfügbar.

### Starten des 3D-Grafikeditors:

- Klicken Sie in der Offline-Experimentierumgebung mit der rechten Maustaste auf die Tabelle, die Sie bearbeiten wollen, und wählen Sie **Calibrate** aus dem Kontextmenü.

Es erscheint eine Eingabeaufforderung mit einer Liste von Editoren, die für die ausgewählte Tabelle zur Verfügung stehen.

- Wählen Sie aus der Liste **Graphical 3D Map Viewer** aus.
- Klicken Sie auf **OK**, um den Editor zu öffnen.

Der 3D-Grafikeditor wird geöffnet. Solange der 3D-Grafikeditor aktiviert ist, bleibt die numerische Darstellung der Tabelle verborgen.

Unterhalb der grafischen Darstellung finden Sie den Eingabebereich. Wenn die Option **Select Access Point** aktiviert ist, werden hier die Werte des aktuell markierten Netzpunktes angezeigt und können durch Eingabe anderer Werte verstellt werden.

### **Bearbeiten der Netzpunkte im 3D-Grafikeditor:**

---

- Wählen Sie die Option **Select Access Point** oberhalb der Liste

*oder*

- drücken Sie die Taste <S>.

Rechts neben der Liste wird jetzt ein Kreuz aus vier Richtungspfeilen dargestellt.



Außerdem ist jetzt bereits ein Netzpunkt markiert: Der Punkt, der dem Ursprung des 3D-Koordinatensystems am nächsten steht, ist durch weiße Verbindungslinien zu seinen Nachbarpunkten hervorgehoben.

- Klicken Sie jetzt auf die Richtungspfeile, um die Markierung nach oben oder unten bzw. links oder rechts zu bewegen,

*oder*

- benutzen Sie die Cursor-Tasten ← ↑ ↓ →.

Die Abtastpunkte der x- und y-Achse werden zusammen mit dem entsprechenden Ausgabewert in der Spalte „Value“ der Tabelle angezeigt. Die Tabelle gibt auch die Bereiche aller drei Achsen an.

- Markieren Sie den Netzpunkt, den Sie bearbeiten wollen.

- Geben Sie die neuen Werte für X, Y und Z in die Spalte „Value“ ein.

### Hinweis

*Bei Gruppen- und Festkennfeldern können nur die Werte der z-Achse geändert werden.*

- Klicken Sie auf das Quadrat in der Mitte der Pfeile

oder

- drücken Sie die Taste <0> (null), um die Markierung zurück zum Ausgangspunkt zu setzen.

Zur Markierung ungünstig gelegener Punkte kann es erforderlich sein, das Koordinatensystem vorher zu rotieren.

### Drehen des Koordinatensystems:

- Wählen Sie die Option **Rotate** oberhalb der Liste

oder

- drücken Sie die Taste <R>.

- Klicken Sie auf die Richtungspfeile in der Drehsteuerung unten rechts im Fenster des 3D-Grafikeditors.

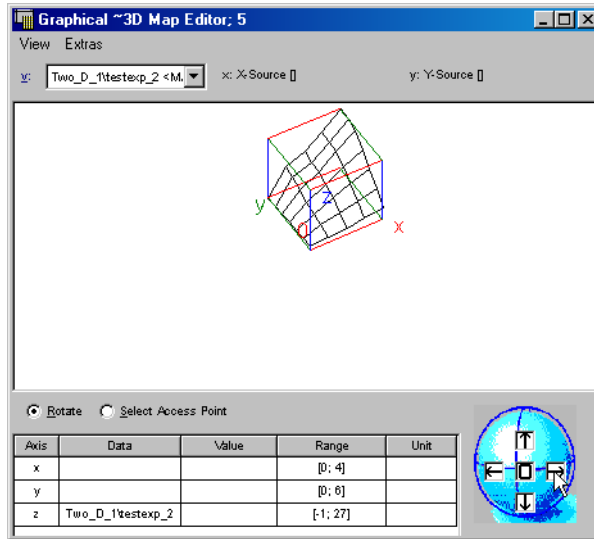


Sie können das Koordinatensystem horizontal oder vertikal in jede Richtung drehen.

- Klicken Sie auf die Schaltfläche **0**, um zum Ausgangsbetrachtungswinkel zurückzukehren.

- Wenn Sie das Koordinatensystem schneller drehen wollen, klicken Sie auf eine Pfeilfläche und halten diese gedrückt.

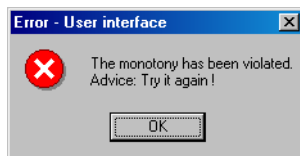
Das Koordinatensystem wird verkleinert, um das Drehen zu beschleunigen.



#### 6.2.4 Verstellen von Stützstellen im Tabelleneditor

Neben den schon beschriebenen Verstellbefehlen im Menü **Edit** kommen für den Tabelleneditor weitere Befehle für das Editieren der Stützstellen auf der X- und Y-Achse hinzu (Menü **Axis**).

Wenn Sie Stützstellenwerte editieren, wird der von Ihnen eingegebene Wert auf die geforderte Monotonie überprüft. Bei Verletzung der Monotonie öffnet sich ein Dialogfenster, um Sie auf die Problematik hinzuweisen.



Die Stützstellenwerte von Kennlinien/-feldern können über den Tabelleneditor in den Tabellenfeldern x und y (y nur bei Kennfeldern) editiert werden. Allerdings können nicht mehrere Stützstellen gleichzeitig verändert werden. Für

Matrizen und Arrays stehen die entsprechenden Befehle nicht zur Verfügung, selbst wenn sie im selben Editorfenster wie Kennlinien/-felder angezeigt werden.

### Verstellen von Stützstellen im Tabelleneditor:

Der hier beschriebene Vorgang gilt für Kennlinien und -felder gleichermaßen. Für Kennlinien stehen Ihnen im Menü **Axis** allerdings nur die Befehle zur Anpassung der X-Achse zur Verfügung.

- Klicken Sie mit der Maus direkt auf den X-Achsen-Wert, den Sie bearbeiten möchten,

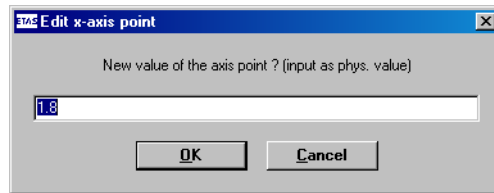
*oder*

- drücken Sie <CTRL> + <X>

*oder*

- wählen Sie **Axis** → **Edit X Axis Point**.

Folgendes Dialogfenster öffnen sich:



- Geben Sie den gewünschten Wert ein.
- Bestätigen Sie mit **OK**

*oder*

- klicken Sie auf **Cancel**, um den Vorgang abzubrechen.

- Wenn Sie den Wert um den voreingestellten Betrag verkleinern wollen:

– Wählen Sie **Axis** → **Decrement X Axis Point**

*oder*

– drücken Sie <CTRL> + <J>.

- Wenn Sie den Wert um den voreingestellten Betrag vergrößern wollen:



- Wählen Sie **Axis** → **Increment X Axis Point**.

*oder*

- Drücken Sie die Taste <CTRL> + <K>.
- Wenn Sie die ausgewählte Stützstelle entfernen wollen, wählen Sie **Axis** → **Remove X Axis Point**.
- Wenn Sie eine Stützstelle hinzufügen wollen, wählen Sie **Axis** → **Add X Axis Point**.

Der Vorgang zum Bearbeiten der Stützstellen auf der Y-Achse (nur Kennfelder) ist identisch. Sie müssen lediglich die entsprechenden Befehle für die Y-Achse verwenden.

Bei Fest- und Gruppenkennlinien und -feldern sowie bei Verteilungen können die Stützstellen nicht bearbeitet werden. Das Menü **Axis** ist deaktiviert (Gruppen-/Festkennlinien/-felder) oder fehlt ganz (Verteilungen).

## 6.3 Messfenster

---

Je nachdem, welche Messgrößen verwendet werden sollen, wählen Sie das geeignete Messfenster sowie die geeigneten Darstellungs- und Messoptionen für das jeweilige Messfenster aus. Mit den Messfenstern lassen sich die Werte jedes Elements in ASCET während eines Experiments anzeigen. Für jedes gemessene Element wird ein Messkanal eingerichtet; es ist möglich, verschiedene Messkanäle für ein Element anzulegen, z.B. dann, wenn die Anzeige in zwei verschiedenen Messfenstern erfolgen sollen. Es besteht auch die Möglichkeit, Messdaten anzuzeigen, ohne sie zu speichern, Messdaten anzuzeigen und die Aufzeichnung gleichzeitig zu starten und zu stoppen oder Messdaten aufzuzeichnen ohne vorherige Anzeige.

### 6.3.1 Messfenster auswählen

---

Sie haben die Auswahl zwischen einer Anzahl von Messfenstern. Folgende Messfenster können ausgewählt werden:

- Oszilloskop
- Numerische Anzeige
- Vertikale Balkenanzeige
- Horizontale Balkenanzeige
- Bitanzeige
- Recorder

Wie Sie aus der Offline-Experimentierumgebung ein Messfenster öffnen, ist im Abschnitt „Das Mess-System“ auf Seite 625 beschrieben.

### 6.3.2 Allgemeine Beschreibung von Menüfunktionen

---

Je nach Experimentierumgebung und markiertem Fenster ändern sich die zur Verfügung stehende Einträge.

- **Extras:**
  - *Change title*  
Verändern des Namens für das markierte Messfenster.
  - *Message when out of bounds*  
Meldet dem Benutzer, wenn ein Messwert über oder unter der definierten Messgrenze liegt.
  - *Setup* (<CTRL> + <S>)  
Öffnet das Setup-Fenster.
  - *Colors*  
Verändert die Farbeinstellung für das gesamte Messfenster.  
*Black & white* – Schaltet auf die Schwarzweißdarstellung um.  
*Default colors* – Schaltet auf die Standardfarben zurück.  
*Invert colors* – Invertiert die aktuellen Farben.
  - *Physical representation* (<CTRL> + <P>)  
Stellt die markierte(n) Messgröße(n) als physikalischen Wert dar.
  - *Hexadec. representation* (<CTRL> + <H>)  
Stellt die markierte(n) Messgröße(n) hexadezimal dar.
  - *Decimal representation* (<CTRL> + <Z>)  
Stellt die markierte(n) Messgröße(n) dezimal dar.
  - *Binary representation* (<CTRL> + <R>)  
Stellt die markierte(n) Messgröße(n) binär dar.
  - *Copy variable to window*  
Kopiert die markierte Messgröße in ein anderes Messfenster.
  - *Move variable to window*  
Verschiebt die markierte Messgröße in ein anderes Messfenster.
  - *Remove variable*  
Entfernt die markierte Messgröße aus dem Messfenster.
  - *Change measure rate*  
Verändert die Abtastzeit (abhängig von der Hardwarekonfiguration).
  - *About Variable* (<CTRL> + <I>)  
Öffnet ein Fenster mit Informationen zu der markierten Messgröße.

- *Attributes*
  - Copy* → kopiert Darstellungsoption der markierten Messgröße.
  - Paste* → weist Darstellungsoption der markierten Messgröße zu (nur möglich, wenn vorher Copy ausgeführt wurde).
- *Move* (nur bei mehreren Messgrößen in einem Fenster)
  - Up* → verschiebt die markierte Messgröße eine Position nach oben.
  - Down* → verschiebt die markierte Messgröße eine Position nach unten.
  - Left* → verschiebt die markierte Messgröße eine Position nach links (bei der vertikalen Balkenanzeige).
  - Right* → verschiebt die markierte Messgröße eine Position nach rechts (bei der vertikalen Balkenanzeige).
- **File:**
  - *Print*  
Öffnet eine Dialogfenster zum Drucken des Oszilloskop- oder Recorderfensters.
  - *Copy To Clipboard*  
Das Oszilloskop- oder Recorderfenster wird in die Zwischenablage kopiert (Screenshot) und kann so in beliebige Anwendungen eingefügt werden.
  - *Save Selected Channels*  
Abspeichern des ausgewählten Messkanals.  
*MDF* → Messkanal im MDF-Format abspeichern.  
*FAMOS* → Messkanal im FAMOS-Format abspeichern.
  - *Save All Channels*  
Abspeichern aller Messkanäle.  
*MDF* → Alle Messkanäle im MDF-Format abspeichern.  
*FAMOS* → Alle Messkanäle im FAMOS-Format abspeichern.
- **Edit:**
  - *Define trigger*  
Ermöglicht das Definieren einer Triggerbedingung (wirkt sich nur auf die Anzeige aus).
  - *Activate trigger*  
Aktiviert bzw. deaktiviert den Trigger.
  - *Trigger manually*  
Löst das Triggersignal manuell aus.
  - *Autoscale* (<CTRL> + <A>)  
Passt die Skalierung der Y-Achse an einen markierten Messkanal an.

- *Autoscale all channels*  
Passt die Skalierung der Y-Achse an alle Messkanäle an.
- *Undo last scaling* (<CTRL> + <U>)  
Setzt den letzten Skalierungsbefehl zurück.
- *Autodistribution*  
Verteilt die markierten Kanäle auf getrennte Anzeigebereiche.
- *Analyse measure data* (<CTRL> + <V>)  
Wechselt in den Analysemodus.
- *Analysis setup*  
Definiert Voreinstellungen für den Analysemodus (nur im Analysemodus aktiv).
- *Analyse next point*  
Bewegt den Messcursor zum nächsten Messpunkt (nur im Analysemodus aktiv).
- *Analyse previous point*  
Bewegt den Messcursor zum vorigen Messpunkt (nur im Analysemodus aktiv).
- **View:**
  - *Show selected channel*  
Blendet den oder die markierten Messkanäle ein bzw. aus.
  - *Grid*  
Ermöglicht Ihnen die Definition eines Hintergrundgitters. Mögliche Einstellungen sind „kein“, „dynamisches“ und „festes“ Gitter.
  - *Show measure channel lists*  
Blendet die Liste mit den Messkanälen ein bzw. aus.
  - *Min/Max*  
Blendet in der Messkanal-Liste die Min/Max-Anzeige ein bzw. aus.
  - *Rate*  
Blendet in der Messkanal-Liste die Abtastrate ein bzw. aus.
  - *Value at active cursor*  
Blendet die Werte der Messkanäle an der aktiven Cursorposition in der Messkanal-Liste ein bzw. aus (nur im Analysemodus).
  - *Differences between cursors*  
Blendet die Differenzwerte zwischen den beiden Cursorpositionen für die Messgrößen in der Kanalliste ein bzw. aus (nur im Analysemodus).

- *Show key help*  
Blendet die wichtigsten Tastaturbefehle in der Fußzeile ein bzw. aus.
- *Show Setup*  
Blendet in der Fußzeile Einstelloptionen ein bzw. aus.
- *Larger font*  
Schaltet die Anzeige auf große Schrift.

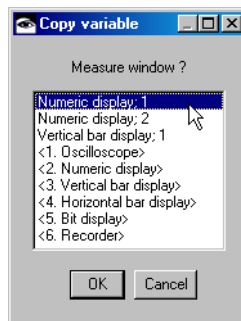
### 6.3.3 Arbeiten mit Messfenstern

Jedes Messfenster kann die Daten von verschiedenen Messkanälen anzeigen, wobei es möglich ist, die Messkanäle zwischen den Fenstern zu kopieren und zu verschieben. Beim Verschieben eines Kanals wird dieser Kanal in einem Fenster gelöscht und einem anderen Fenster hinzugefügt. Beim Kopieren eines Kanals wird ein neuer Messkanal für das gleiche Element in einem neuen Fenster angelegt.

#### **Kopieren von Kanälen zwischen Messfenstern:**

- Wählen Sie den oder die zu kopierenden Kanäle aus, indem Sie auf die entsprechenden Felder im Messfenster klicken.
- Wählen Sie **Extras** → **Copy variable to window**.

Das Dialogfenster „Copy variable“ wird geöffnet. Es enthält eine Liste aller verfügbaren Messfenster.



- Wählen Sie das Fenster aus, in das Sie die markierten Kanäle kopieren wollen.

- Klicken Sie auf **OK**.

Wenn Sie ein neues Fenster auswählen, wird es mit den kopierten Kanälen geöffnet. Andernfalls werden die Kanäle in das vorhandene Fenster kopiert.

### **Verschieben von Kanälen zwischen Messfenstern:**

---

- Wählen Sie den oder die Kanäle aus, die Sie verschieben wollen.

- Wählen Sie **Extras** → **Move variable to window**.

Das Dialogfenster „Move variable“ wird geöffnet. Es enthält eine Liste aller verfügbaren Messfenster.

- Wählen Sie das Fenster aus, in das Sie die markierten Kanäle verschieben wollen.

- Klicken Sie auf **OK**.

Die markierten Kanäle werden in das ausgewählte Fenster (neu oder vorhanden) verschoben und aus dem ursprünglichen Fenster entfernt. Wenn das alte Fenster dadurch leer wird, wird es geschlossen.

### **Kanäle aus einem Messfenster löschen:**

---

Um Kanäle aus einem Messfenster zu löschen, gehen Sie wie folgt vor:

- Wählen Sie den oder die zu löschenden Kanäle aus.

