

Peregrine

# AssetCenter 4.3

---

## Verwendung erweiterter Funktionen

© Copyright 2004 Peregrine Systems, Inc.

Alle Rechte vorbehalten.

Die in diesem Dokument enthaltenen Informationen sind Eigentum von Peregrine Systems Inc. und dürfen ohne vorherige schriftliche Zustimmung von Peregrine Systems Inc. nicht verwendet und weitergegeben werden. Ohne vorherige schriftliche Genehmigung durch Peregrine Systems Inc. ist die Vervielfältigung dieses Handbuchs sowie seiner Teile unzulässig. In der vorliegenden Dokumentation werden zahlreiche Produkte mit ihrem Markennamen bezeichnet. Im Allgemeinen sind diese Marken im Besitz ihrer jeweiligen Eigentümer.

Peregrine Systems und AssetCenter® sind eingetragene Warenzeichen von Peregrine Systems Inc.

Die in diesem Handbuch beschriebene Software wird im Rahmen eines Lizenzvertrags zwischen Peregrine Systems Inc. und dem Endbenutzer zur Verfügung gestellt und muss in Übereinstimmung mit den Bedingungen dieses Vertrags eingesetzt werden. Die in diesem Dokument enthaltenen Informationen können ohne Vorankündigung geändert werden. Aus den vorliegenden Informationen lassen sich keinerlei Verpflichtungen seitens Peregrine Systems Inc. ableiten. Die aktuellste Version dieses Dokuments erhalten Sie auf Anfrage beim Kundendienst von Peregrine Systems Inc.

Die Namen der in diesem Handbuch, in der Demodatenbank und dem Lernprogramm genannten Personen und Firmen sind frei erfunden und dienen lediglich zur Veranschaulichung der Softwarefunktionen. Ähnlichkeiten mit lebenden und toten Personen oder Firmen sind rein zufällig.

Technische Informationen zu diesem Produkt sowie Dokumentationen zu einem für Sie lizenzierten Produkt erhalten Sie beim Kunden-Support von Peregrine Systems Inc. unter folgender E-Mail-Adresse: [support@peregrine.com](mailto:support@peregrine.com).

Ihre Kommentare und Anregungen zu diesem Dokument senden Sie bitte an die Abteilung Technische Dokumentation bei Peregrine Systems ([support@peregrine.com](mailto:support@peregrine.com)).

Die vorliegende Ausgabe betrifft die Version 4.3 des lizenzierten Programms.

AssetCenter

<b>Kapitel 1. Aufzählungen</b> . . . . .	<b>13</b>
Anpassbare Aufzählungen . . . . .	13
Systemaufzählungen . . . . .	15
<b>Kapitel 2. Anlegen von Historien</b> . . . . .	<b>17</b>
Erstellen eines Datensatzes . . . . .	19
Ändern des Felds einer Tabelle oder einer 1-Verknüpfung (Beispiel: Benutzer eines Vermögensgegenstands) . . . . .	19
Hinzufügen einer n-Verknüpfung zu einer anderen Tabelle (Beispiel: durch Verträge abgedeckte Vermögensgegenstände) . . . . .	19
Löschen der n-Verknüpfung zu einer anderen Tabelle . . . . .	20
Ändern der n-Verknüpfung zu einer anderen Tabelle . . . . .	20
Anlegen von Historienzeilen für Merkmale und Merkmalwerte . . . . .	20
<b>Kapitel 3. AQL-Abfragen</b> . . . . .	<b>23</b>
Einführung . . . . .	23
Empfehlungen zum Schreiben von AQL-Abfragen . . . . .	26
Sortiervorgänge und Indizes . . . . .	37
Abfrageeditor . . . . .	40
Syntax der Programmiersprache AQL . . . . .	46
AQL-Funktionen . . . . .	60

Abfragebeispiele . . . . .	65
<b>Kapitel 4. Formulare . . . . .</b>	<b>71</b>
Definition eines Formulars . . . . .	71
Vorkonfigurierte Formulare in der Datenbank mit den Lebenddaten installieren . . . . .	71
Formulare erstellen . . . . .	73
Formulare und Objekte bearbeiten . . . . .	73
Eigenschaften von Formularobjekten . . . . .	76
Eingaberaster . . . . .	79
Formularseiten einrichten . . . . .	79
Einfache Berichte mühelos erstellen . . . . .	80
Formulare eines bestimmten Moduls identifizieren . . . . .	81
Formulare und Aktionsschaltflächen zuordnen . . . . .	81
<b>Kapitel 5. Crystal-Berichte . . . . .</b>	<b>83</b>
Funktionsprinzip und Installation des Berichtgenerators . . . . .	83
Detail eines Berichts . . . . .	87
Crystal Reports-Berichte ändern . . . . .	88
Crystal Reports-Statistiken . . . . .	88
Detailberichte erstellen . . . . .	89
Berichte drucken . . . . .	92
Crystal-Berichte eines bestimmten Moduls identifizieren . . . . .	93
Berichte und Aktionsschaltflächen zuordnen . . . . .	93
<b>Kapitel 6. Aktionen . . . . .</b>	<b>95</b>
Definition einer Aktion . . . . .	95
Aktionen erstellen . . . . .	97
Beispiele für Aktionen . . . . .	107
Arbeiten mit Variablen . . . . .	114
Sonderfall der Sybase SQL Anywhere-Engine . . . . .	115
Aktionen testen . . . . .	115
Aktionen ausführen . . . . .	116
Aktionen und Aktionsschaltflächen zuordnen . . . . .	117
<b>Kapitel 7. Nachrichtensystem . . . . .</b>	<b>119</b>
Allgemeine Funktionsweise des Nachrichtensystems . . . . .	119
<b>Kapitel 8. Workflows . . . . .</b>	<b>123</b>
Definition der Konzepte im Rahmen der Workflow-Modelle . . . . .	123
Allgemeine Funktionsweise . . . . .	127

Für den Workflow hauptsächlich verwendete Tabellen . . . . .	128
Arbeiten mit dem Workflow-Grafikeditor . . . . .	130
Workflow-Modelle einrichten . . . . .	133
Beispiel eines Workflow-Modells für die Bewilligung von Anforderungen . . . . .	133
Kontext eines Workflow-Modells . . . . .	151
Workflow-Rollen . . . . .	153
Workflow-Aktivitäten . . . . .	154
Aufgaben . . . . .	161
Ereignisse . . . . .	164
Workflow-Übergänge . . . . .	176
Warnungen und Fristen innerhalb eines Workflow-Modells . . . . .	176
Workflow-Ausführungsgruppen . . . . .	178
Workflow-Protokoll . . . . .	179
Abgeschlossene Workflow-Instanzen löschen . . . . .	179
<b>Kapitel 9. Export von Daten und Verwaltung von SQL-Ansichten . . . . .</b>	<b>185</b>
Definitionen eines Skripts und einer Exportabfrage . . . . .	185
Daten der AssetCenter-Datenbank exportieren . . . . .	186
SQL-Ansichten der AssetCenter-Datenbank verwalten . . . . .	188
Empfehlungen . . . . .	189
Exportskripte definieren . . . . .	189
Exportskripte ausführen . . . . .	195
<b>Kapitel 10. Skripte . . . . .</b>	<b>199</b>
Definition eines Skripts . . . . .	199
Anwendungsbereiche von Skripten . . . . .	201
Einführung in die Funktionen . . . . .	202
Klassifizierung der BASIC-Funktionen . . . . .	207
Erste Schritte zum Schreiben eines Skripts . . . . .	207
Skriptbibliotheken . . . . .	210
Hinweise und Tipps . . . . .	212
Erstes Beispiel für ein Skript . . . . .	217
Zweites Beispiel für ein Skript . . . . .	220
<b>Kapitel 11. Kalender . . . . .</b>	<b>223</b>
Allgemeine Übersicht und Funktionsweise eines Kalenders . . . . .	223
Auswirkung der Kalender auf bestimmte Funktionalitäten . . . . .	224
Vorgehensweise beim Erstellen von Kalendern . . . . .	224
Beschreibung der Erstellung eines Kalenders . . . . .	225
<b>Kapitel 12. Zeitzonen . . . . .</b>	<b>233</b>
Vorteile der Verwaltung von Zeitzonen . . . . .	233

Zeitzone einrichten . . . . .	234
Zeitzone erstellen . . . . .	235
Zeitzone verwalten . . . . .	235
Zeitzone in AssetCenter Server verwalten . . . . .	242
Auswirkungen der Zeitzone auf verschiedene Vorgänge . . . . .	243
<b>Kapitel 13. Berechnete Felder . . . . .</b>	<b>249</b>
Definition eines berechneten Felds . . . . .	249
Vorteile der berechneten Felder . . . . .	250
Berechnete Felder erstellen . . . . .	250
Arbeiten mit berechneten Feldern . . . . .	257
<b>Kapitel 14. Assistenten . . . . .</b>	<b>259</b>
Schreibweisenkonventionen . . . . .	259
Definitionen zu Assistenten . . . . .	260
Stukturmodelle . . . . .	263
Modell der Seite eines Assistenten . . . . .	265
Allgemeine Informationen zu Assistenten . . . . .	266
Allgemeine Struktur und Syntax eines Assistentenknotens . . . . .	266
Eigenschaften eines Knotens . . . . .	267
Verkettung von Assistenten . . . . .	272
BASIC-Funktionen . . . . .	273
Definition eines Knotens vom Typ Root . . . . .	273
Syntax eines Knotens vom Typ Root . . . . .	274
Eigenschaften des Knotens Root . . . . .	274
Dem Knoten Root untergeordnete Knoten . . . . .	281
Definition des Knotens Page . . . . .	282
Syntax eines Knotens vom Typ Page . . . . .	282
Eigenschaften eines Knotens vom Typ Page . . . . .	283
Dem Knoten Page untergeordnete Knoten . . . . .	284
Definition eines Knotens vom Typ Transition . . . . .	285
Syntax eines Knotens vom Typ "Transition" . . . . .	285
Eigenschaften eines Knotens vom Typ Transition . . . . .	285
Besonderheiten eines Knotens vom Typ Transition . . . . .	287
Definition eines Knotens vom Typ Finish . . . . .	287
Definition eines Knotens vom Typ Start . . . . .	289
Definition eines Knotens vom Typ Timer . . . . .	290
Definition der Knoten vom Typ Long und String . . . . .	291
Definition eines Knotens vom Typ Steuerelement . . . . .	292
Allgemeine Syntax eines Knotens vom Typ Steuerelement . . . . .	292
Steuerelementtypen mit ihren zugeordneten Eigenschaften . . . . .	293
Arbeiten mit dem Grafikeditor . . . . .	325
Beispiel für die Erstellung eines Assistenten . . . . .	330