- Wählen Sie **Extras** → **Remove variable**

*oder*

- Drücken Sie die Taste <DEL> (nur Oszilloskop/Recorder).

Die markierten Kanäle werden aus dem Fenster entfernt. Wenn das Fenster dadurch leer wird, wird es geschlossen.

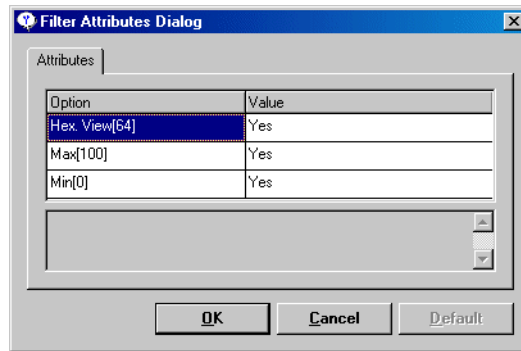
Einstellungen aus einem Messfenster können in ein anderes Messfenster kopiert werden, um die Einrichtung komplexer Experimente zu ermöglichen. Nur die für ein bestimmtes Messfenster geltenden Einstellungen werden kopiert, und Sie können auswählen, welche Einstellungen kopiert werden sollen.

## Austausch von Attributen zwischen Messfenstern:

---

- Wählen Sie in dem Messfenster, aus dem Sie die Einstellungen kopieren wollen, **Extras** → **Attributes** → **Copy**.
- Wählen Sie in dem Messfenster, in das Sie die Einstellungen kopieren wollen, **Extras** → **Attributes** → **Paste**.

Das Dialogfenster „Filter Attributes Dialog“ wird geöffnet. Es enthält alle Attribute, die für das neue Messfenster gelten, sowie deren Einstellungen im alten Messfenster.



- Klicken Sie in der Spalte „Values“ auf den Wert eines Attributs.  
Das Feld wird zum Kombikästchen.
- Wählen Sie aus dem Kombikästchen **Yes**, um das Attribut auszuwählen.
- Wählen Sie aus dem Kombikästchen **No**, um das Attribut nicht auszuwählen.  
Wenn Sie **No** auswählen, gilt das Attribut nicht für das nächste Messfenster.
- Klicken Sie auf **OK**.  
Die ausgewählten Attribute werden in das neue Messfenster kopiert.

## Ändern der Anzeige eines Messfensters:

---

Mit den nachstehend beschriebenen Menüfunktionen können Sie die Anzeige der Messfenster verändern.

- Markieren Sie in einem Messfenster den oder die Kanäle, deren Anzeige Sie ändern wollen.
- Wählen Sie **Extras** → **Physical representation** aus, wenn Sie sich die physikalischen Werte anzeigen lassen wollen.
- Wählen Sie **Extras** → **Hexadec. representation** aus, wenn Sie eine Hexadezimaldarstellung der Messwerte wünschen.  
Diese Option ist für Experimente mit einem Festkomma-Code sinnvoll.

### Hinweis

*Die beiden folgenden Menüfunktionen stehen nur in der Numerischen Anzeige zur Verfügung.*

- Wählen Sie **Extras** → **Decimal representation** aus, um sich die Messwerte als ganze Zahlen anzeigen zu lassen.
- Wählen Sie **Extras** → **Binary representation** aus, um sich die Messwerte im Binärformat anzeigen zu lassen.
- Wählen Sie **Extras** → **Message when out of bounds**, damit immer dann, wenn ein Messwert die für den Kanal festgelegten Überwachungsgrenzen überschreitet, ein Mitteilungsfenster geöffnet wird.  
Dieser Befehl ist im Oszilloskop bzw. Recorder nicht verfügbar.

## Titel eines Messfensters ändern:

---

Um den Titel eines Messfensters zu ändern, gehen Sie wie folgt vor:

- Wählen Sie in dem Messfenster, dessen Titel Sie ändern wollen, **Extras** → **Change title**.



- Geben Sie im Eingabefenster den neuen Titel ein.

Wenn der neue Titel Leerzeichen enthält, wird nur der Teil vor dem ersten Leerzeichen übernommen. Mehrere Worte können z.B. mit Binde- oder Unterstrich verbunden werden.

- Klicken Sie **OK**.

Der neue Titel erscheint in der Kopfzeile des Messfensters.

### Informationen über Variable anzeigen:

---

Um Informationen über einen oder mehrere ausgewählte Kanäle anzuzeigen, gehen Sie wie folgt vor:

- Wählen Sie in einem Messfenster einen oder mehrere Kanäle aus.
- Wählen Sie in diesem Fenster **Extras** → **About Variable**

oder

- drücken Sie <CTRL> + <I>.

Für jeden ausgewählten Kanal wird ein Fenster „Information“ geöffnet, das Informationen über die jeweilige Variable enthält.

- Klicken Sie auf **OK**, um das Fenster „Information“ zu schließen.

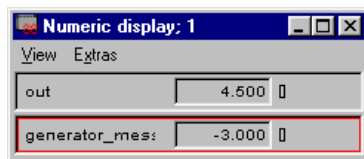
### 6.3.4 Die verschiedenen Messfenster

---

#### *Numerische Anzeige*

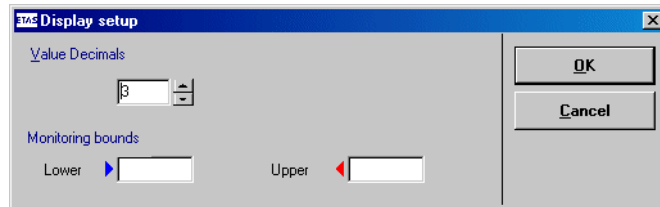
---

Die numerische Anzeige zeigt die Werte der Messkanäle, für die sie gestartet wurde, wobei die Darstellung dezimal, binär oder hexadezimal erfolgen kann.



## Einrichten einer numerischen Anzeige:

- Wählen Sie die numerische Anzeige aus, die Sie einrichten wollen.  
Sie können mehr als einen Kanal innerhalb des gleichen Fensters auswählen, indem Sie auf die entsprechenden Kanäle klicken und dabei <CTRL> gedrückt halten.
- Wählen Sie **Extras** → **Setup**.  
Es erscheint das folgende Dialogfenster.



- Geben Sie im Feld „Value Decimals“ die Anzahl der Dezimalstellen an, mit denen der Messwert angezeigt wird.
- In den Feldern unter „Monitoring bounds“ können Sie eine oder zwei Überwachungsgrenzen definieren.
- Klicken Sie **OK**, um das Fenster zu schließen.  
Über- oder Unterschreiten der Überwachungsgrenzen wird durch eine rote Warnlampe in der Titelzeile signalisiert. Zusätzlich zeigt ein blaues Quadrat links neben der Anzeige ein Unterschreiten der unteren, ein rotes Quadrat ein Überschreiten der oberen Grenze an. Zusätzlich öffnet sich ein Meldungsfenster, das automatisch verschwindet, wenn der Messwert wieder innerhalb der Grenzen liegt.



## Ändern der Anzeigeeoptionen einer numerischen Anzeige:

---

- Wählen Sie mit **Edit** → **<Format> representation** ein Format für die Darstellung der Messwerte wie auf Seite 704 beschrieben.
- Wählen Sie **Extras** → **Move** → **Up** oder **Down**, um den ausgewählten Messkanal im numerischen Anzeigefenster nach oben oder nach unten zu bewegen.  
Dieser Befehl steht nur zur Verfügung, wenn das Fenster mehr als einen Kanal enthält.
- Wählen Sie **View** → **Larger font**, um die Messung in einer größeren Schrift anzeigen zu lassen.

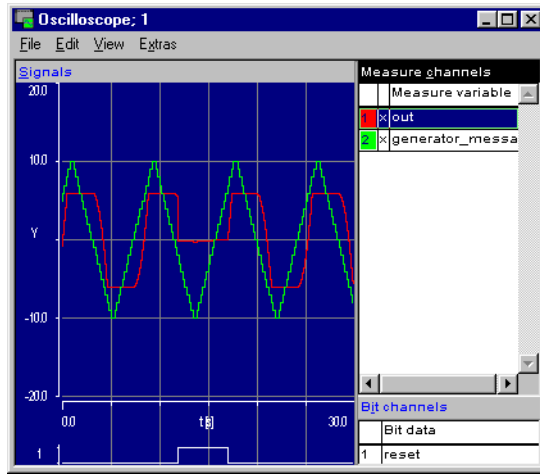
### *Oszilloskop*

---

Das Oszilloskop bietet eine Reihe von hochgradig flexiblen Darstellungsoptionen, die mit denen eines tatsächlichen Oszilloskops vergleichbar sind. Oft kann es von Vorteil sein, verschiedene Oszilloskopfenster gleichzeitig zu öffnen, da zu viele Kanäle die Lesbarkeit der Anzeige stören können.

Wenn Sie das Oszilloskop auswählen, beginnen Sie gewöhnlich damit, die Anzeigeeoptionen einzurichten, d.h. die Art und Weise festzulegen, in der die verschiedenen Werte im Fenster dargestellt werden. Während des Experiments können Sie verschiedene Operationen an den angezeigten Daten ausführen, wie z.B. Daten speichern, analysieren, Trigger definieren usw.

Die Fläche „Signals“ zeigt die Kurven für die gemessenen numerischen Werte an. Darunter befindet sich der Anzeigebereich „Bit Channels“, der die gemessenen logischen Werte darstellt. Rechts von der Fläche „Signals“ befindet sich das Feld „Measure channels“, wo die Namen der Kanäle mit den zugehörigen verschiedenen benutzerdefinierbaren Informationen angezeigt werden.



### Ändern der Oszilloskopoberfläche:

Die folgenden Optionen beeinflussen die ausgewählten Messkanäle. Sie können einen Kanal, alle Kanäle oder eine beliebige Kombination von Kanälen auswählen. Die Optionen können sowohl vor Beginn als auch während einer Messung gesetzt werden.

### Einrichten von Messkanälen:

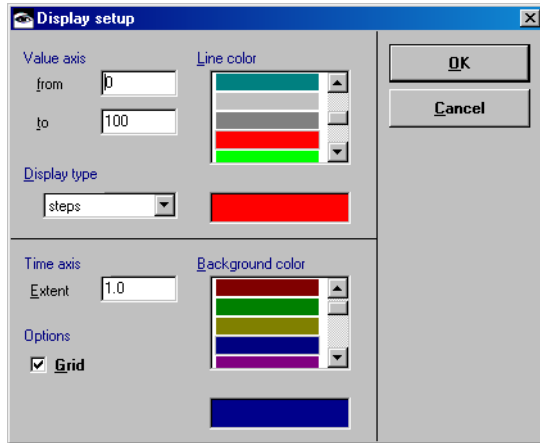
- Wählen Sie in der Fläche „Measure channels“ die Kanäle aus, die Sie einrichten wollen.  
Sie können mehr als einen Kanal auswählen, indem Sie auf die entsprechenden Kanäle klicken und dabei die Taste <CTRL> gedrückt halten.

#### Hinweis

*Bit-Kanäle können nicht eingerichtet werden.*

- Wählen Sie **Extras** → **Setup**  
oder

- doppelklicken Sie auf den gewünschten Kanal  
oder
- drücken Sie die Taste <S>.  
Das Dialogfenster „Display setup“ wird geöffnet.



- Stellen Sie in den Feldern „from“ und „to“ die untere und obere Grenze ein.  
Die Werte, die Sie hier eingeben, bilden die untere und obere Begrenzung der y-Achse in der Oszilloskopoberfläche. Der Standardbereich beträgt 0 – 100.
- Wählen Sie im Feld „Line color“ eine Farbe für den Kanal aus.

### Hinweis

*Diese Option ist nur verfügbar, wenn Sie im Feld „Measure channels“ einen einzigen Kanal ausgewählt haben.*

Der gewählte Kanal wird in dieser Farbe dargestellt. Sie können sich verschiedene Kanäle mit der gleichen Farbe anzeigen lassen, aller-

dings wird die Anzeige dann leicht unübersichtlich. Als Standardeinstellung ist jedem Kanal eine andere Farbe zugeordnet.

- Wählen Sie aus dem Kombikästchen „Display type“ die Linienart aus.

Wenn Sie `steps` auswählen, wird eine Stufe zwischen jeweils zwei Messwerten gezeichnet; wählen Sie dagegen `line` aus, werden sie durch

- Klicken Sie auf **OK**, um die Einstellungen zu übernehmen.

Das Dialogfenster „Display setup“ enthält einige Optionen, die keine Auswirkungen auf die einzelnen Kanäle, sondern auf das gesamte Oszilloskopfenster haben.

#### **Einstellen des Oszilloskopfensters:**

---

- Öffnen Sie das Dialogfenster „Display setup“, wie oben beschrieben.

- Passen Sie den Wert im Feld „Time axis“ an.

Der Wert, den Sie hier eingeben, bestimmt das Zeitfenster, das vom Oszilloskop angezeigt wird, d.h., wenn Sie den Wert auf 1 stellen, zeigt das Oszilloskop die Ausgabe einer Sekunde an; geben Sie 0,5 ein, erfolgt die Anzeige von einer halben Sekunde usw. Der Standardwert ist 1 Sekunde.

#### **Hinweis**

*Beachten Sie, dass bei Offline-Experimenten diese Werte nicht der Echtzeit entsprechen, die für die Berechnungen benötigt wird.*

- Wählen Sie aus dem Feld „Background color“ eine Hintergrundfarbe aus.
- Setzen Sie die Option **Grid** aktiv, um sich das Achsengitternetz im Display-Bereich anzeigen zu lassen, oder entfernen Sie das Häkchen, um dieses Gitternetz zu verbergen.

- Klicken Sie auf **OK**.

Die Einstellungen werden übernommen.

Wenn Sie das Zeitfenster geändert haben, werden die schon in der Anzeigefläche dargestellten Kurven gelöscht.

### **Farbeeinstellungen für das Oszilloskop wechseln:**

---

- Wählen Sie **Extras** → **Colors** → **Black & white**, um die Schwarzweißanzeige auszuwählen.
- Wählen Sie **Extras** → **Colors** → **Default colors**, um die Standardfarben wiederherzustellen.
- Wählen Sie **Extras** → **Colors** → **Invert colors**, um die aktuellen Farbeinstellungen zu invertieren.

Die folgenden Optionen dienen dazu, die Kanäle bei laufender oder angehaltener Messung automatisch zu skalieren. Die automatische Skalierung verwendet den aktuellen Fensterinhalt zur Berechnung der Anpassung. Sie können einen Kanal, alle Kanäle oder eine beliebige Kombination von Kanälen auswählen.

### **Messkanäle skalieren:**

---

- Wählen Sie den oder die Kanäle, die Sie skalieren möchten.
- Wählen Sie **Edit** → **Autoscale**.  
Die Y-Achse wird für jeden Kanal so an die aktuellen Werte im Fenster angepasst, dass die gesamte Anzeigefläche ausgenutzt wird.
- Wählen Sie **Edit** → **Autoscale all channels**, um gleichzeitig eine automatische Skalierung aller Kanäle vorzunehmen.

Das Ergebnis ist das gleiche wie bei der Durchführung der Operation für jeden einzelnen Kanal.

- Wählen Sie **Edit** → **Autodistribution**, um die markierten Kanäle auf getrennte Anzeigebereiche zu verteilen.

Die Kanäle werden, ausgehend vom aktuellen Fensterinhalt, so skaliert, dass sie sich in der Anzeigefläche nicht überschneiden.

- Wählen Sie **Edit** → **Undo last scaling**

oder

- drücken Sie <CTRL> + <U>, um die letzten Skalierungsoperation rückgängig zu machen.

#### **Hinweis**

Wurde **Autoscale all channels** angewendet, ohne dass alle Kanäle ausgewählt waren, wird die Umskalierung der nicht gewählten Kanäle mit diesem Befehl **nicht** rückgängig gemacht.

#### **Messkanäle ein-/ausblenden:**

Sie können einen oder mehrere Kanäle ausblenden, ohne sie gleich aus dem Oszilloskop (oder Recorder) zu löschen. Gehen Sie wie folgt vor:

- Wählen Sie einen oder mehrere Kanäle.
- Wählen Sie **View** → **Show selected channel**

oder

- drücken Sie die Taste <X>.

Die gewählten Kanäle werden in der Anzeigefläche ein- oder ausgeblendet, bleiben aber in der Liste „Measure channels“.

#### **Listen ein-/ausblenden:**

Um die Listen „Measure channels“ und „Bit channels“ ein- oder auszublenden, gehen Sie wie folgt vor:

- Wählen Sie **View** → **Show measure channel lists**

oder

- drücken Sie die Taste <L>.

Die Listen „Measure channels“ und „Bit channels“ werden ausgeblendet.



- Wiederholen Sie den vorigen Schritt, um die Listen wieder einzublenden.

#### **Hinweis**

*Die beiden Listen können nur zusammen ein- oder ausgeblendet werden.*

### **Listen einrichten (Erfassungsmodus):**

---

Um die Listen „Measure channels“ und „Bit Channels“ einzurichten, gehen Sie wie folgt vor:

- Wählen Sie **View** → **Min/Max**.  
Die Spalte „Min..Max“ wird in den Listen „Measure channels“ und „Bit channels“ hinzugefügt. Sie enthält die Minimal- und Maximalwerte der Kanäle.
- Wählen Sie **View** → **Rate**.
- Die Spalte „Rate“ wird hinzugefügt. Sie enthält die verwendete Abtastrate für jeden Kanal.

### **Tastaturbefehle anzeigen:**

---

Um sich die Tastaturbefehle anzuzeigen, gehen Sie wie folgt vor:

- Wählen Sie **View** → **Show key help**.  
Die Tastaturbefehle werden unten im Oszilloskopfenster angezeigt.  
Es werden nur die Tastenkurbefehle angezeigt, die in einem bestimmten Kontext relevant sind; z.B. werden die Kurzbefehle für den Analysemodus nicht im Erfassungsmodus angezeigt.
- Wiederholen Sie den Befehl, um die Anzeige zu entfernen.

### **Setup anzeigen:**

---

Um die aktuellen Einstellungen für einen Kanal anzuzeigen, gehen Sie wie folgt vor:

- Wählen Sie einen Kanal.

- Wählen Sie **View** → **Show Setup**.  
Die aktuellen Einstellungen für die Achsen werden unten im Oszilloskopfenster angezeigt.  
Wenn Sie bei angeschalteter Anzeige einen anderen Kanal auswählen, werden dessen Einstellungen angezeigt.
- Wiederholen Sie den Befehl, um die Anzeige zu entfernen.

Mit der Standardeinstellung befindet sich das Oszilloskop im *Erfassungsmodus*, d.h. es zeigt die im Experiment erzeugten Daten an. Es gibt darüber hinaus einen *Analysemodus* für die weitere Analyse Ihrer Daten. Der Analysemodus steht nur zur Verfügung, wenn das Experiment beendet oder unterbrochen wurde.

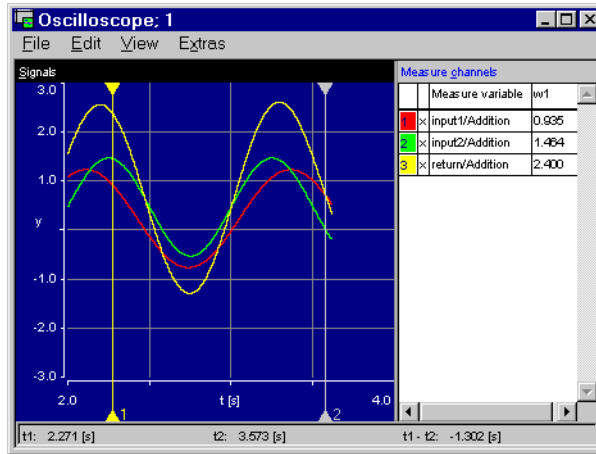
### **Analysemodus starten:**

---

Um den Analysemodus zu starten, gehen Sie wie folgt vor.

- Wählen Sie **Edit** → **Analyse measure data**  
oder
- drücken Sie die Taste <v>.  
Zwei senkrechte Linien erscheinen in der Anzeigefläche. Die anfänglich linke Linie hat am unteren Ende die Nummer 1, die andere die Nummer 2. Diese Linien sind die Messcursor; sie legen zwei Punkte fest, an denen Werte gemessen werden. Der aktive Messcursor ist gelb, der andere grau. Am unteren Rand des Oszilloskopfensters werden die Zeit-

werte, d.h. die Position von beiden Cursors auf der x-Achse, sowie deren Differenz angezeigt.



### Listen einrichten (Analysemodus):

Um die Listen „Measure Channels“ und „Bit Channels“ einzurichten, gehen Sie wie folgt vor:

- Wählen Sie **View** → **Value at active cursor**.  
Die Spalte „wx“ wird in den Listen „Measure channels“ und „Bit channels“ hinzugefügt. Sie enthält die Werte der Kanäle an der Position des aktiven Cursors. „x“ nimmt den Wert 1 oder 2 an, je nachdem, welcher Messcursor aktiv ist.
- Wählen Sie **View** → **Difference between cursors**.  
Die Spalte „wx - wy“ wird hinzugefügt. Sie enthält die Differenz zwischen den Werten am aktiven und am inaktiven Cursor. x steht für die Nummer des aktiven Messcursors, y für die des inaktiven.
- Ändern Sie die Fenstergröße oder die Größe des Anzeigebereichs so, dass alle Informationen sichtbar sind.

- Ziehen Sie bei Bedarf an den senkrechten Trennlinien, um die Spaltenbreiten so zu verändern, dass alle Information lesbar sind.

### **Analysieren von Messdaten:**

---

Um die im Oszilloskop gemessenen Daten zu analysieren, gehen Sie wie folgt vor:

- Starten Sie den Analysemodus (s. Seite 714).
- Markieren Sie den Messcursor, den Sie verschieben wollen.  
Der Messcursor wird aktiv. Die Spalten „wx“ und „wx - wy“ in den Listen „Measure channels“ und „Bit channels“ werden aktualisiert. x ist die Nummer des aktiven Cursors, y die des inaktiven Cursors.
- Ziehen Sie den Messcursor an die gewünschte Position

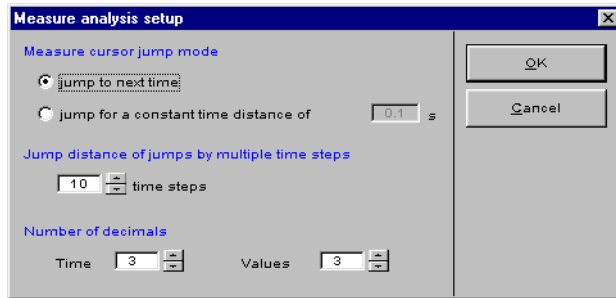
*oder*

- bewegen Sie den Cursor mit den Pfeiltasten.  
Die aktuellen Messwerte an der Position des Cursors werden in der Liste „Measure channels“ angezeigt. Die x-Achsen-Werte erscheinen unten im Oszilloskopfenster.
- Vergrößern Sie bei Bedarf das Oszilloskopfenster.
- Ziehen Sie bei Bedarf an den senkrechten Trennlinien zwischen den Spalten, damit alle Informationen lesbar werden.

Es gibt verschiedene Möglichkeiten, die die Art und Weise beeinflussen, in der Messungen analysiert werden können. Diese Optionen stehen nur zu Verfügung, wenn sich das Oszilloskop im Analysemodus befindet.

## Einrichten der Analyse:

- Wählen Sie **Edit** → **Analysis setup**.  
Das Dialogfenster „Measure analysis setup“ wird geöffnet.



- Wählen Sie unter „measure cursor jump mode“ den für Sie passenden Sprungmodus für die Messcursor.
  - **jump to next time** bedeutet, dass sich der Cursor kontinuierlich von einem Wert zum nächsten bewegt.
  - **jump for a constant time distance of ...** bedeutet, dass sich der Cursor in Schritten bewegt, deren Schrittlänge Sie im Kästchen **s** neben der Option festlegen können.  
Geben Sie zum Beispiel einen Wert 0,1 ein, bewegt sich der Cursor in Schritten von 0,1 Sekunden auf der x-Achse.
- Geben Sie im Kästchen „time steps“ unterhalb von „Jump distance of jumps by multiple time steps“ einen Faktor für den Wert im Kästchen **s** ein.

Wenn Sie z.B. erst einen Wert von 0,1 und dann einen Faktor 3 eingeben, beträgt die tatsächliche Schrittlänge auf der x-Achse für jede Cursorbewegung 0,3 Sekunden.

- Geben Sie im Kästchen „Time“ unterhalb von „Number of Decimals“ die Anzahl der Dezimalstellen für die Zeitwerte ein.
- Geben Sie im Kästchen „Values“ die Anzahl der Dezimalstellen für die Messwerte ein.
- Klicken Sie auf **OK**.

Die neuen Einstellungen werden übernommen. Sie sind erst dann aktiv, wenn Sie einen Messcursor bewegen.

Wurde ein Triggerzustand aktiviert, zeigt das Oszilloskop keine Werte an, bis der Triggerzustand erreicht wurde. Nach Erreichen des Zustands zeigt das Oszilloskop die Werte ganz normal an. Die Triggerfunktion hat nur Einfluss auf die Anzeige der Werte im Oszilloskop, nicht auf die Berechnung im Experiment.

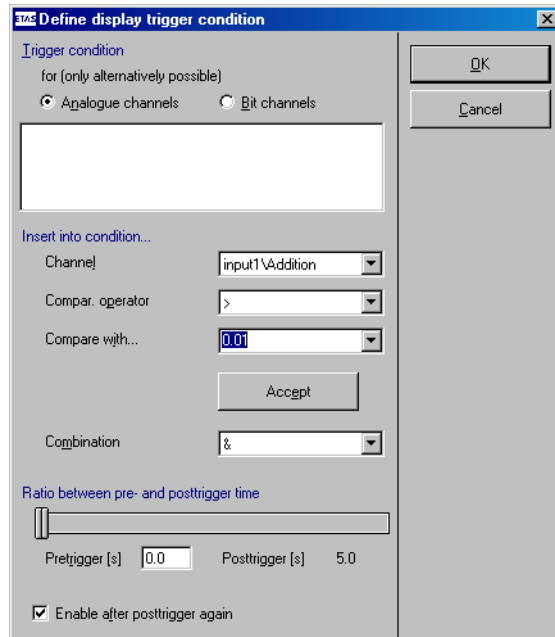
Sie können nur dann einen Triggerzustand definieren, wenn das Experiment beendet oder unterbrochen wurde.

## Definieren von einfachen Triggerereignissen:

Um einen einfachen Trigger zu definieren, gehen Sie wie folgt vor:

- Wählen Sie **Edit** → **Define trigger**.

Das Dialogfenster „Define display trigger condition“ öffnet sich.



- Wählen Sie einen Triggermodus aus, indem Sie auf die Option **Analogue channels** oder **Bit channels** klicken.

Triggerzustände können entweder nur für Analog- oder nur für Bit-Kanäle definiert werden.

- Wählen Sie im Kombikästchen „Channel“ einen Kanal.

- Wählen Sie im Kombikästchen „Compar. operator“ einen Vergleichsoperator.

Ihnen stehen die Operatoren == (gleich), > (größer als), < (kleiner als), `exceeds` und `falls short` zur Verfügung.

- Wählen Sie im Kombikästchen „Compare with“ den Kanal, den Sie mit dem ersten vergleichen wollen,

oder

- tragen Sie eine Zahl bzw. `true` oder `false` ein.

### **Hinweis**

*Sie müssen hier als Dezimaltrennzeichen einen Punkt verwenden. Das Komma führt zu einer Fehlermeldung.*

- Klicken Sie auf **Accept**.

Die von Ihnen in den vorausgegangenen Schritten definierte Bedingung wird in das Textfenster übernommen.

- Alternativ zu den vorigen Schritten können Sie die Bedingung auch direkt im Textfenster eingeben.

- Klicken Sie **OK**, um das Fenster zu schließen.

Der Triggerzustand ist nun definiert. Er ist automatisch aktiviert und wird verwendet, wenn Sie die nächste Messung starten.

Im hier gezeigten Beispiel wird der Triggerzustand ausgelöst, wenn der Wert des Kanals `input1/Addition` größer als `0.01` ist. Es ist auch möglich, Kanäle miteinander zu vergleichen, indem ein Kanal im Kombikästchen „Compare with...“ ausgewählt wird.

### **Mehrteiligen Trigger definieren:**

Der Triggerzustand kann mehrere Teile umfassen, die sich durch ein logisches `and` oder ein logisches `or` kombinieren lassen. Um einen mehrteiligen Trigger zu definieren, gehen Sie wie folgt vor:

- Öffnen Sie das Dialogfenster „Define display trigger condition“.



- Wählen Sie einen Triggermodus aus, indem Sie auf die Option **Analogue channels** oder **Bit channels** klicken.  
Triggerzustände können entweder nur für Analog- oder nur für Bit-Kanäle definiert werden.
- Definieren Sie den ersten Teil der Triggerbedingung.  
Das Vorgehen ist dasselbe wie für einen einfachen Trigger (s. Seite 719).
- Wählen Sie im Kombikästchen „Combina- tion“ als Verknüpfung & (and) oder | (or).
- Definieren Sie den zweiten Teil der Triggerbedingung.  
Wenn Sie weitere Teile anfügen wollen, wiederholen Sie die nötigen Schritte.
- Klicken Sie **OK**.  
Der Triggerzustand ist nun definiert. Er ist automatisch aktiviert und wird verwendet, wenn Sie die nächste Messung starten.

Da das Oszilloskop die Werte zwischenspeichert, auch wenn der Triggerzustand noch nicht eingetreten ist, können Werte für einen einstellbaren Zeitraum (Vortriggerzeit) vor dem Triggerzeitpunkt nachträglich angezeigt werden. Die NachtrIGGERzeit bestimmt die Dauer, für die die Werte angezeigt werden, nachdem das Triggerereignis abgelaufen ist. Die Vor- und NachtrIGGERzeit werden auf die komplette Zeitachsenlänge des Oszilloskopfensters zusammengefasst. Wurden zum Beispiel für die Zeitachsenlänge des Oszilloskopfensters 2 Sekunden eingegeben und ist das Verhältnis zwischen Vor- und NachtrIGGERzeit 0,4/0,6, beträgt die Vortriggerzeit 0,8 Sekunden und die NachtrIGGERzeit 1,2 Sekunden.

### **Vor-/NachtrIGGERzeit einstellen:**

---

Die Vor- und NachtrIGGERzeiten können nur im Erfassungsmodus bei angehaltener Messung bearbeitet werden. Um die Vor- und NachtrIGGERzeit einzustellen, gehen Sie wie folgt vor:

- Öffnen Sie das Dialogfenster „Define display trigger condition“.
- Definieren Sie einen Triggerzustand.

- Passen Sie Vor- und Nachtriggerzeit an, indem Sie den Schieberegler **Ratio between pre- and posttrigger time** verstellen,

*oder*

- geben Sie die Vortriggerzeit in Sekunden im Feld „Pretrigger [s]“ unter dem Schieberegler direkt ein.

Die Vortriggerzeit kann maximal die Länge der Zeitachse im Oszilloskopfenster betragen. Die Nachtriggerzeit wird automatisch berechnet, sie ist die Differenz aus der Länge der Zeitachse und der Vortriggerzeit.

- Aktivieren Sie die Option **Enable after posttrigger again**, wenn die Triggerung nach Ablauf der Nachtriggerzeit wieder neu beginnen soll.
- Klicken Sie **OK**.

Ihre Einstellungen werden übernommen.

Sobald Sie einen Triggerzustand festgelegt haben, wird dieser automatisch aktiviert. Er wird verwendet, sobald das Experiment beim nächsten Mal gestartet wird. Sie können den Trigger selbst aktivieren, deaktivieren und manuell auslösen.

### **Trigger aktivieren/deaktivieren:**

---

Sie können einen Trigger nur im Erfassungsmodus bei angehaltener Messung aktivieren oder deaktivieren. Um einen Trigger zu aktivieren, gehen Sie wie folgt vor:

- Wählen Sie im Oszilloskopfenster das Menü **Edit**.  
Der Trigger ist aktiviert, wenn neben der Menüfunktion **Activate trigger** ein Häkchen ist. Er wird verwendet, sobald Sie die Messung starten.
- Wählen Sie die Menüfunktion **Activate trigger**, um den Trigger zu deaktivieren.
- Wählen Sie erneut **Edit** → **Activate trigger**, um den Trigger wieder zu aktivieren.

### Trigger manuell auslösen:

---

Die manuelle Aktivierung eines Triggers ist nur im Erfassungsmodus bei laufender Messung möglich. Um einen Trigger manuell zu aktivieren, gehen Sie wie folgt vor.

- Wählen Sie **Edit** → **Trigger manually**.  
Dies hat denselben Effekt wie das Erreichen des Triggerzustandes.

Ein Oszilloskopfenster kann in die Zwischenablage kopiert und somit in andere Anwendungen eingefügt werden. Weiterhin besteht die Möglichkeit ein Oszilloskopfenster auszudrucken.

### Oszilloskopfenster kopieren:

---

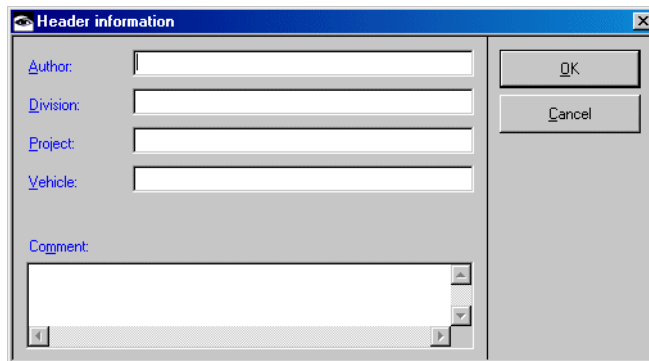
- Halten Sie das Experiment an.
- Markieren Sie das Oszilloskopfenster, das Sie in die Zwischenablage kopieren wollen.
- Wählen Sie **File** → **Copy To Clipboard**.  
Das Oszilloskopfenster wird in die Zwischenablage kopiert.