Beispiel für die Erstellung eines Abfrageassistenten (QBE) . . . . .	336
Häufig gestellte Fragen . . . . .	341
<b>Kapitel 15. Mitteilungen . . . . .</b>	<b>347</b>
Definition einer Mitteilung . . . . .	347
Allgemeine Funktionsweise von Mitteilungen . . . . .	348
Bedeutung der Mitteilungen definieren . . . . .	348
Mitteilungen verteilen . . . . .	348
Empfängerliste für Mitteilungen . . . . .	349
Mitteilungen anzeigen . . . . .	349
<b>Index . . . . .</b>	<b>351</b>





1.1. Aufzählung - Fenster . . . . .	14
3.1. Abfrageeditoren - Entwicklungsarten . . . . .	40
4.1. Formulare - Kopffeld . . . . .	80
6.1. Aktion vom Typ Programm - Detailfenster . . . . .	108
6.2. "Standardbrief.doc" . . . . .	109
6.3. Aktion vom Typ Nachricht mit einem Referenzobjekt - Detailfenster . . . . .	112
7.1. Nachrichtensystem - Allgemeine Funktionsweise . . . . .	120
8.1. Workflows - Vereinfachtes Modell . . . . .	124
8.2. AssetCenter-Workflow - Allgemeine Übersicht . . . . .	128
8.3. Workflow - Zum Definieren eines Workflow-Modells hauptsächlich verwendete Tabellen . . . . .	129
8.4. Workflow - Im Rahmen des Ablaufs einer Workflow-Instanz hauptsächlich verwendete Tabellen . . . . .	129
8.5. Workflow - Modell zur Bewilligung von Anforderungen . . . . .	134
8.6. Workflow - Modell zur Bewilligung von Anforderungen . . . . .	147
8.7. Registerkarte Parameter für ein Ereignis vom Typ Bei Datenbankänderung . . . . .	167
8.8. Beispiel eines synchronen Workflows . . . . .	173
8.9. Beispiel eines asynchronen Workflows . . . . .	174
8.10. Workflow-Modell mit einem abschließenden Ereignis . . . . .	175
10.1. Skript - Editor . . . . .	204
11.1. Kalender - Registerkarte Arbeitszeiten . . . . .	226
11.2. Kalender - Registerkarte Vorschau . . . . .	231
14.1. Assistenten - Strukturmodell . . . . .	265

14.2. Schaltflächen zur Ausführung und Fehlerbehebung . . . . .	326
14.3. Assistent - Beispiel . . . . .	332

1.1. Werte der Systemaufzählungen . . . . .	16
3.1. AQL - Syntaxkonventionen . . . . .	46
3.2. AQL - Logische Operatoren . . . . .	51
3.3. ASL - Vergleichsoperatoren . . . . .	51
3.4. AQL - Funktionen vom Typ Aggregat . . . . .	61
3.5. AQL - Funktionen vom Typ String . . . . .	62
3.6. AQL - Funktionen vom Typ Datum . . . . .	62
3.7. AQL - Funktionen vom Typ Datum . . . . .	64
3.8. AQL - Funktionen vom Typ Numerisch . . . . .	64
3.9. AQL - Funktionen vom Typ Test . . . . .	65
8.1. Beschränkungen für Workflow-Instanzen . . . . .	169
8.2. Möglichkeiten zur Verarbeitung von Ereignissen . . . . .	170
8.3. Möglichkeiten zur Verarbeitung von Ereignissen . . . . .	172
10.1. Funktionen/Parameter - Typen . . . . .	206
13.1. Typen von berechneten Feldern . . . . .	251
14.1. Verwendete Konventionen . . . . .	260
14.2. Logische Eigenschaften des Knotens "Root" . . . . .	275
14.3. Physische Eigenschaften des Knotens "Root" . . . . .	278
14.4. Dem Knoten "Root" untergeordnete Knoten . . . . .	281
14.5. Logische Eigenschaften eines Knotens vom Typ "Page" . . . . .	283
14.6. Physische Eigenschaften eines Knotens vom Typ "Page" . . . . .	284
14.7. Dem Knoten "Page" untergeordnete Knoten . . . . .	284

14.8. Logische Eigenschaften eines Knotens vom Typ "Transition" . . . . .	286
14.9. Logische Eigenschaften eines Knotens vom Typ "Finish" . . . . .	288
14.10. Physische Eigenschaft mit einem Knoten vom Typ "Finish" . . . . .	288
14.11. Logische Eigenschaft eines Knotens vom Typ "Start" . . . . .	290
14.12. Logische Eigenschaften vom Typ "Timer" . . . . .	290
14.13. Logische Eigenschaft vom Typ Long oder String . . . . .	292
14.14. Logische Eigenschaften, die allen Steuerelementen gemein sind . . . . .	293
14.15. Physische Eigenschaften, die allen Steuerelementen gemein sind . . . . .	295
14.16. Eigenschaft des Steuerelements "CHECKBOX" . . . . .	298
14.17. Physische Eigenschaften des Steuerelements "COMBOBOX" . . . . .	299
14.18. Physische Eigenschaften des Steuerelements "OPTIONBUTTONS" . . . . .	300
14.19. Physische Eigenschaften des Steuerelements "LISTBOX" . . . . .	301
14.20. Methoden des Steuerelements "LISTBOX" . . . . .	303
14.21. Obligatorische, logische Eigenschaft des Steuerelements "LISTBOX" . . . . .	304
14.22. Physische Eigenschaften des Steuerelements "LABEL" . . . . .	305
14.23. Physische Eigenschaften des Steuerelements "PROGRESSBAR" . . . . .	306
14.24. Physische Eigenschaften des Steuerelements "COMMANDBUTTON" . . . . .	306
14.25. Physische Eigenschaften des Steuerelements "DBLISTBOX" . . . . .	308
14.26. Physische Eigenschaften des Steuerelements "DBQUERYBOX" . . . . .	312
14.27. Physische Eigenschaften des Steuerelements "DBEDIT" im Modus "Normal" . . . . .	315
14.28. Obligatorische, physische Eigenschaften des Steuerelements "DBPATH" . . . . .	317
14.29. Logische Eigenschaft des Steuerelements "LINKEDIT" . . . . .	317
14.30. Physische Eigenschaft des Steuerelements "TEXTBOX" . . . . .	319
14.31. Logische Eigenschaften des Steuerelements "CHART" . . . . .	320
14.32. Physische Eigenschaften des Steuerelements "CHART" . . . . .	321
14.33. Eigenschaften des Steuerelements "FILEEDIT" . . . . .	323
14.34. Eigenschaften des Steuerelements TICKEDIT . . . . .	323
14.35. Eigenschaften des Steuerelements NUMBOX . . . . .	324
14.36. Eigenschaft des Steuerelements COMBOEDIT . . . . .	324
14.37. Eigenschaften des Steuerelements . . . . .	325
15.1. Schaltflächen in der Mitteilungsanzeige . . . . .	350

# 1 Aufzählungen

## KAPITEL

Eine Aufzählung ist eine Liste mit Werten, die AssetCenter zur Eingabe in bestimmte Felder, z. B. in das Standardfeld eines Detailbildschirms oder in den Wert eines Merkmals, vorschlägt. Dazu gehören u. a. Anrede, Funktion, Land oder Marke.

Auf diese Weise können die Werte der betreffenden Felder standardisiert werden, und die Eingabe wird vereinfacht.

Die Liste der Werte erscheint in Form einer Dropdown-Liste, in der Sie den gewünschten Wert für das entsprechende Feld wählen.

AssetCenter verwaltet zwei Arten von Aufzählungen:

- Anpassbare Aufzählungen
- Systemaufzählungen

## Anpassbare Aufzählungen




AssetCenter-Administratoren greifen über das AssetCenter-Menü **Verwaltung/ Aufzählungen** auf die anpassbaren Aufzählungen zu.