### Drucken der Oszilloskopausgabe:

---

Um die Anzeigefläche „Signals“ auszudrucken, gehen Sie wie folgt vor:

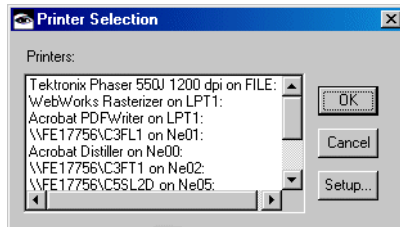
- Halten Sie das Experiment an.
- Wählen Sie **File** → **Print**.  
Das Fenster „Header information“ öffnet sich.



The image shows a dialog box titled "Header information" with a standard Windows-style title bar (minimize, maximize, close buttons). The dialog is divided into two main sections. The left section contains four single-line text input fields labeled "Author:", "Division:", "Project:", and "Vehicle:". Below these is a larger text area labeled "Comment:" with a vertical scrollbar on the right and horizontal scrollbars at the top and bottom. The right section of the dialog contains two buttons: "OK" and "Cancel", stacked vertically.

- Tragen Sie die notwendigen Angaben zu Verfasser, Abteilung, Projekt und Fahrzeug in die entsprechenden Felder „Author“, „Division“, „Project“ und „Vehicle“ ein.
- Geben Sie weitere Kommentare im Feld „Comment“ ein.
- Klicken Sie **OK**.

Ihre Einstellungen werden übernommen und das Fenster „Printer Selection“ öffnet sich.



- Wählen Sie einen Drucker und klicken Sie auf **OK**.

Die Fläche „Signals“ wird auf dem ausgewählten Drucker ausgedruckt.

Die Listen „Measure channels“ und „Bit channels“ werden nicht mit ausgedruckt.

Sie können die Messdaten im Dateisystem entweder im MDF- oder FAMOS-Format speichern. Das Dateiformat ist standardisiert und kann von anderen ETAS-Produkten gelesen werden.

Aufgrund des Multitasking-Verhaltens von Windows kann es möglich sein, dass der in der Datei gespeicherte Datensatz während eines Offline-Experiments nicht immer vollständig ist. Verwenden Sie in einem Online-Experiment die Datenprotokollfunktion (siehe Abschnitt „Der Data-Logger“ auf Seite 641) um die volle Genauigkeit zu erreichen.

### **Speichern von Oszilloskopdaten in einer Datei:**

- Halten Sie das Experiment an.
- Wählen Sie den oder die Kanäle aus, die Sie speichern wollen.

- Wählen Sie **File** → **Save Selected Channels** → **<format>**.

Sie haben die Wahl zwischen den Formaten MDF und FAMOS.

*Oder*

- Wählen Sie **File** → **Save All Channels** → **<format>**, um den Inhalt aller Kanäle im Oszilloskopfenster zu speichern.

Das Dialogfenster „Store measure data“ wird geöffnet.

- Wählen Sie einen Pfad und einen Dateinamen für die Ausgabedatei aus.

- Klicken Sie auf **Öffnen**.

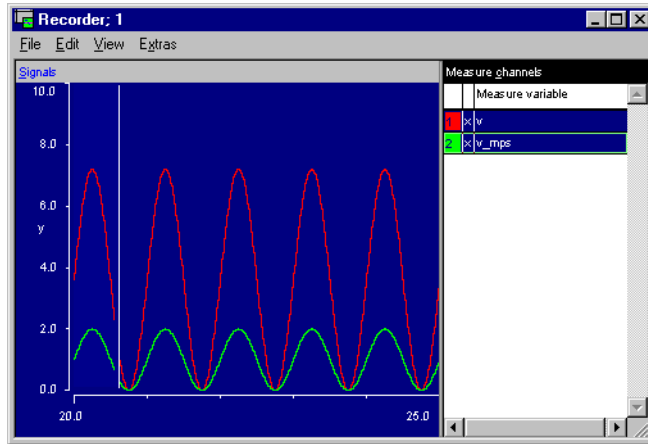
Die Daten der Kanäle im aktuellen Oszilloskopfenster werden im gewählten Format in der ausgewählten Datei gespeichert.

### *Recorder*

---

Der Recorder funktioniert ähnlich wie das Oszilloskop. Der Hauptunterschied besteht darin, dass der Display-Bereich zwischen Durchläufen nicht aktualisiert wird. Das Oszilloskop ist im vorangegangenen Abschnitt beschrieben. In einem Oszilloskopfenster wird der Inhalt des Display-Bereichs jedesmal gelöscht, wenn die Ausgabekurven die rechte Seite des Fensters erreichen.

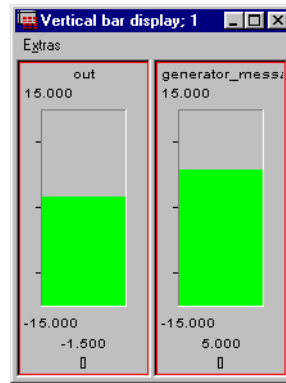
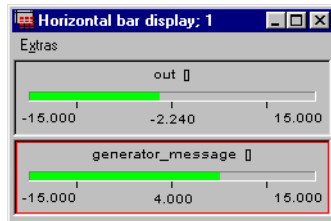
In einem Recorder-Fenster ist dies nicht der Fall; die Ausgabekurven bleiben auf dem Bildschirm, bis sie ausdrücklich überschrieben werden. Ein Cursor in der Form einer weißen senkrechten Linie zeigt an, bis zu welchem Punkt die Kurven überschrieben wurden; die Kurven rechts der Linie gehören zum vorangegangenen Durchlauf.



Der einzige andere Unterschied zwischen dem Recorder und dem Oszilloskop besteht darin, dass Sie sich im Display-Bereich eines Recorder-Fensters kein Raster anzeigen lassen können, da dieses Gitternetz mit dem Cursor mitlaufen müsste. Die Bildschirmoptionen, Messdatenanalyse und Triggerfunktion funktionieren wie im Oszilloskop. Für die Analyse sind allerdings nur die Teile der Kurven zugänglich, die zum gegenwärtigen Durchlauf gehören, der Bereich rechts der weißen Linie kann nicht analysiert werden.

## Horizontale und Vertikale Balkenanzeige

Die Balkenanzeige zeigt Messwerte als farbige Balken. Die Ober- und Untergrenze der Anzeige können Sie frei definieren. Fällt der Messwert unter die eingestellte Untergrenze, bleibt die Anzeige leer. In der Mitte unter dem Balken wird der Messwert zusätzlich numerisch angezeigt. Die vertikale Balkenanzeige kann nur in der Offline-Experimentierumgebung gewählt werden.

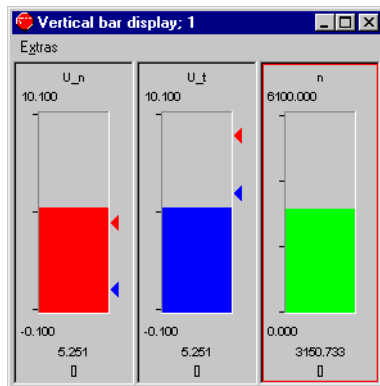


### Einrichten der Balkenanzeige:

- Wählen Sie die Balkenanzeige aus, die Sie einrichten wollen.
- Sie können auch mehrere Balkenanzeigen gleichzeitig anwählen, indem Sie bei gehaltenen <CTRL>-Taste nacheinander darauf klicken.
- Wählen Sie **Extras** → **Setup**.  
Folgende Dialogbox erscheint:

- Wählen Sie im Feld „Value Decimals“ die Zahl der Nachkommastellen für die Anzeige des numerischen Wertes.
- Stellen Sie in den Feldern „Min“ und „Max“ den Anzeigebereich ein.
- Geben Sie in den Feldern „Lower“ und „Upper“ unter „Monitoring bounds“ eine oder zwei Überwachungsgrenzen.

Die Überwachungsgrenzen sind in der Balken-  
anzeige durch farbige Dreiecke markiert. Fällt  
der Messwert unter die untere Grenze, wech-  
selt die Balkenfarbe nach Blau. Steigt der  
Messwert über die obere Grenze, wechselt die  
Balkenfarbe nach Rot. Bleibt der Messwert im  
Bereich dazwischen, ist die Balkenfarbe grün.



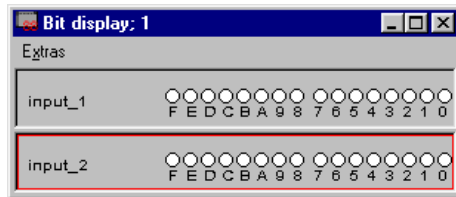
- Klicken Sie auf **OK**, um die Einstellungen zu übernehmen.



## Bitanzeige

---

Die Bitanzeige zeigt Messwerte als Bit-Array an. Sie eignet sich besonders für die Darstellung binärer Kanäle sowie allgemein zum raschen Ablesen von Dualzahlwerten bei angehaltener Messung.



Die Bitanzeige kann nicht eingerichtet werden. Sie können Kanäle kopieren (Seite 701), verschieben (Seite 702) und löschen (Seite 702), Attribute mit andere Messfenstern austauschen (Seite 703), den Titel des Fensters ändern (Seite 704) und Informationen über die Kanäle anzeigen (Seite 705).

### Kanäle im Fenster bewegen:

---

- Wählen Sie **Extras** → **Move** → **Up**, um den ausgewählten Messkanal in der Bitanzeige nach oben zu bewegen.
- Wählen Sie **Extras** → **Move** → **Down**, um den ausgewählten Messkanal nach unten zu bewegen.

Diese Befehle stehen nur zur Verfügung, wenn das Fenster mehr als einen Kanal enthält.

## Monitor

---

Monitore sind eine einfache Möglichkeit zur Ansicht numerischer und logischer Werte innerhalb eines Blockdiagramms. Sie können besonders für die Beobachtung, wie sich verschiedene Werte gegenseitig beeinflussen, in komplizierten Diagrammen mit vielen Elementen von Nutzen sein.

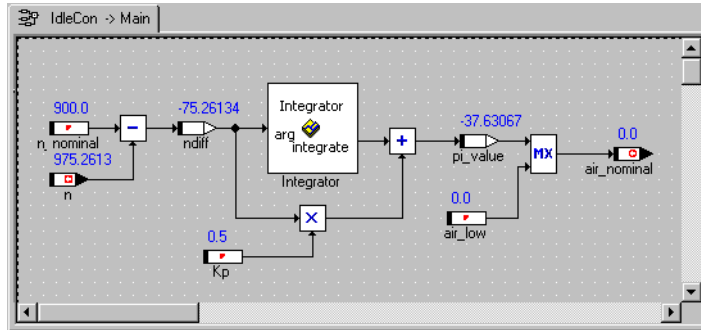
### Überwachen eines Elements:

---

- Klicken Sie im Zeichenbereich der Offline-Experimentierumgebung mit der rechten Maustaste auf das Element, das Sie sich ansehen möchten.

- Wählen Sie **Monitor** aus dem Kontextmenü aus.

Wenn Sie das Experiment starten, wird der Wert im Fenster „Physical Experiment“ über dem Element angezeigt.



- Wählen Sie den Befehl **Monitor** erneut, um die Bildschirmzuordnung aufzuheben.
- Wählen Sie **View** → **Monitor All**, um allen Elementen Monitore zuzuweisen.
- Wählen Sie **View** → **Delete Monitors**, um alle Monitore im Diagramm zu löschen.

## 7 Automatische Dokumentation

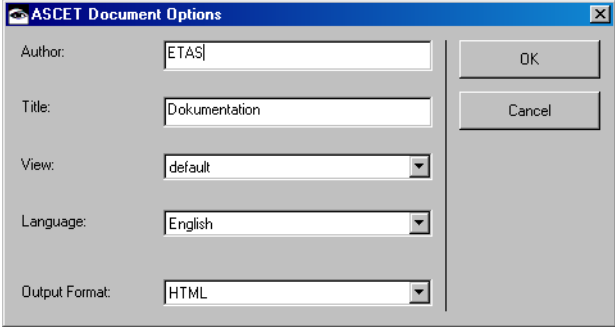
ASCET kann für jeden Ordner oder Datenbankeintrag automatisch eine Dokumentationsdatei erzeugen. Die Dokumentationsdatei enthält Informationen zum dokumentierten Objekt, z. B. zur Schnittstelle und zu Spezifikationsdiagrammen, sowie eventuelle vom Benutzer hinzugefügte Hinweise. Dokumentation kann in RTF-, HTML-, ASCII- oder Postscript-Format erzeugt und dann ausgedruckt oder innerhalb anderer Dokumente verwendet werden.

### 7.1 Erzeugen von Dokumentation

Ehe sich die Dokumentation erzeugen lässt, müssen die zu dokumentierenden Ordner oder Datenbankeinträge zum Inhalt der Dokumentation hinzugefügt werden. Dokumentation kann für einen oder mehrere Gesamtordner oder einfach für einen oder mehrere Datenbankeinträge erzeugt werden.

#### Einstellen der Optionen für die Dokumentationserzeugung:

- Wählen Sie im Komponentenmanager **Tools** → **Documentation** → **Options** zum Öffnen des Dialogfenster „ASCET Document Options“.



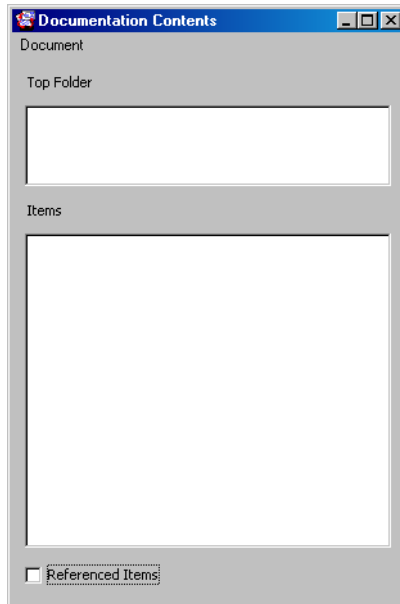
- Geben Sie Ihren Namen in das Feld „Author“ und einen Titel für das erzeugte Dokument in das Feld „Title“ ein.
- Wählen Sie die Sicht, für die die Dokumentation erstellt werden soll, aus dem Kombikästchen „View“ aus (Details siehe Kapitel 7.3).

- Wählen Sie die Sprache für die Dokumentationsdatei aus dem Kombikästchen „Language“ aus.  
Sie können Dokumentation in Englisch oder in Deutsch erzeugen.
- Wählen Sie das Dokumentenformat aus dem Kombikästchen „Output Format“ aus.  
Die Dokumentationsdatei kann im Format ASCII, RTF, HTML oder Postscript erzeugt werden.
- Klicken Sie auf **OK**.

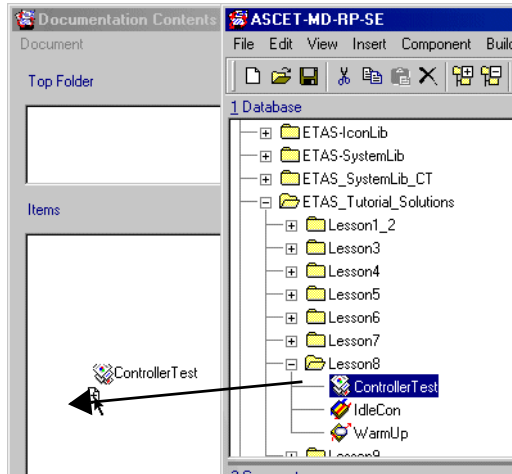
#### **Auswählen der Einträge zum Erzeugen von Dokumentation:**

---

- Wählen Sie im Komponentenmanager **Tools** → **Documentation** → **Contents**, um das Fenster „Documentation Contents“ zu öffnen.



- Aktivieren Sie die Option **Referenced Items** im unteren Bereich des Fensters, damit beim Hinzufügen neuer Einträge referenzierte Einträge in den Dokumentationsinhalt mit aufgenommen werden.
- Ziehen Sie den Eintrag oder den Ordner aus dem Komponentenmanager in das Fenster „Documentation Contents“, Fläche „Items“.



Wenn Sie einen Ordner in das Fenster ziehen, werden alle Objekte in der Fläche „Items“ angezeigt; der Name des übergeordneten Ordners wird in der oberen Fläche angezeigt.

### Ändern der getroffenen Auswahl:

- Öffnen Sie das Fenster „Documentation Contents“.
- Wählen Sie in der Fläche „Items“ einen Eintrag aus.
- Klicken Sie mit der rechten Maustaste auf den Eintrag und wählen Sie aus dem Kontextmenü **Delete**.

Der Eintrag wird aus der Fläche „Items“ entfernt, d.h. er wird beim Erzeugen von Dokumentation nicht mit aufgenommen.

- Wählen Sie **Move Up** oder **Move Down**, um die Position des Objekts in der Fläche „Items“ zu ändern.

Die Position bestimmt die Reihenfolge, in der Einträge in der erzeugten Dokumentation erscheinen.

Sie können den Inhalt von erzeugter Dokumentation als Liste von Datenbankeinträgen und deren Reihenfolge in einer Datei speichern.

#### **Speichern/Laden des Dokumentationsinhalts:**

---

- Wählen Sie im Fenster „Documentation Contents“ **Document** → **Save Document Items To File**.

Ein Dateiauswahlfenster öffnet sich. Die Dateierweiterung \*.cfg ist fest vorgegeben.

- Wählen Sie den Pfad und den Dateinamen für die Inhaltsdatei aus und klicken sie auf **Öffnen**.

Die Datei wird am angegebenen Ort abgelegt.

- Wählen Sie **Document** → **Load Document Items From File**, um einen gespeicherten Dokumentationsinhalt zu laden.

#### **Erzeugen der Dokumentation:**

---

- Wählen Sie im Fenster „Documentation Contents“ **Document** → **Generate Document**.

Ein Benennungsfenster öffnet sich.

- Geben Sie einen Dateinamen ein und klicken Sie auf **OK**.

Die Dokumentation wird für alle im Fenster „Documentation Contents“ angezeigten Einträge erzeugt.

Die erzeugten Dateien werden im Dokumentationsverzeichnis Ihrer ASCET-Installation gespeichert. Dieses Verzeichnis legen Sie im Fenster „Options“, Knoten „Documentation“, fest. (siehe Kapitel 2.2.5 auf Seite 55). Bei jedem Erzeugen von Dokumentation werden ältere, gleichnamige Dateien überschrieben.

## Ansehen der erzeugten Dokumentation:

---

- Wählen Sie im Fenster „Documentation Contents“ **Document** → **Preview Document**.

Der Betrachter für den erzeugten Dokumentationsstyp öffnet sich mit der erzeugten Datei.

## 7.2 Ausgabeformate für Dokumentationsdateien

---

Alle von Ihnen erzeugten Dokumentationen werden in das Dokumentationsverzeichnis geschrieben. Dieses Verzeichnis stellen Sie in den ASCET-Optionen ein (s. „Optionen für die automatische Dokumentation“ auf Seite 55). ASCET verwendet standardmäßig für die Dokumentationsdateien immer die gleichen Dateinamen; eventuelle im Verzeichnis vorhandene alte Dateien werden überschrieben.

### *ASCII-Format*

---

Haben Sie ASCII als Ausgabeformat ausgewählt, wird der gesamte erzeugte Text in eine Datei mit der Bezeichnung `text.txt` geschrieben. Enthält der dokumentierte Datenbankeintrag Diagramme, werden diese in Postscript-Dateien geschrieben, und zwar jeweils eine Datei für jedes Diagramm. Die Textdatei enthält den Dateinamen jedes Diagramms. Enthält die Klasse C-Code, wird dieser in die Textdatei aufgenommen.

Sie können den von ASCET erzeugten ASCII-Text in einen beliebigen Editor oder ein beliebiges Textverarbeitungssystem laden und ihn als Grundlage für Ihre eigene Dokumentation von mittels ASCET entwickelten Anwendungen verwenden.

### *RTF-Format*

---

RTF ist ein Standardformat zum Austausch formatierter Dokumente. Es kann von den meisten Textverarbeitungsprogrammen wie z. B. Microsoft Word oder WordPerfect gelesen werden. Die formatierte Dokumentation kann bearbeitet oder in andere Dokumente eingebaut werden. Das Dokument wird in eine Datei mit der Bezeichnung `docu.rtf` geschrieben.

### *HTML-Format*

---

HTML ist das Standardformat für den Datenaustausch im World Wide Web. HTML-Dokumente kann man mit einem beliebigen Internet-Browser wie z. B. Netscape oder Internet Explorer ansehen. Der HTML-Ursprungstext des Dokuments wird in eine Datei mit der Bezeichnung `docu.htm` geschrieben. Alle Diagramme werden im GIF-Format gespeichert, bei dem es sich ebenfalls um ein Standardformat im World Wide Web handelt.

## *Postscript-Format*

---

Haben Sie als Ausgabeformat Postscript gewählt, wird alles in eine Postscript-datei geschrieben; die Diagramme werden innerhalb des Textes positioniert. Die Ausgabedatei hat die Bezeichnung `docmain.ps`.

## 7.3 Sichten

---

Im Blockdiagrammeditor und im Zustandsautomaten-Editor von ASCET existieren sogenannte *Sichten* (engl. *views*). Automatisch erzeugte Dokumentation und Diagramme können in verschiedenen Sichten durch Ausblenden von Teilinformationen unterschiedlich gestaltet werden.

Die Sichten können individuell eingestellt werden. Die Einstellmöglichkeiten umfassen globale und elementbezogene Optionen für die Darstellung im Blockdiagramm-/Zustandsautomaten-Editor und die automatische Dokumentation.

Global können folgende Informationen ein- oder ausgeschaltet werden:

- Sequenzaufrufe und Verbinder
- Methodennamen bei methodenlokalen Elementen, Prozessnamen bei prozesslokalen Elementen
- Grafische Kommentare
- Für verschiedene Elementgruppen kann eine Standard-Darstellungsweise für jede Sicht definiert werden (z.B. alle Parameter vom Typ `cont` als Linie).

Die elementbezogenen Einstellungen sind im Abschnitt „Bearbeiten der Sichten eines Diagrammelements:“ auf Seite 238 beschrieben.

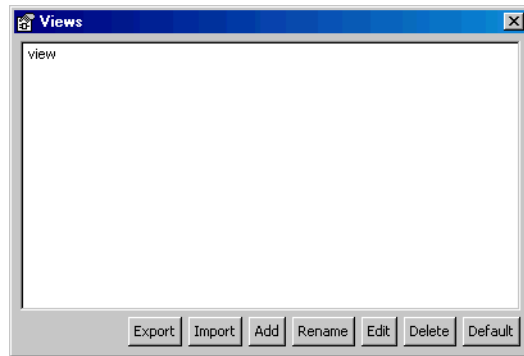
Bei Auswahl einer Sicht werden alle elementbezogenen und globalen Einstellungen wirksam. Die Darstellung eines Diagramms im Editor entspricht der Darstellung des Diagramms in der Dokumentation.



### Erzeugen und Verwalten von Sichten:

---

- Wählen Sie im Komponentenmanager **Tools** → **Views**.  
Das Fenster „Views“ wird geöffnet.



Es enthält zunächst nur eine Sicht mit der Benennung `view`.

- Klicken Sie auf die Schaltfläche **Add**.  
Eine neue Sicht wird hinzugefügt.
- Geben Sie einen Namen für die Sicht ein und drücken Sie `<Enter>`.
- Klicken Sie auf **Rename**, wenn Sie eine ausgewählte Sicht umbenennen wollen.
- Klicken Sie auf **Delete**, um eine ausgewählte Sicht zu löschen.

#### Hinweis

Die Schaltflächen **Add**, **Rename**, **Edit**, **Delete** und **Default** sind auch als Kontextmenü verfügbar.

- Klicken Sie auf **OK**, um das Fenster zu schließen und die Änderungen zu übernehmen.

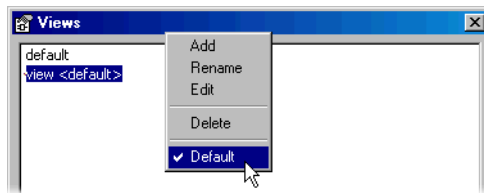
- Klicken Sie auf **Cancel**, um das Fenster zu schließen, ohne die Änderungen zu übernehmen.

### Auswählen einer Standardsicht:

Sie können eine der Sichten als Standard auswählen. Wenn Sie anschließend eine Komponente in einem der Blockdiagrammeditoren öffnen, wird sie in der Standardsicht dargestellt.

- Wählen Sie im Fenster „Views“, eine der vorhandenen Sichten aus.
- Klicken Sie auf die Schaltfläche **Default**  
oder

- wählen Sie aus dem Kontextmenü **Default**.  
Die ausgewählte Sicht wird als Standard markiert; das Kontextmenü **Default** wird für diese Sicht mit einem Haken markiert.



- Um die Auswahl als Standard aufzuheben, gehen Sie wie folgt vor:
  - Wählen Sie die Standardsicht.
  - Klicken Sie erneut auf **Default**.  
Die Auswahl als Standard wird aufgehoben und die Markierung entfernt. Nun ist keine Sicht als Standard gewählt.
- Um eine andere Sicht als Standard zu wählen, gehen Sie wie folgt vor:
  - Wählen Sie die neue Sicht aus.
  - Klicken Sie auf **Default**.  
Die Markierung der alten Standardsicht wird entfernt und die neue Sicht wird als Standard ausgewählt.

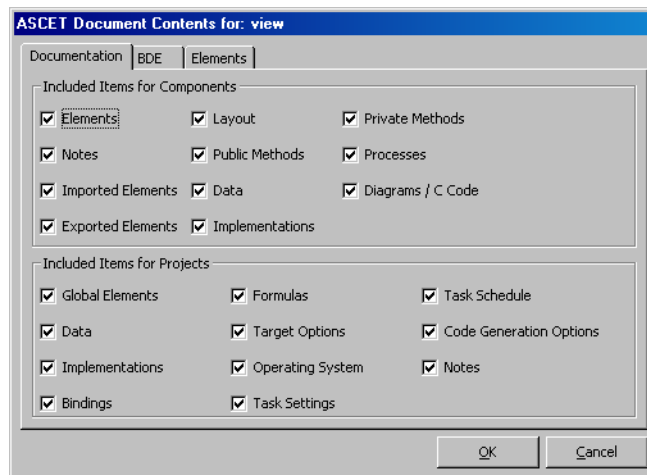
Die Auswahl der Sicht im Zustandsautomateneditor oder Blockdiagrammeditor (siehe Seite 193) bleibt von der Auswahl einer Standardsicht unberührt.

### Bearbeiten einer Sicht:

Sie legen fest, welche Informationen in eine bestimmte Sicht mit aufgenommen werden sollen.

- Wählen Sie im Fenster „Views“, die Sicht, die bearbeitet werden soll.
- Klicken Sie auf die Schaltfläche **Edit**.

Das Dialogfenster „ASCET Document Contents for“ wird geöffnet. Es enthält drei Register, eins für den Inhalt der Dokumentation, eins für die Darstellung im Blockdiagramm-/ Zustandsautomaten-Editor und eins für die Darstellung bestimmter Elemente.



- Stellen Sie in den Registern die Optionen für die bearbeitete Sicht ein.

Die Bedeutung der Optionen ist in den Abschnitten „Optionen für die Dokumentation“, „Optionen für die Editoren“ und „Optionen für Elementtypen“ beschrieben.

- Klicken sie auf **OK**.

Die Sichten, die Sie so eingerichtet haben, werden beim Export von Datenbankeinträgen nicht mit exportiert. Stattdessen können Sie aus dem Fenster „Views“ eine oder mehrere Sichten in eine XML-Datei exportieren und zuvor exportierte Sichten importieren.

### Sichten exportieren:

---

- Wählen Sie im Fenster „Views“ eine oder mehrere Sichten.
- Klicken Sie auf die Schaltfläche **Export**.  
Das Fenster „Export File“ öffnet sich. Es zeigt alle XML-Dateien im ASCET-Exportverzeichnis (s. Seite 63).
- Geben Sie Pfad und Namen (mit Erweiterung \*.xml) der Exportdatei ein.
- Klicken Sie auf **Speichern**.  
Die gewählten Sichten werden in die angegebene Datei geschrieben.

Eine solche Exportdatei ist sehr einfach strukturiert. Der Name der Sicht ist im Element <View> enthalten, die Einstellungen in den Registern des Fensters „ASCET Document Contents for“ sind in den Elementen <Components> und <Projects> gespeichert. Dabei entspricht jede Option eines Registers einem Attribut des jeweiligen Elements; die Reihenfolge der Attribute entspricht der Reihenfolge der Optionen erst in der linken, dann in der rechten Spalte des jeweiligen Registers.

```
<Views>
<View name="view">
<DocumentationSettings>
  <Components notes="true"
    layout="true"
    publicMethods="true"
    privateMethods="true"
    processes="true"
    graphic="true"
    elements="true"
    importedElements="true"
    exportedElements="true"
    data="true"
    implementation="true" />
```

```

    <Projects notes="true"
        codegenOptions="true"
        targetOptions="true"
        operatingSystem="true"
        taskSettings="true"
        taskSchedule="true"
        elements="true"
        bindings="true" data="true"
        implementation="true"
        formulas="true" />
</DocumentationSettings>
<BDESettings sequenceCalls="true"
    processNameLocals="false"
    graphicalComments="true" />
<ElementDefaults>
    <AsLine>
        <Element type="Scalar"
            modelType="Continuous"
            kind="Parameter"
            scope="*"
            existence="Non-Virtual"
            dependency="*"
            memory="*"
            calibration="*" />
    </AsLine>
    <Invisible></Invisible>
    <Contour></Contour>
    <HideContents></HideContents>
</ElementDefaults>
</View>
</Views>

```

Nur das Element `<View>` mit dem Namen der Sicht muss vorhanden sein, die anderen Elemente und Attribute sind optional. Fehlende Attribute werden beim Import auf `true` gesetzt.

Wenn Sie mehrere Sichten gleichzeitig exportieren, wird für jede Sicht ein eigenes Element `<View>` angelegt.

## Sichten importieren:

---

Sie können exportierte Sichten wieder importieren. Wenn eine Exportdatei mehrere Sichten erhält, werden *alle* Sichten gemeinsam importiert.

- Klicken Sie im Fenster „Views“ auf die Schaltfläche **Import**.

Eine Warnung erscheint, dass bereits vorhandene Sichten (gleicher Name) beim Import überschrieben werden.

- Bestätigen Sie die Warnung mit **Yes**.

Das Fenster „Import File“ öffnet sich. Es zeigt alle XML-Dateien im ASCET-Exportverzeichnis (s. Seite 65).

- Wählen Sie die Datei aus, die Sie importieren wollen.

- Klicken Sie auf **Öffnen**.

Die Sichten in der gewählten Datei werden in die ASCET-Datenbank importiert.

## *Optionen für die Dokumentation*

---

Das Register „Documentation“ im Fenster „ASCET Document Contents for“ enthält die Einstellmöglichkeiten für den Inhalt der automatischen Dokumentation, oben für Komponenten, unten für Projekte.

### **Optionen für Komponenten:**

Elements	Alle Elemente der Komponente werden angezeigt.
Notes	Die Notizen, die mit der aktuellen Sicht verknüpft sind, werden angezeigt.
Imported Elements	Alle importierten Elemente, d.h. Messages und globale Variable, werden angezeigt.
Exported Elements	Alle exportierten Elemente werden angezeigt.
Layout	Das Blocklayout der Komponente wird angezeigt.
Public Methods	Alle Methoden in einem öffentlichen Diagramm werden angezeigt.
Data	Alle Datensätze der Komponente werden angezeigt.
Implementations	Alle Implementierungen der Komponente werden angezeigt.

Private Methods	Alle Methoden in einem privaten Diagramm werden angezeigt.
Processes	Alle Prozesse der Komponente werden angezeigt (gilt nur für Module).
Diagrams / C Code	Enthält die Komponente Blockdiagramme oder Zustandsautomaten-Diagramme, werden diese angezeigt. Enthält die Komponente C-Code, wird dieser angezeigt.

### **Optionen für Projekte:**

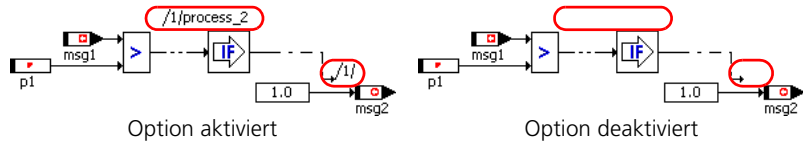
Global Elements	Alle innerhalb des Projekts definierten globalen Elemente werden angezeigt.
Data	Alle Datensätze des Projekts werden angezeigt.
Implementations	Alle Implementierungen des Projekts werden anzeigt.
Bindings	Zeigt an, mit welchen Elementen die innerhalb des Projekts deklarierten globalen Elemente verknüpft sind.
Formulas	Alle im Projekt deklarierten globalen Formeln werden angezeigt.
Target Options	Die aktuellen Target-Einstellungen für das Projekt werden angezeigt.
Operating System	Alle im Betriebssystemeditor vorgenommenen Task- und Prozesseinstellungen werden angezeigt.
Task Settings	Die Einstellungen für jede Task des Projekts werden angezeigt.
Task Schedule	Informationen zum Schedule der Tasks im Projekt werden angezeigt.
Code Generation Options	Die aktuellen Einstellungen für Codegenerierung für das Projekt werden angezeigt.
Notes	Die Notizen, die mit der aktuellen Sicht verknüpft sind, werden angezeigt.

### *Optionen für die Editoren*

Das Register „BDE“ enthält globale Optionen für die Darstellung im Blockdiagramm- und Zustandsautomaten-Editor. Folgende Optionen sind vorhanden:

- **Show Sequence Calls**

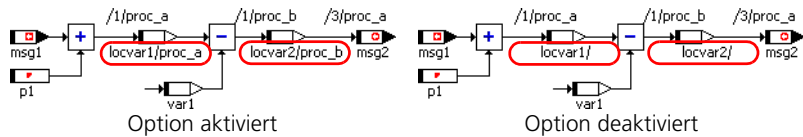
Wird diese Option für eine Sicht deaktiviert, werden in dieser Sicht alle Sequenzaufrufe und Verbinder in den Diagrammen ausgeblendet.