AssetCenter bietet zwei Arten von anpassbaren Aufzählungen:

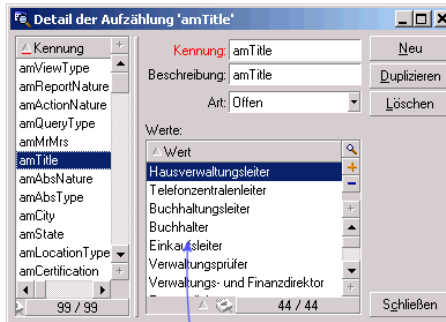
- Aufzählungen, die Sie selber erstellen. Sie können diese Aufzählungen mit Merkmalen verknüpfen, während eine Zuordnung zu Feldern nicht möglich ist. Die Zuordnung einer Aufzählung zu einem Feld erfolgt in der Software.
- Aufzählungen, die mit Feldern der Datenbank verbunden sind. Auch diesen Aufzählungen können Merkmale zugeordnet werden. Wenn Sie eine Aufzählung dieser Art löschen oder ihren Namen ändern, erstellt AssetCenter die Aufzählung unter ihrem ursprünglichen Namen und ohne ihre zugeordneten Werte noch einmal, wenn sie zur Erstellung eines Datensatzes herangezogen wird, in dem Feld mit ihr verknüpft wird.

## Werte der Aufzählungen

Die Aufzählungswerte werden beim Ausfüllen von Feldern, denen Aufzählungen zugeordnet sind, im Detailbereich angezeigt.

Ein Administrator kann die Werte durch Klicken auf die Schaltflächen ,  und  rechts neben der Liste löschen, ändern oder hinzufügen.

**Abbildung 1.1. Aufzählung - Fenster**



Liste der Werte, die bei Eingaben im entsprechenden Feld angezeigt wird.

## Offene Aufzählungen

Im Feld **Typ** (SQL-Name: seType) des Aufzählungsdetails erscheint der Wert **Offen**.

AssetCenter-Benutzer können zusätzlich zu den bereits vorhandenen Werten weitere Werte eingeben.

Wenn ein Benutzer einen neuen Wert eingibt, wird dieser der Liste der Werte für die Aufzählung hinzugefügt. Diese Liste ist allen Benutzern gemein. Über eine Meldung der Benutzer zur Bestätigung dieser Erstellung aufgefordert.

## Geschlossene Aufzählungen

Im Feld **Typ** des Aufzählungsdetails erscheint der Wert **Geschlossen**.

AssetCenter-Benutzer können der Liste keine Werte für die entsprechende Aufzählung hinzufügen.



Anmerkung:

Alle Werte, die der Administrator in Aufzählungen definiert, werden in der Datenbank gespeichert, sobald der Administrator im Fenster zur Verwaltung der Aufzählungen auf die Schaltfläche **Erstellen** oder **Ändern** klickt. Auf den einzelnen AssetCenter-Clients wird eine Änderung des Aufzählungstyps erst nach dem Trennen und dem Wiederherstellen der Verbindung berücksichtigt.

## Systemaufzählungen

Die Liste der Werte für eine Systemaufzählung wird von AssetCenter vorgegeben und kann weder vom Administrator noch vom Benutzer angepasst werden.

Diese Aufzählungen lassen sich nicht über das Menü **Verwaltung/ Aufzählungen** bearbeiten.

## Werte der Systemaufzählungen

Die angezeigten Werte unterscheiden sich von den in der Datenbank gespeicherten Werten.

Bei den in der Datenbank gespeicherten Werten handelt es sich um Zahlen.

Beispiel für das Feld **Zuordnung** (SQL-Name: seAssignment) im Detailbereich eines Vermögensgegenstands.

**Tabelle 1.1. Werte der Systemaufzählungen**

<b>Wert in Datenbank</b>	<b>Angezeigter Wert</b>
0	<b>In Betrieb</b>
1	<b>Lagerhaltig</b>
2	<b>Abgang (oder verbraucht)</b>
3	<b>Ausstehende Lieferung</b>

Beim Zugreifen auf die Werte einer Systemaufzählung haben Sie die Wahl zwischen mehreren Methoden:

- Über die Kontexthilfe des Felds, in dem die Systemaufzählung erscheint
- Über AssetCenter Database Administrator
- Über die Datei **database.txt** mit der Beschreibung der Datenbankstruktur



# 2 | Anlegen von Historien

---

## KAPITEL

Die Änderungen, die Sie an den Feldern und Verknüpfungen der Datenbanktabellen vornehmen, können in einer Historie festgehalten werden. Jedes Mal, wenn Sie einen Wert in einem historisierte Feld erstellen, ändern oder löschen, generiert AssetCenter eine Historienzeile in der Registerkarte **Historie** des entsprechenden Bildschirms.

Gehen Sie zum Anlegen einer Historie für ein Feld oder eine Verknüpfung wie folgt vor:

- 1 Verwenden Sie die Option **Objekt konfigurieren** im Kontextmenü.
- 2 Setzen Sie den Cursor auf die Registerkarte **Allgemein** des Konfigurationsfensters.
- 3 Wählen Sie für das Feld **Historisiert** den Wert **Ja**.
- 4 Bestätigen Sie die Angaben durch Klicken auf **OK**.

---

 **Anmerkung:**

Sobald Sie im Fenster zur Anpassung der Datenbank auf die Schaltfläche **OK** klicken, werden die Änderungen bzgl. der Historienerstellung in der Datenbank gespeichert. Die Angabe der Felder und Verknüpfungen, für die eine Historie angelegt werden soll, kann auch über AssetCenter Database Administrator erfolgen.

---

Historisierte Felder und Verknüpfungen können von allen AssetCenter-Benutzer eingesehen werden.

Sobald eine Historie für ein Feld oder eine Verknüpfung einer Tabelle erstellt wird, erscheint die Registerkarte **Historie** im Datensatzdetail dieser Tabelle. Sie enthält die Historienzeilen mit einer detaillierten Beschreibung der an dem Feld oder der Verknüpfung vorgenommenen Änderungen.

In den Historienzeilen sind nachstehende Informationen enthalten:

- **Änderungsdt.** (SQL-Name: dtLastModif): Datum, an dem die Änderung vorgenommen wurde.
- **Autor** (SQL-Name: Author): Benutzer, der die Änderung eingegeben hat (Login, Name und Vorname).
- **Feld oder Verknüpfung** (SQL-Name: Field): Name des geänderten Felds bzw. der Verknüpfung (kurze Beschreibung).
- **Alter Wert** (SQL-Name: PreviousVal): Wert vor der Änderung (mit Ausnahme der Felder vom Typ "Kommentar").
- **Neuer Wert** (SQL-Name: NewVal): Neuer Wert für ein geändertes Feld (mit Ausnahme der Felder vom typ "Kommentar"). Dieses Feld wird nicht standardmäßig in der Liste angezeigt. Um es anzuzeigen, klicken Sie mit der rechten Maustaste auf die Liste und wählen im daraufhin erscheinenden Kontextmenü die Option **Liste konfigurieren**.

---

 **Anmerkung:**

Beim Import von AssetCenter-Datenbanken aus älteren Versionen bleibt das Feld **Neuer Wert** in den Historienzeilen leer.

- 
- **Alter Kommentar** (SQL-Name: memPreviousCmt): Wert des Felds vom Typ "Kommentar" vor der Änderung. Die Bearbeitung der Felder vom Typ "Kommentar" unterscheidet sich von den übrigen Feldern in der Art ihrer Speicherung in der Datenbank (max. 32767 Zeichen).

Im Folgenden sind die Einsatzbereiche von Historienzeilen aufgeführt.

## Erstellen eines Datensatzes

Die Erstellung von Datensätzen wird gespeichert, sofern Sie festgelegt haben, dass AssetCenter für die am Primärschlüssel der Tabelle vorgenommenen Änderungen eine Historienzeile anlegen soll.

AssetCenter speichert die nachstehenden Informationen:

- **Änderungsdt.:** Datum, an dem die Erstellung vorgenommen wurde
- **Autor:** Benutzer, der die Erstellung vorgenommen hat
- **Feld oder Verknüpfung:** "Erstellung"
- **Alter Wert:** "Erstellung"

## Ändern des Felds einer Tabelle oder einer 1-Verknüpfung (Beispiel: Benutzer eines Vermögensgegenstands)

AssetCenter speichert die nachstehenden Informationen:

- **Änderungsdt.:** Datum, an dem die Änderung vorgenommen wurde
- **Autor:** Benutzer, der die Änderung vorgenommen hat
- **Feld oder Verknüpfung:** Name des geänderten Felds
- **Alter Wert:** Wert des Felds vor der Änderung
- **Neuer Wert:** Wert des Felds nach der Änderung

## Hinzufügen einer n-Verknüpfung zu einer anderen Tabelle (Beispiel: durch Verträge abgedeckte Vermögensgegenstände)

AssetCenter speichert die nachstehenden Informationen:

- **Änderungsdt.:** Datum, an dem die Verknüpfung hinzugefügt wurde
- **Autor:** Benutzer, der die Verknüpfung hinzugefügt hat
- **Feld oder Verknüpfung:** Name der Verknüpfung
- **Alter Wert:** Referenzen des verknüpften Datensatzes, der hinzugefügt wurde
- **Neuer Wert:** Wert der Verknüpfung nach der Änderung

## Löschen der n-Verknüpfung zu einer anderen Tabelle

AssetCenter speichert die nachstehenden Informationen:

- **Änderungsdt.:** Datum, an dem die Verknüpfung gelöscht wurde
- **Autor:** Benutzer, der die Verknüpfung gelöscht hat
- **Feld oder Verknüpfung:** Referenzen des verknüpften Datensatzes, der gelöscht wurde
- **Alter Wert:** Referenz des verknüpften Datensatzes, der gelöscht wurde
- **Neuer Wert:** Wert der Verknüpfung nach der Änderung (leer)

## Ändern der n-Verknüpfung zu einer anderen Tabelle

Die Änderungen der Verknüpfung werden von AssetCenter nicht gespeichert. Wenn Sie die Änderung trotzdem protokollieren möchten, müssen Sie die alte Verknüpfung löschen und eine neue Verknüpfung hinzufügen.