Die Option ist nicht identisch mit den Menüfunktionen **Sequence Calls** → **Show** → \* und **Sequence Calls** → **Hide** → \*. Die Menüfunktionen haben keine Wirkung, wenn **Show Sequence Calls** für die angezeigte Sicht deaktiviert ist.

- **Show Method/Process for Locals**

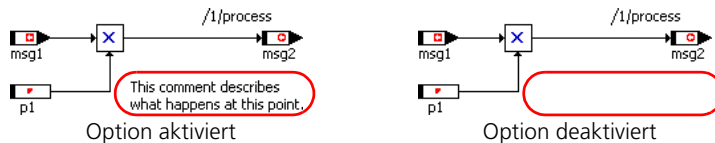
Wird diese Option für eine Sicht deaktiviert, werden die Methodennamen bei methodenlokalen Elementen und die Prozessnamen bei prozesslokalen Elementen ausgeblendet. Das Trennzeichen / bleibt stehen.



- **Show Graphical Comments**

Wird diese Option für eine Sicht deaktiviert, werden alle Kommentare in den Diagrammen ausgeblendet.

Wenn die Option aktiviert ist, werden die Kommentare angezeigt. Wenn für einen oder mehrere Kommentare in den elementspezifischen Optionen die Eigenschaft `Invisible` ausgewählt wurde, werden die betreffenden Kommentare dennoch nicht angezeigt. Mehr zu element-spezifischen Optionen finden Sie in den Abschnitten „Bearbeiten der Sichten eines Diagrammelements:“ auf Seite 238 und „Optionen für Elementtypen“ auf Seite 745.





## Optionen für Elementtypen

---

Im Register „Elements“ stellen Sie die Darstellung verschiedener Elementgruppen im Blockdiagramm-/Zustandsautomaten-Editor zentral ein. Diese globalen Elementoptionen wirken auf alle in der Datenbank vorhandenen Elemente der Gruppe, für die keine individuelle Einstellung vorgenommen wurde.

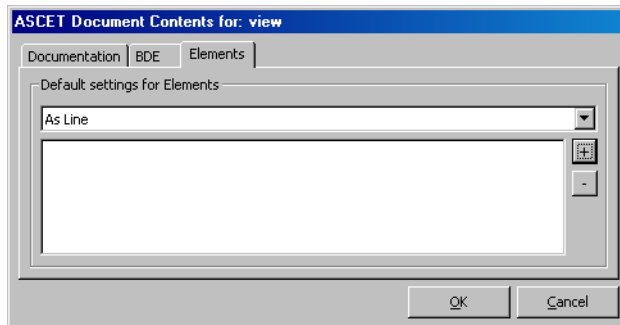
### Hinweis

*Diagrammelemente, für die eine andere individuelle Einstellung als Normal gewählt wurde (siehe „Aussehen von Diagrammelementen“, Seite 238), werden von den globalen Einstellungen nicht beeinflusst.*

### Zentrale Einstellungen für Elementgruppen:

---

- Öffnen Sie für eine Sicht das Fenster „ASCET Document Content for“ im Register „Elements“.



- Wählen Sie im Kombikästchen die Darstellung, die Sie für die Elementgruppe verwenden wollen.  
Die Darstellungen sind auf Seite 239 beschrieben.



- Klicken Sie auf die Schaltfläche +, um eine Elementgruppe hinzuzufügen.  
Das Fenster „Element Properties“ öffnet sich.

Element Properties	
Type	Scalar
Model type	*
Kind	*
Scope	*
Attributes	
Existence	*
Dependency	*
Memory	*
Calibration	*

- Wählen Sie Typ, Modelltyp, Art, Geltungsbereich und die verschiedenen Attribute.  
Die Felder hier haben dieselbe Bedeutung wie die gleichnamigen Bereiche im Elementeditor (s. Abb. 4-2 auf Seite 484).  
Wenn keine explizite Auswahl getroffen ist (\* in einem Feld), gilt die im Register „Elements“ gewählte Darstellung für *alle* Auswahlmöglichkeiten.
- Klicken Sie **OK**, um die Einstellungen zu übernehmen und das Fenster zu schließen.  
Die Elementgruppe wird im Fenster „ASCET Document Content“, Register „Elements“, unter dem Kombikästchen angezeigt.
- Klicken Sie **OK**, um die Einstellungen zu übernehmen und das Fenster zu schließen.

Wenn die Elementgruppen für eine Sicht so definiert werden, dass bestimmte Elemente in mehreren Gruppen gehören (z.B. `as Line` für exportierte Systemkonstanten, `hide` für logische Systemkonstanten), ergibt sich die tatsäch-

liche Darstellung aus der Bearbeitungsreihenfolge der Sichtenoptionen in ASCET; sie ist nicht eindeutig bestimmt. Im Beispiel kann eine logische exportierte Systemkonstante als Linie angezeigt oder komplett versteckt werden.

### **Hinweis**

*Achten Sie bei der Definition von Elementgruppen darauf, dass keine Konflikte durch mehrfache Zugehörigkeiten von Elementen zu Elementgruppen entstehen.*

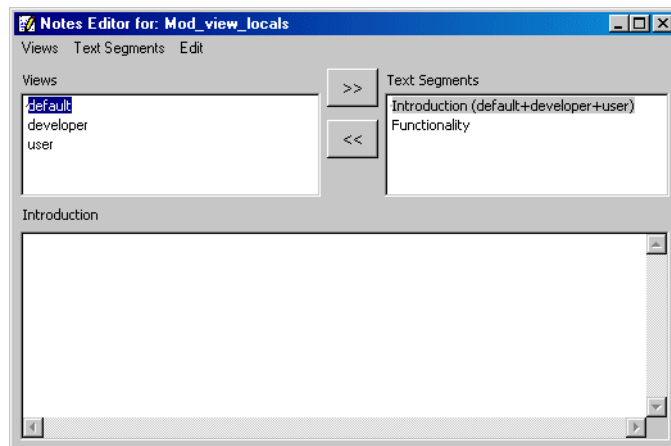
## 7.4 Notizen

Mit jedem Ordner oder Datenbankeintrag können Notizen verknüpft sein. Dabei handelt es sich nicht um Kommentare, die in Blockdiagrammen enthalten sind und mit diesen ausgedruckt werden. Eine Notiz ist mit einem Ordner oder Datenbankeintrag verknüpft und wird Teil des Texts der erzeugten Dokumentation. In Notizen können Informationen mit aufgenommen werden, die nicht Bestandteil der Funktionsspezifikation sind.

### **Erstellen einer Notiz:**

- Wählen Sie einen Ordner oder Datenbankeintrag, zu dem die Notiz gehören soll.
- Wählen Sie **Component** → **Notes** zum Hinzufügen einer neuen Notiz.

Der Notizeneditor öffnet sich.



Eine Notiz besteht aus einer Anzahl von Textsegmenten, die jeweils einer Sicht zugewiesen sind. Ein Textsegment kann mehreren Sichten zugewiesen werden. Damit ist es nicht erforderlich, den gleichen Text für unterschiedliche Sichten jeweils neu zu schreiben. Beim Erzeugen eines Dokuments werden alle zur gewählten Sicht gehörenden Textsegmente mit aufgenommen.

### **Erstellen eines Textsegments:**

---

- Wählen im Notizeneditor Sie **Text Segments** → **Add**.
- Geben Sie einen Namen für das Textsegment ein und drücken Sie <ENTER>. Der Name des Textsegments wird in der Fläche „Text Segments“ aufgelistet. Das neue Textsegment wird allen aktuell ausgewählten Sichten zugewiesen.
- Geben Sie den Text für das Segment in der Texteingabefläche unterhalb der Fläche „Text Segment“ ein. Der Name des aktuellen Textsegments wird als Titel der Texteingabefläche angezeigt.
- Wählen Sie **Text Segment** → **Rename** zum Umbenennen des Textsegments.
- Wählen Sie **Text Segment** → **Delete** zum Löschen des Textsegments.

### **Bearbeiten eines Textsegments:**

---

- Wählen Sie **Edit** → **Cut**, **Copy** oder **Paste** zum Bearbeiten des Textes im Fenster. Diese Operationen funktionieren mit der standardmäßigen Zwischenablage von Windows und können somit zum Austausch von Text mit anderen Anwendungen verwendet werden.
- Wählen Sie **Edit** → **Select All** zum Auswählen des gesamten Texts in der Texteingabefläche.
- Wählen Sie **Edit** → **Read from File** zum Einlesen von Daten aus einer externen Datei. Es wird eine Kopie des Ursprungstexts erstellt; die Ursprungsdatei wird durch diese Operation nicht verändert.

- Wählen Sie **Edit** → **Write to File** zum Schreiben des Texts einer Notiz in eine externe Datei. Sie werden aufgefordert, einen Pfad und einen Dateinamen einzugeben, und die Datei wird an der betreffende Stelle gespeichert.
- Wählen Sie **Edit** → **Save** zum Speichern des aktuellen Textsegments.

### Zuweisen eines Textsegments zu einer Sicht:

---

Ein Textsegment wird bei seiner Erstellung allen in der Fläche „Views“ ausgewählten Sichten zugewiesen.

- Wählen Sie in der Fläche „Text Segment“ ein Textsegment aus.
- Wählen Sie in der Fläche „Views“ eine Sicht aus.
- Wählen Sie **Views** → **Assign Views**

*oder*



- klicken sie auf die Schaltfläche >>. Einem Textsegment können mehrere Sichten zugewiesen werden. Dazu sind diese in der Fläche „Views“ bei gedrückter <CTRL>-Taste auszuwählen.

### Rückgängigmachen der Zuweisung einer Sicht:

---

- Wählen Sie ein Textsegment aus.
- Wählen Sie in der Fläche „Views“ eine Sicht aus.
- Wählen Sie **Text Segment** → **Deassign Views**

*oder*

- klicken Sie auf die Schaltfläche <<.

### Ansehen von Zuweisungen von Textsegmenten:

---

- Wählen Sie in der Fläche „Views“ eine Sicht aus.

- Wählen Sie **Views** → **Select Text Segments**.  
Die der Sicht zugewiesenen Textsegmente werden in der Fläche „Text Segments“ hervorgehoben.
- Wählen Sie in der Fläche „Text Segments“ ein Textsegment aus.
- Wählen Sie **Text Segments** → **Select Assigned Views**.  
Alle Sichten, denen das Textsegments zugewiesen wurde, werden in der Fläche „Views“ hervorgehoben.

---

# Index

## Numerics

- 3D-Kennfeld-Betrachter
  - Rotate 694

## A

- abhängige Parameter
  - Formel bearbeiten 491
  - Formel erstellen 489
- Abschlussaktion definieren 175
- Abtastpunktanzeige 678
- Actions/Conditions-Diagramm
  - Blockdiagramm 305
  - ESDL 306
- Adams-Moulton 364
- Aktion 285
  - als Blockdiagramm 305, 308
  - hinzufügen 307
  - im Übergangseditor 317
  - im Zustandsdiagramm 315
  - im Zustandseditor 316
  - in ESDL 305, 308
  - in separatem Diagramm 305
- Outlining 320
  - spezifizieren 303
  - zuweisen 312
- AMD-Datei 94
  - importieren 99
  - Konsistenzprüfung 103
- AMD-Import
  - automatische Problembhebung 107
  - Besonderheiten 103
  - Fenster "Import Problems" 105
- Anschluss 269
  - eines Elements 220
- Ansicht
  - Standardsicht wählen 738
- Ansichtskonzept 29
  - Arbeiten im ~ 86
  - Daten bearbeiten 89
  - Datenbankeintrag bearbeiten 86
  - Datensatz auswählen 93
  - Element bearbeiten 87
  - Implementierung bearbeiten 90
  - Implementierungssatz auswählen 93
  - Layout bearbeiten 92
  - sortieren 87

- Anzeige
    - kritische Grenzen 706
  - Application mode
    - siehe Betriebsmodus
  - Argument
    - hinzufügen 202
    - Implementierung 542
    - Kommentar 203
  - Arithmetik
    - Festkomma- ~ 399, 443, 455
    - Gleitkomma- ~ 399, 443, 469
    - quantisierte Gleitkomma- ~ 399, 443, 473
  - Arithmetische Dienste 559–595
    - Definition 561
    - Editor (siehe *AS-Editor*) erstellen 568
    - Funktionsdeklaration 565
    - Funktions Schlüssel 561
    - Funktionsweise 560
    - `services.ini` 559
    - speichern 568
    - Verwendung in ASCET 568
    - zulässige ~ 562
    - Zulässige Typen 564
  - Array
    - als Schnittstellenelement 209
    - bearbeiten 497, 675
    - erzeugen 222
    - Rückgabe 209
  - Array-Editor 497, 673
    - Anzeige ändern 676
    - Datenaustausch 676
    - einzelnen Ausgabewert bearbeiten 674
    - mehrere Ausgabewerte bearbeiten 675
  - ASAM-MCD-2MC-Datei
    - importieren 99
  - ASAM-MCD-2MC-Optionen 442
  - ASAM-MCD-2MC-Projekt 78
  - ASCET
    - allgemeine Optionen 37
    - benutzerspezifische Optionen 37
    - einrichten 37
    - Optionen 37
    - Optionen exportieren 40
    - Optionen importieren 40
    - Optionen setzen 39
    - starten 14
  - AS-Editor 572–595
    - Browse-Bereich 579
    - Datei laden 582
    - Editierbereich 580
    - Editoroberfläche 579
    - Eintrag aktualisieren 591
    - Eintrag erstellen 589
    - Eintrag in Kommentar umwandeln 594
    - Eintrag löschen 593
    - Eintrag spezifizieren 590
    - Eintrag suchen 595
    - Einträge verwalten 587
    - Kommentar in Eintrag umwandeln 594
    - Menüfunktionen 575
    - Satz auswählen 584
    - Satz duplizieren 586
    - Satz erstellen 585
    - Satz löschen 585
    - Satz umbenennen 586
    - Sätze verwalten 584
    - Schaltflächen 577
    - starten 574
  - ausführbare Dateien
    - Optionen 69
  - Auto-Inlining 308
    - ein-/auschalten 308
  - Autostartaktion definieren 174
- ## B
- Balkenanzeige 625, 727
    - Einrichten 727
  - Basiselement
    - erzeugen 218
  - Bedingung 285
    - als Blockdiagramm 305, 308
    - hinzufügen 306
    - im Übergangseditor 317
    - im Zustandsdiagramm 315
    - in ESDL 305, 308
    - in separatem Diagramm 305
    - Outlining 320
    - spezifizieren 303



- Bedingungstabelle 377–398
    - Aktion eingeben 394
    - Bedingung eingeben 394
    - Codegenerierung 397
    - Editor 378
    - einrichten 385
    - Element hinzufügen 385
    - Element löschen 392
    - Element zugänglich machen 386
    - erstellen 378
    - experimentieren 397
    - Methoden verwenden 395
    - Spalte entfernen 392
    - Spalte erzeugen (Aktionsbereich) 389
    - Spalte erzeugen (Bedingungsbereich) 388
    - Spalten verschieben 391
    - spezifizieren 393
    - Zeile anlegen 390
    - Zeile löschen 392
    - Zeilen verschieben 391
  - Benutzerauswahl 73
  - benutzerdefinierte Funktion 169
    - Abschlussaktion definieren 175
    - Autostartaktion definieren 174
    - Menüfunktion definieren 170–174
    - Skriptdatei 175–184
  - Benutzeroberfläche
    - AS-Editor 575
    - Blockdiagrammeditor 188
    - Boolescher Tabelleneditor 367
    - C-Code-Editor 329
    - Editor f. Bedingungstabellen 377
    - ESDL-Editor 351
    - Komponentenmanager 15
    - Monitorfenster 150
    - Optionsfenster 38
    - Projekteditor 398
  - Benutzerprofil 73
    - aktivieren 76
    - hinzufügen 74
  - Betriebsmodus 431
    - erstellen 431
    - zuordnen 432
  - Betriebssystem 399
    - Einstellungen 420
  - Betriebssystemeditor 419–432
    - Betriebsmodus 431
    - Hook-Routinen einstellen 428
    - Monitoring 430
    - OSEK-Einstellungen 429
    - Task einrichten 425
  - Bitanzeige 729
  - Block
    - zeitkontinuierlich 356
  - Blockdiagramm
    - analysieren 275
    - bearbeiten 232
    - erzeugen 217
    - Gitter 281
    - navigieren 273
    - neu zeichnen 280
    - Seitenlayout 281
  - Blockdiagrammeditor 186–285
    - Anzeigemodi 188
    - Größe d. Zeichenbereichs 189
    - Menüfunktionen 193
    - Zeichenbereich 217
  - Blockdiagramm-Optionen 56
  - Boolesche Tabelle 367
    - Ein-/Ausgang hinzufügen 372
    - Ein-/Ausgang löschen 373
    - Ein-/Ausgang spezifizieren 371
    - Ein-/Ausgang umbenennen 374
    - erstellen 367
    - experimentieren 376
    - Kombination hinzufügen 372
    - Kombination löschen 373
    - Spalte verschieben 374
    - Standardmatrix 372
    - überprüfen 376
    - Zeile verschieben 374
  - Boolescher Tabelleneditor 367
    - Menüfunktionen 368
  - Build-Optionen 50, 443
    - Generate Dependency Files 444
- ## C
- C-Code 329
    - Deklaration einer Funktion 331
    - Editor 329
    - ersetzen 342
    - externer ~ 348, 349
    - Hauptteil einer Funktion 331