## Anlegen von Historienzeilen für Merkmale und Merkmalwerte

Mit AssetCenter können Sie Historienzeilen für Merkmale anlegen, wie für jedes andere Feld der Datenbank. Das Anlegen von Historienzeilen betrifft folgende Vorgänge:

- Hinzufügen eines Merkmals
- Löschen eines Merkmals
- Ändern eines Merkmalwerts

Im Folgenden sind die unterschiedlichen Anwendungsbereiche aufgeführt.

### Hinzufügen eines Merkmals

Das Hinzufügen von Merkmalen wird gespeichert, wenn der Parameter **Historie anlegen** (SQL-Name: seKeepHistory) des Merkmals den Wert **Ja** aufweist, und das Kontrollkästchen **Historie auch bei Erstellen von Hauptdatensatz anlegen** (SQL-Name: bCreationHistory) markiert ist.

AssetCenter speichert die nachstehenden Informationen:

- **Änderungsdt.:** Datum, an dem das Merkmal hinzugefügt wurde.

- **Autor:** Benutzer, der das Merkmal hinzugefügt hat.
- **Alter Wert:** Name und Wert des Merkmals: "Erstellung"
- **Feld oder Verknüpfung:** SQL-Name des Merkmals

## Löschen eines Merkmals

Das Löschen von Merkmalen wird gespeichert, wenn der Parameter **Historie anlegen** des Merkmals den Wert **Ja** aufweist.

AssetCenter speichert die nachstehenden Informationen:

- **Änderungsdt.:** Datum, an dem das Merkmal gelöscht wurde.
- **Autor:** Benutzer, der das Merkmal gelöscht hat.
- **Feld oder Verknüpfung:** SQL-Name des Merkmals.
- **Alter Wert:** "Merkmal ('Merkmalwert') löschen".
- **Neuer Wert:** Wert des Merkmals nach der Änderung (leer)

## Ändern eines Merkmalwerts

Das Ändern von Merkmalwerten wird gespeichert, wenn der Parameter **Historie anlegen** des Merkmals den Wert **Ja** aufweist.

AssetCenter speichert die nachstehenden Informationen:

- **Änderungsdt.:** Datum, an dem das Merkmal geändert wurde.
- **Autor:** Benutzer, der das Merkmal geändert hat.
- **Feld oder Verknüpfung:** SQL-Name des Merkmals.
- **Alter Wert:** Alter Wert des Merkmals.
- **Neuer Wert:** Wert des Merkmals nach der Änderung

---

### **WARNUNG:**


Mit dem Löschen eines Datensatzes gehen gleichzeitig alle Historienzeilen verloren, und zwar entweder direkt beim Löschen oder aufgrund von Vorgängen, die von AssetCenter Server ausgeführt werden.

---

## Anlegen, Ändern und Löschen einer Historienzeile

Das Generieren von Historienzeilen bei der Anlage von Historien ist nicht möglich.

### Anlegen von Historienzeilen

Wenn Sie die Anlage von Historienzeilen für ein Merkmal starten möchten, muss das Feld **Historie anlegen** den Wert **Ja** aufweisen. Öffnen Sie zu diesem Zweck die Registerkarte **Parameter** im Merkmaldetail, und klicken Sie neben den Parameterzeilen auf die Schaltfläche .

AssetCenter öffnet daraufhin das Detailfenster mit den Merkmalparametern. Das Feld **Historie anlegen** befindet sich auf der Registerkarte **Einstellungen**.

Erscheint in diesem Feld der Wert **Ja**, legt AssetCenter automatisch Historienzeilen für dieses Merkmal an. Diese können dann auf der Registerkarte **Historie** der Tabelle eingesehen werden, der das Merkmal zugeordnet ist.

---

#### **WARNUNG:**

Mit dem Löschen eines Datensatzes gehen gleichzeitig alle Historienzeilen verloren, und zwar entweder direkt beim Löschen oder über AssetCenter Server. Das Generieren von Historienzeilen bei der Anlage von Historien ist nicht möglich.

---

# 3 | AQL-Abfragen

---

## KAPITEL

In diesem Kapitel wird das Schreiben von AQL-Abfragen erläutert.

## Einführung

In diesem Abschnitt werden die Programmiersprache AQL vorgestellt und die Teile der Software aufgeführt, in denen Sie Abfragen verwenden können.

## AQL

AQL (Advanced Query Language) ist eine Abfragesprache, die AssetCenter beim Zugreifen auf die Daten der AssetCenter-Datenbank verwendet. AQL ähnelt der Programmiersprache SQL und wird bei der Verwendung von Abfragen der Datenbank-Engine automatisch in die Sprache SQL übersetzt.

**Anmerkung:**

Bevor Sie direkt mit AQL arbeiten, sollten Sie unbedingt über ausreichende Kenntnisse in SQL verfügen und sich gut mit der Funktionsweise von Datenbanken auskennen.

## Vorteile der Sprache AQL

Aus den folgenden Gründen ist AQL besser zum Schreiben von Abfragen für die AssetCenter-Datenbank geeignet als SQL:

### Unabhängigkeit in Bezug auf die Datenbank-Engine

Während die von AssetCenter unterstützten Datenbank-Engines alle SQL-Varianten verwenden, auch wenn diese Sprachen nicht unbedingt miteinander kompatibel sind, funktioniert AQL unabhängig von der jeweils eingesetzten Datenbank-Engine.

Wenn Sie also Abfragen in AQL schreiben und dann zwischen verschiedenen Datenbank-Engines wechseln, lassen sich die Abfragen weiterhin benutzen.

Ein Beispiel: AQL verwendet für alle Datenbank-Engines den gleichen Funktionssatz.

Die AQL-Funktion **Substring** entspricht damit der Funktion **Substr** in SQL Oracle for WorkGroups und der Funktion **Substring** in SQL Microsoft SQL Server.

### Generierung des optimierten SQL-Codes

AQL generiert unter Berücksichtigung der Datenbank-Engine einen optimierten SQL-Code.

Das macht sich vor allem bei der Verwendung der Indizes bemerkbar. Ein Beispiel: Sie wollen den vollständigen Namen von Modellen und die Verwendung der Indizes **Modell-ID (Model\_lModelId)** und **Vollständiger Name (FullName)** erzwingen. Sie schreiben also die nachstehende AQL-Abfrage:

```
SELECT FIRST_ROWS lModelId, FullName FROM amModel
```

Je nach Datenbank-Engine wird ein unterschiedlicher SQL-Code generiert und unter Berücksichtigung der jeweils verwendeten Datenbank-Engine optimiert. Der entsprechende SQL-Code von Oracle lautet wie folgt:



```
SELECT /*+ FIRST_ROWS INDEX_ASC(M1 Model_lModelId) */ M1.lModelId, M1.
FullName FROM amModel M1
```

Bei Microsoft SQL Server oder Sybase SQL Server lautet er folgendermaßen:

```
SELECT M1.lModelId, M1.FullName FROM amModel M1 ORDER BY M1.lModelId
```

Der Code von IBM DB2 lautet:

```
SELECT lModelId, FullName FROM amModel OPTIMIZE FOR 100 ROWS
```

## Zugriffsmöglichkeiten auf die AssetCenter-Datenbankstruktur

Mit AQL lassen sich Verknüpfungen und Beziehungen mühelos verwalten, was gleichzeitig den Zugriff auf die Datenbankstruktur beim Schreiben von Abfragen im Vergleich zur direkten Verwendung von SQL wesentlich vereinfacht.

Zusätzlich gestaltet AQL auch den Zugriff auf die Merkmale übersichtlicher, indem sich die Merkmale wie Felder in den Tabellen bearbeiten lassen, auf die sie sich beziehen.

Weiterhin erleichtert AQL die Verwendung von berechneten Feldern.

## AQL und SQL im Vergleich

AQL unterstützt keine DDL-Befehle (Data Definition Language).

AQL bietet Erweiterungen zur Erleichterung der Verwaltung von Verknüpfungen sowie der Verwendung von Merkmalen und berechneten Feldern.

### WARNUNG:

Sie dürfen grundsätzlich keine SQL-Schreibbefehle zur direkten Eingabe in die AssetCenter-Datenbank verwenden.

## Abfragen in AssetCenter

Die Abfragen ermöglichen das Kombinieren mehrerer Auswahlkriterien, die sich auf Informationen einer Tabelle oder auf Informationen miteinander verknüpfter Tabellen beziehen.

Abfragen dienen unterschiedlichen Zwecken:

- Erstellen von Filtern für Datensatzlisten. In diesem Fall weisen die Abfragen im Allgemeinen eine einfache, auf die Klausel "WHERE" aufbauende Struktur auf.
- Definieren von Ansichten.
- Definieren von Exportbedingungen im Modul Export.
- Erstellen von Crystal Reports-Berichten.
- Erstellen von Assistenten.
- Verwenden der AssetCenter API.
- Verwenden von AssetCenter als DDE-Befehlsserver.

Die Abfragen werden in der Programmiersprache AQL (Advanced Query Language) geschrieben. Dabei handelt es sich um eine interne Sprache, die AssetCenter beim Zugreifen auf die Daten der AssetCenter-Datenbank verwendet.

AssetCenter enthält einen Abfrageeditor, der Ihnen beim Schreiben von Abfragen zwei Möglichkeiten bietet:

- Verwendung der Funktionen der grafischen Benutzeroberfläche (Unterstützung beim Schreiben von Abfragen)
- Direkte Verwendung von AQL

---

 **WARNUNG:**

Zum besseren Verständnis wird in den folgenden Beispielabfragen die gesamte AQL-Syntax dargestellt. Dies betrifft insbesondere die Klauseln SELECT, WHERE, FROM usw. Die Software enthält Funktionen, die die Erstellung von Abfragen erleichtern. Dazu gehören die Abfragefilter (der Benutzer definiert nur die Klausel WHERE der AQL-Abfrage) und der Ausdruckeditor. Bei Verwendung dieser Funktionen sind bestimmte Klauseln für den Benutzer transparent. Aus diesem Grund ist es nicht möglich, die in diesem Kapitel vorgestellten Beispiele ohne Änderung zu übernehmen.

---

## Empfehlungen zum Schreiben von AQL-Abfragen

Wir empfehlen Ihnen, diesen Abschnitt vor dem Schreiben von AQL-Abfragen zu lesen.

In diesem Abschnitt finden Sie folgende Informationen:

- AQL-spezifische Notationen

- Besonderheiten der Sprache AQL und der AssetCenter-Datenbank, die sich auf die Schreibweise von AQL-Abfragen auswirken

Die Abschnitte [Syntax der Programmiersprache AQL](#) [Seite 46] und [AQL-Funktionen](#) [Seite 60] dieses Kapitels ergänzen die hier behandelten Themen.

### **WARNUNG:**

In den AQL-Abfragen kommen die SQL-Namen (SQLName) der Felder, Verknüpfungen und Tabellen der AssetCenter-Datenbank zur Anwendung. Eine ausführliche Liste dieser Namen finden Sie in der Datei **database.txt**, die die Beschreibung der Datenbankstruktur enthält.

Diese Datei befindet sich im folgenden Ordner:  
**[AssetCenter-Installationsordner]/doc/infos**

## AQL-Beziehungen

### Definition

Eine Beziehung ist die Verbindung, die in einer Abfrage zwischen zwei oder mehreren verknüpften Tabellen hergestellt wird.

### AQL-Beziehungen

Neben Tabellen und Feldern definiert das AssetCenter-Datenmodell auch die Beziehungen zwischen den Tabellen. Das ermöglicht das Automatisieren der Generierung von Beziehungsbedingungen auf der AQL-Ebene.

Die AQL-Verknüpfungen stellen sich dar wie folgt:

```
Verknüpfung[.Verknüpfung[.Feld]]
```

Durch die vereinfachte Verwaltung von Beziehungen erleichtert AQL gleichzeitig das Schreiben der meisten Datenbankabfragen.

### Beispiel

Die nachstehende, in AQL geschriebene Abfrage gibt für jedes Modell Folgendes zurück:

- Die **ID (IModelId)**
- Den **vollständigen Namen (FullName)**

- Den **Namen (Name)** der verknüpften Tabelle der Marken (**amBrand**)

```
SELECT lModelId, FullName, Brand.Name FROM amModel
```

Nachstehend dieselbe Abfrage in Oracle SQL oder Microsoft SQL Server:

```
SELECT M1.lModelId, M1.FullName, B2.Name FROM amModel M1, amBrand B2 W  
HERE M1.lBrandId=B2.lBrandId
```

Die beiden Beziehungen zwischen der Tabelle der **Modelle (amModel)** und der Tabelle der **Marken (amBrand)** werden von AQL automatisch verwaltet. Auf der grafischen Benutzeroberfläche des AssetCenter-Abfrageeditors klicken Sie einfach in der Baumstruktur auf die Felder der ausgewählten Tabelle oder auf die Felder der verknüpften Tabellen, um den entsprechenden AQL-Code zu generieren.

#### Anmerkung:

Für alle Systeme mit Ausnahme von Oracle und DB2 ist die Anzahl der externen Beziehungen auf 1 beschränkt.

Um Problemen bei der Ausführung von Abfragen vorzubeugen, können Sie die Datei **amdb.ini** im Verbindungsdetail mithilfe der folgenden Anweisung für Microsoft SQL Server 7 und MSSQL 2000 ändern:

```
UseSql92Join=1
```

Ablageort der Datei: ► Handbuch **AssetCenter - Installation und Aktualisierung**, Kapitel **.ini- und .cfg-Dateien**.

## Vorzüge der Datensätze mit dem Primärschlüssel 0

### Datensätze mit dem Primärschlüssel 0

Die Modelle der AssetCenter-Datenbank weisen einige Besonderheiten auf:

- Die Primärschlüssel und die externen Schlüssel jeder Tabelle sind vom Typ "Integer (32 Bit)".
- Ein externer Schlüssel, der nicht auf einen Datensatz zeigt, erhält den Wert 0 (und nicht NULL).
- Jede Tabelle verfügt über einen leeren Datensatz mit einem Primärschlüssel, für den der Wert "0" gilt.

## Vorzüge

Dank dieser Datensätze mit dem Primärschlüssel 0 kann das Ergebnis einer Abfrage, bei der eine nicht externe Beziehung zwischen zwei Tabellen A und B zur Anwendung kommt, die Datensätze der Tabelle A einschließen, die mit keinem "reellen" Datensatz der Tabelle B verknüpft sind. In diesem Fall ist keine Verknüpfung angegeben. Es handelt sich dabei um die Datensätze der Tabelle A, die mit einem Datensatz der Tabelle B verknüpft sind, für den der Primärschlüssel 0 gilt.

Ein Beispiel:

Die folgende AQL-Abfrage gibt für jeden internen Code eines Ausrüstungselements, den Namen seines Benutzers und den Namen seines Verwalters zurück:

```
SELECT AssetTag, User.Name, Supervisor.Name FROM amPortfolio
```

Auch ein Ausrüstungselement, das keinem Benutzer und/oder Verwalter zugeordnet ist, erscheint im Ergebnis dieser Abfrage. In der Datenbank ist ein solches Ausrüstungselement mit dem Datensatz der Tabelle der Abteilungen und Personen verknüpft, für den der Primärschlüssel 0 gilt.

## Grund für diese Besonderheiten

In diesem Abschnitt wird der Grund für das Vorhandensein von Datensätzen mit dem Primärschlüssel 0 erklärt, obwohl eine Abfrage, bei der eine externe SQL-Beziehung zwischen zwei Tabellen A und B zur Anwendung kommt, alle Datensätze der Tabelle A wählen kann, die mit einem Datensatz der Tabelle B verknüpft sind.

Durch die Datensätze mit dem Primärschlüssel 0 umgehen Sie das Problem der fehlenden Unterstützung von kaskadenartigen externen Beziehungen in bestimmten Datenbankmanagementsystemen: Dank der Datensätze mit dem Primärschlüssel 0 greifen die SQL-Abfragen, die über eine auf Beziehungen aufbauende AQL-Abfrage generiert wurden, nicht auf externe Beziehungen zurück.

Ein Beispiel:

Die nachstehende AQL-Abfrage sucht für jedes Ausrüstungselement den internen Code und den Namen des Standorts des Benutzers. Das Ergebnis beinhaltet die Ausrüstungselemente, für die kein Benutzer angegeben ist, und die Ausrüstungselemente, für deren Benutzer kein Standort angegeben wurde.

```
SELECT AssetTag, user.location.name FROM amPortfolio
```

Wenn die generierte SQL-Abfrage die externen Beziehungen des Datenbankmanagementsystems benutzen würde, sähe die über Sybase SQL Server generierte SQL-Abfrage aus wie folgt:

```
SELECT a.AssetTag, l.name FROM amPortfolio a, amEmplDept e, amLocation l WHERE a.l.UserId *= e.lEmplDeptId AND e.lLocaId *= l.lLocaId
```

Diese Abfrage wird von Sybase SQL Server allerdings nicht unterstützt, da in ihr kaskadenartige externe Beziehungen zur Anwendung kommen.

Da es in der Tabelle der Abteilungen und Personen sowie in der Tabelle der Standorte jedoch einen Datensatz mit dem Primärschlüssel 0 gibt, ist der Zugriff auf externe SQL-Beziehungen gar nicht erforderlich. AssetCenter generiert dementsprechend eine SQL-Abfrage, in der herkömmliche, also keine externen Beziehungen zur Anwendung kommen:

```
SELECT l.name FROM amPortfolio a, amEmplDept e, amLocation l WHERE a.l.UserId = e.lEmplDeptId AND e.lLocaId = l.lLocaId
```

Mit dieser Abfrage erhalten Sie das gewünschte Ergebnis, da die Verknüpfungen **Benutzer (User)** und **Standort (Location)** immer auf einen Datensatz in der Tabelle der Abteilungen und Personen oder in der Tabelle der Standorte zeigen. Wenn die Verknüpfung nicht angegeben ist, verweisen sie auf den Datensatz mit dem Primärschlüssel 0.

## Konsequenzen

- Beim Schreiben von Abfragen müssen Sie unbedingt auf diese leeren Datensätze achten, und zwar vor allem in den Fällen, in denen Sie Aggregat-Funktionen verwenden.

Ein Beispiel:

```
SELECT count(AssetTag) FROM amPortfolio
```

Wenn Sie diese Abfrage zum Zählen der Anzahl von Vermögensgegenständen in der Tabelle der Vermögen und Lose verwenden, wird der Datensatz mit dem Primärschlüssel 0 im Ergebnis berücksichtigt. Sie müssen das Ergebnis also um 1 reduzieren, um die tatsächliche Anzahl der in der Datenbank gespeicherten Vermögensgegenstände zu erhalten.

- Das Generieren externer Beziehungen im DBMS ist nur in seltenen Fällen erforderlich.

 Anmerkung:

Verwenden Sie die AQL-Operatoren "=\*" und "\*=", wenn Sie externe Beziehungen im DBMS generieren möchten.