- C-Code 329
  - Komponente spezifizieren 338
  - kopieren (Modul/Klasse) 339
  - kopieren (Projekt) 416
  - suchen 342
  - suchen/ersetzen in Datenbank 118
- C-Code Debugger 636
  - Information ansehen 636
- C-Code-Editor 329–350
  - Menüfunktionen 331
- Codegenerierung
  - ASAM-MCD-2MC-Optionen 442
  - Build-Optionen 443
  - Einstellungen bearbeiten 439
  - Einstellungen laden 441
  - Einstellungen speichern 441
  - Experimentalcode-Optionen 451
  - Optimierung v. Zustands-  
automaten 453
- Codegenerierungsoptionen
  - Integer-Arithmetik 448
  - Optimierungsoptionen 452
- Compileroptionen 62
- Container 78, 476–481
  - Datenbankobjekt
    - aufnehmen 477, 478
  - Netzwerke in ~ konvertieren 481
  - Objekte bearbeiten 480
  - Objekte entfernen 480
  - Objekte umbenennen 479
- CT-Block 356–366
  - Blockdiagramm 357
  - C-Code 359
  - ESDL-Code 360
  - experimentieren 361, 362
  - Integrationsmethode 362
  - spezifizieren 356
  - Zykluszeit 365

## D

- Data-Logger 641–655
  - Aufzeichnung beenden 653
  - Aufzeichnungsoptionen 648
  - Continuous Polling 642
  - Daten aufzeichnen 651
  - Kanäle einrichten 646
  - Label-Liste 647

- öffnen 644
- Periodic Sampling 642
- Periodic to File 642
- Protokolldatei ändern 654
- Transient Sampling 641
- Transient Sampling vorbereiten 643
- Triggerbedingung definieren 652
- Datebankeintrag
  - ASAM-MCD-2MC-Projekt 78
- Daten
  - Austausch 671, 676
  - einlesen 279, 660
  - exportieren 278
  - generieren 610
  - lesen aus externer Datei 660
  - lesen/schreiben aus aktuellem  
Datensatz 657
  - manipulieren 657
  - registrieren 641
  - reinitialisieren 658
  - schreiben 279
  - schreiben in externe Datei 658
  - speichern 657, 659
  - stimulieren 617
- Datenaustausch-Optionen 68
- Datenbank
  - alte ~ verwenden 133
  - anlegen 125
  - C-Code suchen/ersetzen 118
  - durchsuchen 139–146
  - durchsuchen (Fenster "Query") 140
  - durchsuchen (Schaltflächen-  
leiste) 141
  - Eintag löschen 82
  - Eintrag erstellen 79
  - ESDL-Code suchen/ersetzen 118
  - laden 126
  - löschen 130
  - nach ANSI-C umwandeln 137
  - optimieren 131
  - Passwortschutz 148
  - schließen 130
  - Serviceroutinen 138
  - speichern 85, 127
  - speichern unter anderem Namen 128
  - umwandeln (ASCET 4.x) 134
  - umwandeln (vor ASCET-SD V4.0) 137
  - vergleichen 133

- Datenbankeintrag
    - ausschneiden 83
    - bearbeiten 81
    - Container 78
    - Enumeration 78
    - ersetzen 114
    - erzeugen 79
    - exportieren 94
    - importieren 97
    - Komponente 77
    - kopieren 83
    - Notizen bearbeiten 117
    - Projekt 77
    - Referenzen ersetzen 111
    - Signale 77
    - Symbol 77
    - umbenennen 82
    - verschieben 84
    - Zugriffsberechtigung ändern 147
  - Datenbankzugriff 146
  - Datengenerator 609, 616–624
    - automatisch zuweisen 623
    - einrichten 616
    - Kanal 617
    - Kanal einrichten 617
    - Kanal löschen 621
    - Signal definieren 621
    - Signal entfernen 623
    - Signal interpolieren 623
    - Signalwiederholung 624
    - Stimulationsmodus einrichten 618
  - Datensatz 503
    - Daten bearbeiten 507
    - erstellen 505
    - exportieren 509
    - für globale Elemente 511
    - kopieren 505
    - löschen 506
    - Referenz auf 505
    - Standard- ~ 506
    - Unterschiede 509
  - Diagramm 270
    - Actions/Conditions (BDE) 305
    - Actions/Conditions (ESDL) 306
    - Analyse 275, 355
    - drucken 283
    - erzeugen 270
    - laden 271
    - löschen 272
    - mehrere 270
    - navigieren zwischen mehreren ~ 270
    - öffentlich 270
    - privat 270
    - umbenennen 271
  - Diagrammelement 235
    - Aussehen 237
    - ersetzen 235
    - Get- und Set-Anschlüsse 238
    - Sicht 238
    - Verbindung von 220
  - Distribution
    - erzeugen 224
  - Dokumentation 731
    - ansehen 735
    - ASCII-Format 735
    - Dateiformat 735
    - Datenbankeinträge wählen 732
    - erzeugen 734
    - Generierung 731
    - HTML-Format 735
    - Inhalt 739
    - Notiz 747
    - Notiz erstellen 747
    - Optionen 731
    - Postscript 736
    - RTF-Format 735
    - Sicht 736
    - Sicht bearbeiten 739
    - Sicht erzeugen 737
    - Sicht verwalten 737
    - Standardsicht wählen 738
  - Dokumentationsansicht
    - siehe *Sicht*
  - Dokumentationsoptionen 55
- ## E
- Editor
    - Array- ~ 497, 673
    - AS-Editor 572
    - Bedingungstabellen 377, 378
    - Boolesche Tabellen 367
    - C-Code 329
    - Element-~ 482
    - Enumeration-~ 496
    - Grafik- ~ 683, 688, 692
    - Gruppentabellen- ~ 502

- Editor
  - Layout~ 552
  - logischer ~ 495
  - numerischer ~ 495
  - Symbol~ 600
  - Tabellen- ~ 496, 498, 500, 677, 682
  - Verteilungs- ~ 503
- Editor f. Bedingungstabellen 377
  - Menüfunktionen 379
  - Schaltflächen 384
- Editor-Optionen 55
- einbinden
  - externe Quellen 349
- eindimensionale Tabelle 496
- Einschließen
  - komplexer Komponenten 229
- Einstellungen
  - Benutzerauswahl 73
  - Benutzerprofil 73
- Element
  - Anschluss 220
  - Array erzeugen 222
  - Art 486
  - Basis- ~ 218
  - bearbeiten 481
  - Daten 493, 661
  - Dimension 486
  - Distribution erzeugen 224
  - Eigenschaften 481
  - Enumeration einfügen 219
  - Festkennlinie/-feld erzeugen 223
  - freigeben 488
  - Geltungsbereich 486
  - Get-/Set-Port 488
  - Gruppenkennlinie/-feld erzeugen 224
  - Hinweis 285
  - Instanz 482
  - Kennlinie/-feld erzeugen 223
  - komplex 229
  - Matrix erzeugen 222
  - Messfenster zuweisen 625
  - methodenlokal 331
  - Monitor 627
  - quantisiertes Verstellen 474
  - sperrern 488
  - Typ 486
  - verstellen 493, 628, 660, 661
  - Vorkommen 482
- Elementeditor
  - Konfiguration bearbeiten 486
  - öffnen 482
- Enumeration 78
  - einfügen in BDE 219
  - erstellen 80
  - Implementierung 539
- Enumeration-Editor 496, 673
  - Wert bearbeiten 673
- Ereignis
  - abhängiges ~ 615
  - asynchron 612, 614
  - einrichten 610
  - Einzel- ~ 612
  - generieren 609
  - Priorität 612
  - Segment 612
  - Segmentvariable 612
  - stimulieren 611
  - überwachen 637
  - zeitsynchron 612
- Ereignisgenerator 609, 609–616
  - einrichten 609
  - Ereignis einrichten 610
- Ereignisprotokoll 637
- Erscheinungsbild-Optionen 43
- ersetzen
  - C-Code 342
  - C-Code in Datenbank 118
  - C-Code in Datenbank (alle) 124
  - C-Code in Datenbank (einzeln) 122
  - C-Code in Datenbank (Komponente) 124
  - Diagrammelement 235
  - ESDL 354
  - ESDL-Code in Datenbank 118
  - ESDL-Code in Datenbank (alle) 124
  - ESDL-Code in Datenbank (einzeln) 122
  - ESDL-Code in Datenbank (Komponente) 124

- ESDL 351
    - Diagramm analysieren 355
    - Editor 351
    - Klasse spezifizieren 352
    - Modul spezifizieren 355
  - ESDL-Code
    - bearbeiten 354
    - drucken 354
    - ersetzen 354
    - suchen 354
    - suchen/ersetzen in Datenbank 118
  - ESDL-Editor 351–356
  - Euler 364
  - EXECUTE 175
  - Experiment
    - automatisches Beenden 624
    - Offline-- durchführen 629–638
    - Optionen 60
  - Experimentalcode-Optionen 451
    - experimentieren 603–660
      - mit CT-Block 362
      - mit Projekten 469
      - mit quantisiertem Gleitkomma-Code 473
      - mit Zustandsautomaten 326
      - offline 470, 603
      - online 470, 603
  - Experimentierumgebung 603–660
    - Anzeigeoptionen 656
    - aufrufen 607
    - C-Code Debugger 636
    - Data-Logger 641–655
    - Daten manipulieren 657
    - einrichten 607
    - für Offline-Experiment aufrufen 608
    - Implementierung ansehen 635
    - Menüfunktionen 604
    - Mess-System 625–627
    - navigieren 655
    - Offline-Experiment 603
    - Online-Experiment 603
    - Umgebungen 638
    - verlassen 632
    - Verstellungssystem 627
  - exportieren 94
    - AMD 94
    - binär 94
    - Daten 278
    - Datensatz 509
    - Einträge 96
    - Format \*.a21 95
    - Format \*.amd 94
    - Format \*.axl 95
    - Format \*.exp 94
    - Format \*.xml 95
    - Ordner 96
  - Exportoptionen 63
  - externe Optionen 69
    - Beispiel 73
  - externe Werkzeuge
    - Optionen 61
  - Externer Editor 345
    - Modus beende 348
    - öffnen 346
    - schließen 347
- ## F
- Fenster "CodeGen Message Settings" 162
    - Anzeige einrichten 164
    - Einstellungen exportieren 167
    - Hochstufung entfernen 166
    - Informationen hochstufen 166
    - Meldungen ausblenden 165
    - Meldungen einblenden 165
    - Warnungen hochstufen 166
  - feste Tabelle 500
  - Festkennfeld
    - erzeugen 223
  - Festkennlinie
    - erzeugen 223
  - Festkomma-Arithmetik 399, 443, 455
  - FILE 183
  - flexibles Layout
    - s. *Layout aufgenommener Komponenten*
  - FORK 183
  - Formel 456
    - Anzeige filtern 460
    - bearbeiten 456
    - ersetzen 462
    - Fünf-Parameter 458
    - hinzufügen 456
    - Identität 458
    - Import/Export 461
    - linear 458

Formel  
  löschen 459  
  Moebius 458  
  umbenennen 459  
Funktion  
  benutzerdefiniert (siehe *benutzerdefinierte Funktion*)  
Funktionsschlüssel 561

## G

Get-/Set-Ports 209  
  Bedingungstabelle 386  
  Diagrammelement 238  
  Element 488  
Gitter  
  im Blockdiagramm 281  
Gleitkomma-Arithmetik 399, 443, 469  
Grafikeditor (1D) 683  
  Anzeige ändern 686  
  Anzeige einrichten 687  
  starten 683  
  Tabelle bearbeiten 684  
Grafikeditor (2D) 688  
  Anzeige einrichten 691  
  starten 688  
  Tabelle bearbeiten 689  
  zwischen Perspektiven  
    umschalten 690  
Grafikeditor (3D) 692  
  Koordinatensystem drehen 694  
  Netzpunkte markieren/messen 693  
  Select Acces Point 693  
  starten 692  
grundlegende Optionen 42  
  Automatische Sicherung 42  
  Benutzerprofile 43  
  Send E-Mail 43  
Gruppenkennfeld  
  erzeugen 224  
Gruppenkennlinie  
  erzeugen 224  
Gruppentabelle 502  
  Editor 502  
  Verteilung 503

## H

Heun 364

Hierarchie 265  
  Anschluss 268, 269  
  auflösen 266  
  Aussehen 267  
  Block 265  
  grafisch 265  
  navigieren 268  
Hierarchiezustand 296  
  geschlossen 299  
  offen 302  
Hook-Routinen 428  
hybride Projekte 436

## I

if...then im Blockdiagramm 225  
if...then...else im Blockdiagramm 225  
Im 61  
Implementation-Cast  
  Anzeige ein-/ausschalten 264  
  automatisch einfügen  
    (Operator) 261  
  automatisch einfügen  
    (Verbindung) 263  
  Elementeditor 485  
  Implementierung 542  
  in Blockdiagrammen 260–265  
  in ESDL 353  
  Operatorimplementierung in ~  
    konvertieren 549  
Implementierung 399, 455, 513  
  Argument 542  
  Arrays 535  
  bearbeiten 90, 513–542  
  Enumerationen 539  
  Formel wählen 521  
  Handhabung 455  
  Implementation-Cast 542  
  Implementierungstyp zuweisen 531  
  Komponente 514  
  Limiter einstellen 529  
  logische Elemente 537  
  Masterseite einstellen 523  
  Methoden~ 540  
  methodenlokale Variable 533  
  nichtlogische Elemente 518  
  Null ausschließen 528  
  Operator- ~ 543  
  Projekt 514

- Implementierung
    - Prozess-- 540
    - prozesslokale Variable 533
    - Rückgabewert 542
    - Speicherbereich wählen 531
    - spezifizieren (Master Implementa-  
tion) 524
    - spezifizieren (Master Model) 524
    - Standardoptionen 53, 54
    - Tabellen 535
    - temporäre Variablen 535
    - Zero not included 528
  - Implementierungseeditor
    - arithmetische Elemente 520
    - Basiselemente 518
    - Enumerationen 539
    - Kennlinien/-felder 536
    - Komponenten 514
    - logische Elemente 537
    - Methoden 540
    - Prozesse 540
    - skalare Elemente 520
    - Tabellen 536
  - Implementierungsoptionen 53
  - Implementierungstyp 463
    - bearbeiten 465
    - Einstellungen kopieren 467
    - hinzufügen 463
    - zuweisen 531
  - importieren 97
    - AMD-Datei 99
    - ASAM-MCD-2MC-Datei 99
    - aus Exportdatei (**vor** ASCET 4.0) 109
    - aus Exportdateien 99
    - Projekt 108
    - Verbieten von Überschreiben 98
    - Verzeichnisinhalt 102
  - Importoptionen 65
  - Include-Anweisung 331
  - Instanz 230
  - Instanzname 230
  - Integrationsmethode 362
    - Adams-Moulton 364
    - Euler 364
    - Heun 364
    - konfigurieren 363
    - Mulstep 364
    - Runge-Kutta 364
    - Variable-step Calvo 6(5) 364
    - Variable-step Dormand/Prince  
RK5 364
    - Variable-step Dormand/Prince  
RK8 364
    - Variable-step implicit Gear 1 364
    - Variable-step implicit Gear 2 364
    - Variable-step implicit RK2 364
    - Variable-step implicit RK4 364
  - Integrationsoptionen 63
- ## K
- Kanal
    - Daten-- 617
    - Mess- ~ 697
  - Kennfeld 496
    - als Argument 214
    - erzeugen 223
  - Kennlinie 496
    - als Argument 214
    - erzeugen 223
  - Klasse
    - Argument 201
    - Schnittstelle 201
    - spezifizieren in ESDL 352
  - Knoten
    - bearbeiten 292
    - erstellen 292
  - Kommentar 231
  - komplexes Element 229
  - Komponente 77, 275
    - Code 276
    - Codegenerierung 275
    - erzeugen 79
    - ESDL 351
    - in C-Code 338
    - Layout 552
    - Notizen bearbeiten 231
    - Offline-Experiment 277
    - Schnittstelle 201
    - speichern 280
    - Struktur kopieren 85
  - Komponentenmanager 15–149
    - "Problem Report"-Funktion  
einrichten 34
    - Ansichtskonzepte 29
    - ASCET einrichten 37
    - Containeransicht 33

- Komponentenmanager
  - Daten verwalten 76
  - Datenansicht 30
  - Elementansicht 30
  - Elementenbearbeitung 86
  - Enumerationsansicht 33
  - Export 94
  - Implementierungsansicht 31
  - Import 97
  - kontextsensitive Menüfunktionen 27
  - Layoutansicht 32
  - Menüfunktionen 19
  - Ordneransicht 29
  - Problembereich verschicken 36
  - Referenzen aktualisieren 110
  - Schaltflächen 17
  - Symbolgrafiken 18

## L

- Layout 552
  - Anschluss 552
  - Attribute 555
  - bearbeiten 92, 553
  - Gitter 558
  - Standardattribute 556
  - Symbol zuweisen 557
- Layout aufgenommener Komponenten 241–247
  - aktivieren (global) 56
  - aktivieren (lokal) 553
  - Anschlüsse bearbeiten 243
  - Größe bearbeiten 241
  - neues Standard-Layout 246
  - Ports anzeigen/verstecken 244
  - Standard-Layout herstellen 247
- Layout-Editor 552–558
- Link Only 473
- Lizenzoptionen 67
- Logischer Editor 495, 672
  - Wert bearbeiten 672
- lokale Variable
  - hinzufügen 204

## M

- Make-Variablen
  - E\_HOOKS 429

## Matrix

- als Argument 209
- als Schnittstellenelement 209
- erzeugen 222
- Rückgabe 209
- Menüfunktionen
  - AS-Editor 575
  - Blockdiagrammeditor 193
  - Boolescher Tabelleneditor 368
  - C-Code-Editor 331
  - Editor f. Bedingungstabellen 379
  - Experimentierumgebung 604
  - Komponentenmanager 19
  - Messfenster 698
  - Monitorfenster 150
  - Optionsfenster 38
  - Projekteditor 402
  - Verstellfenster 661
  - Zustandsautomaten-Editor 286
- Menüfunktionen definieren 170–174
  - Namen der Fenster 171
  - Variablen 170
- MENUIITEM 182
  - vorhandene Menüfunktion aufrufen 182
- Messdaten
  - analysieren 716
- Messfenster 609, 697–730
  - Anzeige ändern 704
  - arbeiten mit ~ 701
  - Arten 705–730
  - Austausch von Attributen 703
  - auswählen 697
  - Balkenanzeige 727
  - Bitanzeige 729
  - Informationen über Variable 705
  - Kanäle kopieren 701
  - Kanäle löschen 702
  - Kanäle verschieben 702
  - Menüfunktionen 698
  - Monitor 729
  - numerische Anzeige 705
  - Optionen 60
  - Oszilloskop 707–725
  - Recorder 725
  - Titel ändern 704



- Messung 697
  - Balkenanzeige 625
  - Kanal 697
  - Monitor 729
  - numerische Anzeige 625
  - Oszilloskop 625
- Methode
  - Argument 201
  - Argument hinzufügen 202
  - erzeugen 201
  - freigeben 558
  - lokale Variable hinzufügen 204
  - löschen 206
  - öffentlich 201, 270
  - öffentliche ~ im Zustands-
    - automat 323
    - privat 270
  - Rückgabewert 201
  - Rückgabewert hinzufügen 204
  - Schnittstelle bearbeiten 202
  - sperrern 558
- Modul 208
  - Prozess 208
  - Schnittstelle 208
  - spezifizieren in ESDL 355
- Monitor 729
  - Element 627
- Monitorfenster 149
  - Benutzeroberfläche 150
  - Fenster "CodeGen Message
    - Settings" 162
  - Hochstufung entfernen 162, 166
  - Informationen hochstufen 160, 166
  - Kontextmenüs 151
  - Meldungen ausblenden 159, 165
  - Meldungen einblenden 165
  - Meldungen einblenden (alle) 160
  - Meldungen konfigurieren 159, 162
  - Menüfunktionen 150
  - Register „Build“ 156
  - Register „Monitor“ 153
  - Warnungen hochstufen 160, 166
- Monitoring 430
  - cycleStartTime 430
  - cycleTime 430
  - dT 430
  - runtime\_violation 431
  - Zykluszeit 365

- Mulstep 364
- Multi-Editor
  - siehe* Tabelleneditor

## N

- non-volatile 117, 487
- Notiz 747
  - Datenbankeintrag 117
  - erstellen 747
  - Komponente 231
  - Projekt 415
  - Sicht 749
  - Textsegment erstellen 748
  - Textsegment zuweisen 749
- NOWAIT 176
- numerische Anzeige 625, 705
- Numerischer Editor 495, 668
  - quantisiert 474
- numerischer Editor
  - Datenaustausch 671
  - einrichten 668
  - Wert bearbeiten 669, 670

## O

- OBJECT (siehe *Skriptdatei*) 177
  - generatePopWindowCommandFile
    - 178
  - help 177
  - log 177
  - message 177
  - popWindow 178
  - wait 178
  - windows 178
- Objekt
  - Zugriffskontrolle 488
- Objekt-ID
  - anzeigen 178
- Objektzugriff
  - Zugriffskontrolle 488

- Offline-Experiment 603
  - aufrufen 608
  - beenden 631
  - Datengenerator 616–624
  - durchführen 629–638
  - einrichten 609
  - Ereignisgenerator 609–616
  - Ereignisprotokoll 637
  - getakteter Zeitmodus 634
  - Haltebedingung 634
  - schrittweise 632
  - starten 630
  - unterbrechen 632
  - verlassen 632
- Online-Experiment 603
- Operation
  - atomar 254
- Operator
  - anordnen 219
- Operatorimplementierung
  - ansetzen 545
  - automatisch konvertieren 549
  - entfernen (Datenbank) 545
  - entfernen (einzeln) 544
  - entfernen (Komponente) 545
  - Konvertierungsregeln 549
  - suchen 544
- Optionen
  - allgemeine 37
  - ausführbare Dateien 69
  - benutzerspezifische 37
  - Blockdiagramme 56
  - Build-Optionen 50
  - Compileroptionen 62
  - Datenaustausch 68
  - Dokumentationsoptionen 55
  - Erscheinungsbild 43
  - Experiment 60
  - Export 63
  - exportieren 40
  - externe ~ 69
  - für Editoren 55
  - für externe Werkzeuge 61
  - grundlegende Optionen 42
  - Implementierungsoptionen 53
  - Import 65
  - importieren 40
  - Integration 63
  - Knoten "Appearance" 43
  - Knoten "ASCII Editor" 61
  - Knoten "Autofixes" 67
  - Knoten "Block Diagram" 56
  - Knoten "Build" 50
  - Knoten "Calibration" 60
  - Knoten "Colors" 57
  - Knoten "Compiler" 62
  - Knoten "Confirmation Dialogs" 47
  - Knoten "Data Exchange" 68
  - Knoten "Datalogger" 60
  - Knoten "Defaults" 52
  - Knoten "Documentation" 55
  - Knoten "Experiment" 60
  - Knoten "Export" 63
  - Knoten "External Tools" 61
  - Knoten "HexFile" 69
  - Knoten "Implementation" 53
  - Knoten "Import" 65
  - Knoten "Integration" 63
  - Knoten "Licensing" 67
  - Knoten "Measurement" 60
  - Knoten "Options" 42
  - Knoten "Paper Size" 58
  - Knoten "Sequencing" 58
  - Knoten "Statemachine" 59
  - Knoten "Text" 59
  - Lizenz 67
  - Messfenster 60
  - setzen 39
  - Standardoptionen 52
  - Systemstandard wiederherstellen 39
  - Tabelleneditoren 59
  - Texteditoren 59
  - Verstellfenster 60
  - Zustandsautomaten 59
- OS-Editor
  - siehe *Betriebssystemeditor*
- OSEK-Einstellungen 429
  - Autostart-Option 430
  - Deadline-Option 429
  - Min. Period-Optionen 429
- Oszilloskop 625, 707–725
  - Analyse einrichten 717
  - Analysemodus 714
  - Analysemodus starten 714
  - Ausgabe drucken 723
  - automatische Skalierung 711

- Oszilloskop
  - Bitanzeige 708
  - Daten analysieren 716
  - Daten speichern 724
  - einfachen Trigger definieren 719
  - Erfassungsmodus 714
  - Farbeinstellungen 711
  - Fenster einstellen 710
  - Fenster kopieren 723
  - Gitter 710
  - Listen ein-/ausblenden 712
  - Listen einrichten (Analyse) 715
  - Listen einrichten (Erfassung) 713
  - mehrteiligen Trigger definieren 720
  - Messkanäle ein-/ausblenden 712
  - Messkanäle einrichten 708
  - Messkanäle skalieren 711
  - numerische Anzeige 708
  - Oberfläche ändern 708
  - Setup anzeigen 713
  - Tastaturbefehle anzeigen 713
  - Trigger 718
  - Trigger aktivieren 722
  - Trigger deaktivieren 722
  - Trigger manuell auslösen 723
- Outlining 319
  - ein-/ausschalten 320
- P**
- Parameter
  - abhängige ~ 489
- Passwortschutz 148
- Periodic Sampling 642
- Periodic to File 642
- POST 184
- Problem Report 26
  - einrichten 34
  - Problembericht verschicken 36
- Programmstart 14
- Projekt 77, 398
  - ASAM-MCD-2MC-Optionen 442
  - ausführbaren Code generieren 472
  - Build-Optionen 443
  - C-Code kopieren 416
  - Code abspeichern 472
  - Code generieren 471
  - Codegenerierung 399
  - Compiler 444
  - Editor 398
  - Einstellungen bearbeiten 439
  - Einstellungen laden 441
  - Einstellungen speichern 441
  - erstellen 399
  - Experimentalcode-Optionen 451
  - experimentieren 401, 469
  - Formel 456
  - hybrid 436
  - Import 108
  - Kommunikation 416
  - Komponente hinzufügen 414
  - Notizen bearbeiten 415
  - Projektdateien 432
  - Scheduling 419
  - Standard-- für Komponente 401
  - Überwachungsoption 430
- Projektdatei
  - aktualisieren 435
  - hinzufügen 433
  - kopieren 435
  - löschen 434
  - schreiben 434
  - Verwaltung externer ~ 432
- Projekteditor 398–476
  - Implementierungstyp 463
  - Link Only 473
  - Menüfunktionen 402
- Projekteinstellungen 438
  - Knoten "ASAM-2MC" 442
  - Knoten "Build" 443
  - Knoten "Code Generation" 446
  - Knoten "Experiment Code" 451
  - Knoten "Integer Arithmetic" 448
  - Knoten "Optimization" 452
  - Knoten "OS Configuration" 445
  - Knoten "Production Code" 452
  - Knoten "Statemachine" 453
- Prozess 208, 419, 422
- Q**
- quantisierte Gleitkomma-
  - Arithmetik 399, 443, 473
- R**
- Recorder 725
- Redo 233

- Referenz
  - ~ auf Eintrag anzeigen 110
  - ~ auf Eintrag ersetzen 111
  - aktualisieren 110
- Rückgabewert 201
  - Array 209
  - hinzufügen 204
  - Implementierung 542
  - Kommentar 204
  - Matrix 209
- Runge-Kutta 364
  
- S**
- Schaltflächen
  - AS-Editor 577
  - Editor f. Bedingungstabellen 384
  - Komponentenmanager 17
- Scheduling 419
- Schnittstelle 201
  - e. Klasse 201
- SELECTOR 178
  - Objekt-IDs anzeigen 178
  - verfügbare Befehle 179
- SEND 183
- Sequenzaufruf 247
  - automatisch zuweisen 255
  - bearbeiten (einzeln) 248
  - bearbeiten (mehrere) 255
  - bearbeiten (Sequenzeditor) 249
  - geschützte Folge 254
  - Verbinder 259
  - zeigen/verstecken (einzeln) 254
  - zeigen/verstecken (mehrere) 258
  - zurücksetzen (einzeln) 253
  - zurücksetzen (mehrere) 257
- Sequenzeditor 248
- Sequenzierung 248
- Sicht 736
  - bearbeiten 739
  - erzeugen 737
  - exportieren 740
  - importieren 742
  - verwalten 737
- Signal 77
  - als Stimulus 622
  - automatisches Beenden 624
  - betrachten 597
  - definieren 621
  - erzeugen 597
  - interpolieren 623
  - laden 597
  - Messdaten in ~ importieren 598
  - öffnen 597
  - Wiederholung 624
- Skriptdatei 175–184
  - EXECUTE 175
  - FILE 183
  - FORK 183
  - MENUITEM 182
  - NOWAIT 176
  - OBJECT 177
  - POST 184
  - SELECTOR 178
  - SEND 183
- Speicher
  - flüchtig 117, 487
  - nichtflüchtig 117, 487
- Standardoptionen 52
- Standardprojekt 401
- Startzustand 288
- stimulieren
  - Daten 617
  - Ereignis 611
  - Modus `constant` 618
  - Modus `gaussian` 620
  - Modus `matrix` 622
  - Modus `matrix` 620
  - Modus `table` 619
  - Stimulationsmodus einrichten 618
  - Stimulusmatrix 621
  - Zufallsgenerator 619
  - zyklische Modi 619
- Stützstellen
  - verstellen 696
- suchen 342
  - C-Code 342
  - C-Code in Datenbank 118
  - ESDL 354
  - ESDL in Datenbank 118
- Support 34
- `switch` im Blockdiagramm 226
- Symbol 77
  - Editor 600
  - erzeugen 600
  - Format 602
  - in Datei speichern 602

## Symbol

- laden 601
- öffnen 600
- Skalierung 602

## Symbolgrafiken 18

## T

### Tabelle

- Abtastpunkt bearbeiten 678
- Abtastpunkt einfügen 680
- Bedingungs-~ 377
- Boolesche ~ 367
- eindimensional 496
- feste 500
- Gruppentabelle 502
- Interpolationsmodus 499
- zweidimensional 496, 500, 682

### Tabelleneditor 496, 677, 682

- 1D 498
- 2D 500
- Abtastpunktanzeige 678
- Anzeige ändern 681
- einzelne Ausgabewerte  
bearbeiten 680
- mehrere Ausgabewerte  
bearbeiten 680
- Stützstellen verstellen 696

### Tabellenoptionen 59

### Target

- Optionen 438

### Targetoptionen 438

### Task 420, 422

- Alarm-~ 425
- Betriebsmodus zuordnen 432
- einrichten 425
- erstellen 422
- Grundeinstellungen 425
- Hook-Routinen einstellen 428
- Init-~ 425
- Interrupt-~ 425
- OSEK-Einstellungen 429
- Prozess entfernen 422
- Prozess zuordnen 422
- Software-~ 425
- Timetable-~ 425

### temporäre Variable 493

### Textoptionen 59

## Timer

- Periode 427
- Verzögerung 428

## Transient Sampling 641

- vorbereiten 643

## Trigger

- e. Zustandsautomaten 296
- einfügen 296
- Oszilloskop 719, 720

## U

### Übergang 285

- Aktion 310
- Aussehen bearbeiten 295
- Bedingung 310
- herstellen 293
- Priorität 310
- Trigger 310
- Verlauf bearbeiten 294

### Übergangseditor 310, 317

### Überwachungsoption

- s. Monitoring

### Umgebung 638

- exportieren 641
- laden 640
- Optionen 655
- speichern 639
- umschalten 640

### Undo 232

## V

### Variable

- öffentlich 488
- privat 488
- temporäre ~ 493

### Variable-step Calvo 6(5) 364

### Variable-step Dormand/Prince RK5 364

### Variable-step Dormand/Prince RK8 364

### Variable-step implicit Gear 1 364

### Variable-step implicit Gear 2 364

### Variable-step implicit RK2 364

### Variable-step implicit RK4 364

### Verbinder 259

- automatisch zuweisen 260
- erstellen 259

### Verbindung

- von Diagrammelementen 220

- Verstellen
  - Array 497, 673
  - numerisch 668
  - quantisiert 474
  - skalar 668
  - von Stützstellen 696
  - Werte 660
- Verstellfenster 609, 660–697
  - arbeiten mit ~ 665–695
  - Array-Editor 673
  - Enumeration-Editor 673
  - Grafikeditor (1D) 683
  - Grafikeditor (2D) 688
  - Grafikeditor (3D) 692
  - Informationen über Variable 667
  - logischer Editor 672
  - Menü "Axis" 663
  - Menü "Edit" 661
  - Menü "Extras" 664
  - Menü "View" 663
  - Menüfunktionen 661
  - numerischer Editor 668
  - Optionen 60
  - Tabelleneditor 677
  - Tabelleneditor (2D) 682
  - Titel ändern 666
  - Variablen löschen 666
  - Variablen verschieben 665
- Verstellsystem 627
- Verteilungs-Editor 503
- Verwaltung externer Projektdateien 432
- volatile 117, 487

## W

- `while` im Blockdiagramm 228

## Z

- Zeichenbereich
  - Größe einstellen 189
- Zustand 285
  - Aktion zuweisen 312
  - Animation 327
  - anordnen 288
  - bearbeiten 289
  - Hierarchie- ~ 296
  - Übergang herstellen 293
  - umbenennen 290

- Zustandsautomat 285
  - Aktion 285
  - Animate States 327
  - Ausgang 321
  - Auto-Inlining 308
  - Bedingung 285
  - Editor 285
  - Eingang 321
  - erstellen 287
  - experimentieren 326
  - hierarchische Codegenerierung 297
  - Kommentarzeilen einfügen 319
  - öffentliche Methode 323
  - öffentliche Methode verwenden 324
  - Outlining 319, 320
  - Startzustand 288
  - statische Aktion optimieren 304
  - Trigger 296
  - Triggerargument hinzufügen 322
  - Übergang 285
- Zustandsautomaten-Editor 285–329
  - spezielle Menüfunktionen 286
- Zustandsautomaten-Optionen 59
- Zustandsdiagramm 285
- Zustandseditor 313, 316
- zweidimensionale Tabelle 496
- Zykluszeit 365