## Verwendung des Werts NULL

AssetCenter verwendet den Wert NULL des DBMS lediglich in zwei Fällen:

- Für ein leeres Feld vom Typ "Text"
- Für ein Feld vom Typ "Date" oder "date and time", in das kein Wert eingegeben wurde

AQL bietet Ihnen die Möglichkeit der Verwendung verschiedener Syntaxstrukturen, und zwar durch Konvertieren der AQL-Syntax in die entsprechende SQL-Syntax der Datenbank-Engine.

Für leere Textfelder vom Typ "Text" können Sie eine der nachstehenden Syntaxstrukturen verwenden. Sie müssen lediglich darauf achten, dass in der Datenbank grundsätzlich der Wert NULL gespeichert wird:

**WHERE <Textfeld> = NULL**

**WHERE <Textfeld> IS NULL**

**WHERE <Textfeld> = "**

Für leere Felder vom Typ "Date" oder "Date and time" können Sie eine der nachstehenden Syntaxstrukturen verwenden. Sie müssen lediglich darauf achten, dass in der Datenbank grundsätzlich der Wert NULL gespeichert wird:

**WHERE <Feld mit Datum oder Datum/Uhrzeit> = NULL**

**WHERE <Feld mit Datum oder Datum/Uhrzeit> IS NULL**

**WHERE <Feld mit Datum oder Datum/Uhrzeit> = []**

 Anmerkung:

Wenn in einem numerischen Feld keine Eingabe gemacht wurde, gilt der Wert "0". Für fehlende Verknüpfungen gilt der Wert "Verknüpfung = 0" oder "Externer Schlüssel = 0". Beispiel: "Location=0" oder "ILocaId=0".

## Self

"Self" ist ein Ausdruck, der der Beschreibungsfolge der Tabelle entspricht, auf die der Ausdruck angewendet wird.

Der Ausdruck "Self" erleichtert das Schreiben von Abfragen und berücksichtigt gleichzeitig die Anpassung der AssetCenter-Datenbank.

Ein Beispiel:

Die Beschreibungsfolge der Tabelle der Abteilungen und Personen lautet wie folgt:

```
[Name], [FirstName], ([Phone])
```

Die AQL-Abfrage:

```
SELECT self FROM amEmplDept
```

Die Entsprechung der Abfrage:

```
SELECT (((((Name + ',') + FirstName) + '(') + Phone) + ')') FROM amEmplDept
```

## CurrentUser

Mit Hilfe des Ausdrucks "CurrentUser" schreiben Sie Abfragen unter Berücksichtigung der mit der Datenbank verbundenen Person.

"CurrentUser" kann beispielsweise als Ausdruck in einer Abfrage oder als Verknüpfung verwendet werden. Sie müssen diesen Ausdruck manuell eingeben, da er vom Abfrageeditor nicht vorgeschlagen wird.

### Verwendung des Typs "Ausdruck"

Ein Beispiel: Sie suchen alle Ausrüstungselemente, die von einer mit der Datenbank verbundenen Person benutzt werden.

```
SELECT lPortfolioItemId FROM amPortfolio WHERE User = CurrentUser
```

### Verwendung des Typs "Verknüpfung"

Der Ausdruck "CurrentUser" kann als Verknüpfung betrachtet werden, die von allen Tabellen ausgeht und auf die Tabelle der Abteilungen und Personen zeigt, die dem aktuellen Benutzer entsprechen.

- In der Form "CurrentUser" zeigt diese Funktion auf den Datensatz, der dem aktuellen Benutzer entspricht.



- In der Form "CurrentUser.Feld" gibt diese Funktion den Wert des Felds für den aktuellen Benutzer zurück.

Ein Beispiel: Wenn vom verbundenen Benutzer eine Aktion ausgelöst wird, können Sie kontextbezogen eine andere Aktion vom Typ Nachricht starten, die dem verbundenen Benutzer automatisch eine Warnung sendet. Dazu müssen Sie im Aktionsdetail folgende Angaben machen:

The screenshot shows a configuration window for a 'Nachrichtensystem' (Message System). It has two tabs: 'Beschreibung' (Description) and 'Nachrichtensystem' (Message System). The 'Nachrichtensystem' tab is active. The 'Referenzobjekt:' (Reference Object) is set to 'Normal'. There is a checkbox for 'Empfangsbestätigung' (Receipt Confirmation) which is unchecked. The 'An:' (To) field contains the expression '[CurrentUser.Email]'. The 'Cc:' (Carbon Copy) field is empty. The 'Bcc:' (Blind Carbon Copy) field is empty. The 'Betreff:' (Subject) field contains 'Test'. The 'Text:' field contains 'Test'.

## Systemaufzählungen

Wenn eine AQL-Abfrage auf eine Systemaufzählung zurückgreift, müssen Sie die Werte verwenden, die in der Datenbank gespeichert sind, und nicht die Werte, die auf dem Bildschirm erscheinen.

Ein Beispiel:

Mit der folgenden Abfrage werden die Verträge ausgewählt, für die im Feld **Typ** (SQL-Name: seType) der Wert **Rahmenvertrag** erscheint:

```
SELECT Self FROM amContract WHERE seType = 1
```

Das Feld **Typ** ist eine Systemaufzählung, für die nachstehende Werte in der Datenbank gespeichert sind:

- 0 für einen Vertrag vom Typ **Sonstige**
- 1 für einen Vertrag vom Typ **Rahmenvertrag**
- 2 für einen Vertrag vom Typ **Leasingvertrag**
- 3 für einen Vertrag vom Typ **Versicherungsvertrag**

- 4 für einen Vertrag vom Typ **Wartungsvertrag**



Anmerkung:

Verwenden Sie AssetCenter Database Administrator, oder rufen Sie die Datei **database.txt** mit der Beschreibung der Datenbankstruktur auf, wenn Sie die Werte der Systemaufzählungen einsehen möchten.

Diese Datei befindet sich im folgenden Ordner:

**[AssetCenter-Installationsverzeichnis]/doc/infos**

## Hierarchische Tabellen

Alle hierarchischen Tabellen enthalten folgende Felder:

- Feld "FullName"
- Feld "sLvl"

### Felder "FullName"

Für jeden Datensatz einer hierarchischen Tabelle speichert das Feld "FullName" den Wert eines Datensatzfelds. Diesem Wert ist eine Baumstruktur vorangestellt, die sich aus den Werten der Felder der übergeordneten Datensätze zusammensetzt und bis zur Wurzel reicht.

Die einzelnen Werte sind durch das Zeichen "/" ohne Leerstellen voneinander getrennt. Das Zeichen erscheint sowohl am Anfang als auch am Ende der Baumstruktur.

Beispiele:

- Für die Tabelle der Vermögen und Lose speichert das Feld "FullName" den internen Code des Gegenstands, dem der interne Code des übergeordneten Gegenstands vorangestellt ist. Diesem übergeordneten internen Code ist wiederum der interne Code des übergeordneten Gegenstands vorangestellt usw.

```
FullName = '/PC118/DD054/CR012/'
```

- Für die Tabelle der Standorte speichert das Feld "FullName" den Namen des Standorts, dem die Namen der übergeordneten Standorte vorangestellt sind.

```
FullName = '/Berlin/Standort Hamburger Straße/Gebäude A/5. Stock/'
```

## Felder "sLvl"

Für jeden Datensatz der hierarchischen Tabelle verweist das Feld "sLvl" auf die entsprechende Ebene in der Baumstruktur.

Die Datensätze an der Wurzel sind mit einer 0 gekennzeichnet.

Name	Strichcode
Meteor-Gebäude	DEMO-L01
29. Stock	DEMO-L02
007 - Konferenzraum	DEMO-L03
008 - Büro	DEMO-L04
009 - Büro	DEMO-L05
010 - Büro	DEMO-L06
011 - Büro	DEMO-L07
012 - Büro	DEMO-L08
30. Stock	DEMO-L09

Mit der nachstehenden Abfrage werden die Datensätze "Vertrieb" sowie die Nebenkompontenten gewählt:

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/Vertriebsleitung/Vertrieb/%') AND (sLvl >= 1)
```

Mit der nachstehenden Abfrage werden die Datensätze "Vertrieb" ohne die Nebenkompontenten gewählt:

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/Vertriebsleitung/Vertrieb/%') AND (sLvl = 1)
```

Mit der nachstehenden Abfrage werden die Nebenkompontenten des Datensatzes "Vertrieb", ohne den Datensatz "Vertrieb" gewählt:

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/Vertriebsleitung/Vertrieb/%') AND (sLvl > 1)
```

## Vereinfachte AQL-Notationen

In diesem Abschnitt werden die Notationen aufgeführt, die das Schreiben von AQL-Abfragen erleichtern.

### Externe Schlüssel

Mit Ausnahme der Klauseln SELECT und ORDER BY entspricht der SQL-Name einer Verknüpfung ohne Punkt dem SQL-Namen des zugeordneten externen Schlüssels.

Beispiel: Die Klausel:

```
WHERE location = 0
```

Entspricht der Klausel:

```
WHERE lLocaId = 0
```

Wobei "Location" dem SQL-Namen der Verknüpfung "Standort" entspricht, der von der Tabelle der Abteilungen und Personen auf die Tabelle der Standorte zeigt. "lLocaId" entspricht dem SQL-Namen des zugeordneten externen Schlüssels in der Tabelle der Vermögensgegenstände.

## Beschreibungsfolgen

In den Klauseln SELECT und ORDER BY entspricht der SQL-Name einer Verknüpfung ohne Punkt der Beziehung <SQL-Name der Verknüpfung>.self, die ihrerseits mit der Beziehung <SQL-Name der Verknüpfung>.<Beschreibungsfolge> identisch ist.

Ein Beispiel:

Die Beschreibungsfolge der Tabelle der Abteilungen und Personen lautet:

```
[Name], [FirstName] ([Phone])
```

Die AQL-Abfrage lautet:

```
SELECT user FROM amPortfolio
```

Die entsprechende Abfrage lautet:

```
SELECT user.self FROM amPortfolio
```

Die ihrerseits entsprechende Abfrage lautet:

```
SELECT (((User.Name + ',') + User.FirstName) + '(' + User.Phone) + ')') FROM amPortfolio
```

## Merkmale

AQL bietet einen direkten Zugriff auf die Merkmale einer Tabelle, als ob es sich um direkte Felder der Tabelle handeln würde. Zum Suchen der Werte eines Merkmals in einer bestimmten Tabelle brauchen Sie nur den SQL-Namen des Merkmals einzugeben und dem Name das Präfix **fv\_** voranzustellen.

Ein Beispiel: Mit der nachstehenden Abfrage werden die Werte des Merkmals mit dem SQL-Namen **fv\_WorkUnit** in der Tabelle der **Abteilungen und Personen (amEmplDept)** gesucht:

```
SELECT fv_WorkUnit FROM amEmplDept
```

## Berechnete Felder

AQL erleichtert die Benutzung der berechneten Felder, die einer Tabelle zugeordnet sind.

Sie müssen lediglich den SQL-Namen des berechneten Felds eingeben und dem Namen das Präfix **cf\_** voranstellen.

## Sortiervorgänge und Indizes

Für Abfragen, bei denen Sortiervorgänge (Klausel ORDER BY) zum Einsatz kommen, lässt Ihnen AQL die Wahl zwischen zwei Vorgehensweisen:

- Modus, bei dem AssetCenter die Verwendung der in der Abfrage eventuell angegebenen Indizes erzwingt, und die Ergebnisse der Suche nach und nach anzeigt.
- Modus, bei dem AssetCenter die Verwendung der in der Abfrage angegebenen Indizes nicht erzwingt. In diesem Fall legt die Datenbank-Engine den Sortiervorgang fest.



Anmerkung:

In der SQL Anywhere-Datenbank bietet AQL diese beiden Modi nicht gleichzeitig an, da die Engine den optimalen Zugriff eigenständig auswählt.

## Beispiel

Im Fall der Abfrage:

```
SELECT lModelId, Brand FROM amModel ORDER BY Brand
```

- Zugriff ohne **Indizes setzen**: Die Datenbank-Engine durchsucht die gesamte Tabelle, ohne dabei den in der Abfrage enthaltenen Index "Marke" zu verwenden. Die Engine sucht nach allen der Abfrage entsprechenden Daten, sortiert sie nach "Marke" und sendet das Ergebnis an den Benutzer. Die Anzeige des Ergebnisses nimmt eine gewisse Zeit in Anspruch.
- Im anderen Modus: Die Datenbank-Engine verwendet den Index "Marke" und zeigt die Ergebnisse im Verlauf der Suche nach und nach an. Die ersten Daten erscheinen also relativ schnell auf dem Bildschirm des Benutzers, doch kann für die vollständige Verarbeitung insgesamt mehr Zeit erforderlich sein.

## Verwendung von Indizes erzwingen

Die Vorgehensweise beim Erzwingen der Verwendung von Indizes ergibt sich aus der Methode, die Sie zum Erstellen der Abfrage anwenden.

### Über die Option "Liste konfigurieren"

Für jede AssetCenter-Liste können Sie die Art des Zugriffs auf die entsprechenden Daten konfigurieren. Dabei spielt es keine Rolle, ob es sich um Hauptlisten oder um Registerkartelisten handelt. Gehen Sie wie folgt vor:

- 1 Setzen Sie den Cursor auf die Liste, für die Sie den Zugriff parametrisieren möchten.
- 2 Drücken Sie die rechte Maustaste.
- 3 Wählen Sie im Kontextmenü die Option **Liste konfigurieren**.
- 4 Markieren Sie auf der Registerkarte **Spalten und Sortieren** das Kontrollkästchen **Indizes setzen**, um die in der Abfrage angegebenen Indizes zu verwenden und die Ergebnisse im Verlauf der Suche nach und nach anzuzeigen. Heben Sie die Markierung auf, wenn Sie eine andere Zugriffsart wünschen.

### In AQL

Wenn Sie eine Abfrage direkt in AQL schreiben, erzwingen Sie die Verwendung der Indizes durch Eingabe der Klausel "FIRST\_ROWS".

Ein Beispiel:

```
SELECT FIRST_ROWS AssetTag FROM amAsset ORDER BY AssetTag
```



Anmerkung:

Die Sortierung nach Systemaufzählungen, beispielsweise in der Tabelle der Merkmale für das Feld **seDataType**, verläuft beim Setzen von Indizes mitunter nicht erfolgreich.

## Sortierreihenfolge

Die Sortierreihenfolge ist von nachstehenden Elementen abhängig:

- Datenbank-Engine
- Eventuelles Erzwingen der Indizes

## Oracle

### Mit Erzwingen der Indizes

- Die Datensätze mit dem Wert NULL werden nicht angezeigt.
- Beim Sortieren werden die Werte der ASCII-Codes berücksichtigt, so dass Groß- und Kleinbuchstaben voneinander unterschieden werden (binärer Sortiervorgang).

### Ohne Erzwingen der Indizes

- Die Datensätze mit dem Wert NULL werden angezeigt.
- Die Groß- und Kleinschreibung wird von Oracle nicht berücksichtigt.

#### Beispiel

Sortieren

<b>Ausgangsliste</b>	A B C D a b NULL NULL
<b>Liste mit Erzwingen der Indizes</b>	A B C D a b
<b>Liste ohne Erzwingen der Indizes</b>	NULL NULL A a B b C D

## Microsoft SQL Server oder Sybase SQL Server

Die Sortierreihenfolge ist von dem Parameter abhängig, der bei der Erstellung der Datenbank definiert wurde. Sie können diese Engines auf Wunsch so konfigurieren, dass sie die Groß- und Kleinschreibung, Sonderzeichen, u. a. berücksichtigen.

## Sybase SQL Anywhere

Bei einer Sybase SQL Anywhere-Datenbank ist das Erzwingen der Indizes über eine AQL-Abfrage nicht möglich.

Die Datenbank-Engine legt selbst die Art des Zugriffs auf die Daten sowie den Sortiervorgang fest.

## Sonstige Hinweise

Bei komplexen Abfragen kann es durchaus schwierig sein, sich für oder gegen das Erzwingen von Indizes zu entscheiden. In diesen Fall ist es empfehlenswert, vor der endgültigen Entscheidung einen Testlauf durchzuführen.

Sie sollten vor allem dann einen Testlauf mit und ohne Erzwingen der Indizes durchführen, wenn Sie mit einer gefilterten Liste arbeiten, und zwar sowohl bei Einsatz von expliziten (einfachen Filtern, Abfragefiltern) als auch von impliziten Filtern (über Zugriffsbeschränkungen).

## Abfrageeditor

AssetCenter bietet einen Abfrageeditor für die Definition und die Anzeige von Abfragen. Er ist vor allem für Datenbankadministratoren und Benutzer mit fortgeschrittenen Kenntnissen gedacht.

### Allgemeine Funktionsweise

Der Abfrageeditor lässt Ihnen beim Schreiben der Abfragen die Wahl zwischen zwei Methoden:

- Verwendung der grafischen Benutzeroberfläche (Unterstützung beim Schreiben der Abfragen)
- Schreiben der Abfragen direkt in AQL

Sie sehen die Transkription Ihrer Arbeit in SQL in jedem Fall in Echtzeit, ohne Rücksicht darauf, ob Sie nun die grafische Methode wählen oder es vorziehen, direkt in AQL zu schreiben bzw. beide Konzepte gleichzeitig nutzen. Das direkte Schreiben der Abfragen in der Abfragesprache SQL ist jedoch nicht möglich.

#### Abbildung 3.1. Abfrageeditoren - Entwicklungsarten



Mit Hilfe des Abfrageeditors können ein fortgeschrittener Benutzer oder ein Administrator AQL-Abfragen erstellen, bearbeiten und löschen. Diese Abfragen sind dann je nach Kontext vom ihm selbst oder von anderen Benutzern verwendbar.



## Zugriff auf den Abfrageeditor

Beim Zugreifen auf den Abfrageeditor haben Sie die Wahl zwischen den nachstehenden Vorgehensweisen:

- Über das Menü **Extras/ Abfragen**. Über dieses Menü können Sie Abfragen erstellen, die anschließend von Ihnen selbst und von anderen Benutzern ohne weiteres verwendet werden können. Die Abfragen werden daraufhin wie folgt ausgeführt:
  - Direkt in dem Fenster, das Sie über das Menü **Extras/ Abfragen** anzeigen oder
  - Über einen Abfragefilter bei der Anzeige der Haupttabelle der Abfrage.
- Über die zahlreichen AssetCenter-Funktionen, die auf Abfragen zugreifen: Zugriffsbeschränkungen, Abfragefilter, Listenkonfiguration, Regeln zur Bewilligung von Anforderungen, Steuerformeln usw.
- Über externe Programme: AssetCenter Export- usw.

Je nach der Stelle in der Software, von der aus Sie auf den Abfrageeditor zugreifen, verfügen Sie über eine mehr oder weniger komplexe Version.

Ein Beispiel: Unsere Abfrage lautet wie folgt:

```
SELECT [FIRST_ROWS] <Feld>[, <Feld>...] FROM <Tabelle> [WHERE <Klausel>]
[ORDER BY <Klausel>]
```

In den gekürzten Versionen des Abfrageeditors (einfache Filter, Abfragefilter...) brauchen Sie nur die Klausel WHERE zu definieren. Die übrigen Komponenten der Abfrage (Ausgangstabelle, Felder...) sind implizit vorhanden. Im Fall eines Abfragefilters entspricht die genannte Tabelle eben der Tabelle, auf die der Filter angewendet wird, und die Felder und Sortierkriterien entsprechen den Spalten und Sortierkriterien, die über das Kontextmenü und die Option **Liste konfigurieren** definiert wurden. Das gleiche gilt für den Abfrageeditor, der über das Menü **Extras/ Abfragen** aufgerufen wird.

Aus der folgenden Abfrage, die vollständig ausgeschrieben wie folgt aussieht:

```
SELECT self FROM amModel WHERE Brand.Name='Compaq'
```

wird Folgendes, wenn ein Abfragefilter auf die Tabelle der Modelle angewendet wird (nur die Klausel WHERE ist explizit):

```
Brand.Name='Compaq'
```

Über das Kontextmenü und die Option **Liste konfigurieren** rufen Sie dagegen eine vollständigere Version des Abfrageeditors auf:

- Die Registerkarte **Spalten und Sortieren** definiert die Felder, die in den Spalten erscheinen sollen, sowie die Sortierkriterien (diese Sortierkriterien entsprechen der Klausel ORDER BY).
- Das Kontrollkästchen **Indizes setzen** ersetzt die Klausel SQL FIRST\_ROWS.
- Die Registerkarte **Filter (WHERE-Klausel)** definiert die Klausel WHERE.
- Die Tabelle ist implizit.

## Abfragen mit Hilfe des Abfrageditors erstellen

Wenn Sie eine Abfrage mit Hilfe des Abfrageditors erstellen möchten, wählen Sie zunächst das Menü **Extras/ Abfragen**. Daraufhin erscheint ein Fenster mit zwei Registerkarten, **Filter (WHERE-Klausel)** und **Vorschau**:

- Die Registerkarte **Filter (WHERE-Klausel)** ist eine grafische Benutzeroberfläche, auf der die Filterkriterien für Ihre Abfrage festgelegt werden. In dieser Registerkarte definieren Sie die Elemente der SQL-Klausel WHERE.
- Die Registerkarte **Vorschau** zeigt die Transkription Ihrer Eingaben in SQL an und ermöglicht so die Prüfung der Abfrage.

### Schritt 1: Eingabe der erforderlichen Informationen in der oberen Hälfte des Abfragedetails

Die Angabe der Tabelle, über die die Abfrage erstellt werden soll, ist obligatorisch.

Wenn die von Ihnen erstellte Abfrage auch für andere Benutzer verwendbar sein soll, müssen Sie die Markierung des Kontrollkästchens **Nicht frei** (SQL-Name: bPrivate) aufheben.

---

#### Anmerkung:

Der Administrator greift auf alle Abfragen der Datenbank zu, also auch auf die Abfragen, für die das Kontrollkästchen **Nicht frei** markiert ist.

---


Sobald Sie die grundlegenden Informationen für die Abfrage eingegeben haben, klicken Sie auf die Schaltfläche **Erstellen**, um auf die Registerkarten des Abfragedetails zuzugreifen.

## Schritt 2: Definition der Filterkriterien auf der Registerkarte Filter (WHERE-Klausel)

Der AssetCenter-Abfrageeditor ermöglicht die Verwendung von Kriterien, die sich auf bestimmte Felder beziehen, benutzt mathematische Ausdrücke und integriert Konstanten und Operatoren.

Sie können ein oder mehrere Filterkriterien definieren.

Gehen Sie zum Definieren eines Filterkriteriums wie folgt vor:


- 1 Wählen Sie ein Feld, eine Konstante oder einen Ausdruck (**Feld 1**) in der Ausgangstabelle und führen Sie einen Vergleich mit einem Feld, einer Konstanten oder einem Ausdruck durch (**Feld 2**).
- 2 Bestätigen Sie das Filterkriterium, indem Sie es durch Klicken auf die Schaltfläche  in die untere Hälfte des Bildschirms übertragen.
- 3 Bestätigen Sie die Abfrage durch Klicken auf die Schaltfläche **Ändern** im Abfragedetail.

Gehen Sie zum Definieren mehrerer Filterkriterien, die durch die logischen Operatoren AND und OR miteinander verbunden sind, wie folgt vor:

- 1 Erstellen Sie ein erstes Filterkriterium, wie oben beschrieben.
- 2 Definieren Sie die anderen Kriterien und bestätigen Sie diese Kriterien durch Klicken auf die Schaltfläche **UND** bzw. **ODER**.
- 3 Bestätigen Sie die Abfrage durch Klicken auf die Schaltfläche **Ändern** im Abfragedetail.



**Anmerkung:**

Sollten die eingegebenen Filterkriterien nicht Ihren Anforderungen entsprechen, klicken Sie auf die Schaltfläche , um den Inhalt des Fensters zu löschen oder den AQL-Code direkt zu ändern.




**Anmerkung:**

Wenn Sie den grafischen Assistenten nicht benutzen wollen, können Sie die Abfrage in der unteren Hälfte der Registerkarte **Filter (WHERE-Klausel)** direkt in AQL eingeben.

---

## Schritt 3: Testen der Abfrageausführung

Gehen Sie zum Testen der Abfrage und zum Anzeigen ihrer Transkription in SQL wie folgt vor:

- 1 Setzen Sie den Cursor auf die Registerkarte **Vorschau** des Abfragedetails.
- 2 Klicken Sie auf das Symbol  : AssetCenter zeigt eine Vorschau des Abfrageergebnisses in Form einer Datensatzliste an. Die Anzahl der Datensätze, die der Abfrage entsprechen, erscheint unten rechts im Fenster.



Anmerkung:


Der auf der Registerkarte **Vorschau** erscheinende SQL-Code kann nicht auf dieser geändert werden.

## Bei Abfragen verwendete Felder

Beim Definieren der Filterkriterien für eine Abfrage greifen Sie auf folgende Elemente zurück:


- Ein Feld der Tabelle, in der die Abfrage ausgeführt werden soll
- Ein verknüpftes Feld
- Merkmale, die der Tabelle zugeordnet sind

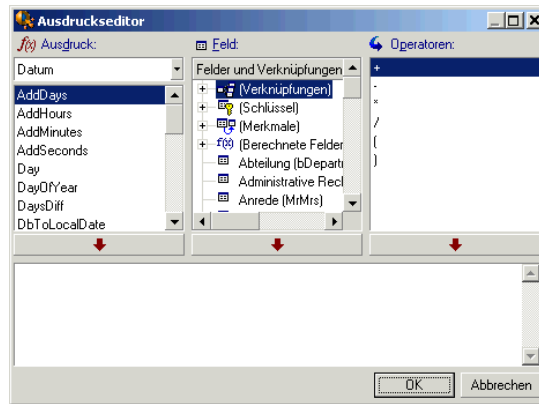
## Ausdrücke schreiben

Mit Hilfe der Ausdrücke  können Sie mathematische Operationen in Ihrer Abfrage ausführen. So verwenden Sie beispielsweise die Funktion "Count", um die Anzahl der Datensätze in einem Abfrageergebnis zu zählen.


Beim Schreiben von Ausdrücken haben Sie die Wahl zwischen zwei Vorgehensweisen:

- Geben Sie den Ausdruck direkt im entsprechenden Feld ein.
- Verwenden Sie den AssetCenter-Ausdruckseditor.


Wenn Sie den Ausdruckseditor verwenden möchten, müssen Sie auf der Registerkarte **Filter (WHERE-Klausel)** im Abfragedetail neben dem Eingabebereich auf die Schaltfläche  klicken.



Der Ausdruckseditor besteht aus drei Spalten:

- In der Spalte "Ausdruck" erscheinen die vorhandenen AQL-Funktionen. Mit einem Mausklick auf die Schaltfläche  wenden Sie einen Filter auf die Liste der AQL-Funktionen an. Dabei sind nachstehende Typen zu unterscheiden: "Summen", "String", "Datum", "Numerisch", "Test".
- Der Spalte "Feld" entnehmen Sie die verschiedenen Felder, die zur Abfrage herangezogen werden können.
- Die Spalte "Operatoren" führt die im Ausdruck verwendbaren Operatoren auf.

Gehen Sie zum Einfügen eines "Ausdrucks", "Felds" oder "Operators" in den Ausdruck wie folgt vor:

- 1 Wählen Sie den Ausdruck, das Feld oder den Operator.
- 2 Klicken Sie auf .


Sobald Sie den Ausdruck fertig geschrieben haben, klicken Sie auf **OK**, um ihn auf die Registerkarte **Filter (WHERE-Klausel)** im Abfragedetail zu übertragen.

## Konstanten


Bei den Konstanten *k* handelt es sich um feste Werte, die Sie den Auswahlkriterien zuordnen. Ein Beispiel: Sie suchen alle Vermögensgegenstände

der Marke **3Com**. Ordnen Sie dem verknüpften Feld **Brand.Name** der Tabelle der Modelle die Konstante **3Com** zu.

Gehen Sie zum Auswählen der Konstante wie folgt vor:

- 1 Klicken Sie auf das Symbol .
- 2 Daraufhin erscheint ein Auswahlfenster mit den Werten, die in der Datenbank für das als Suchkriterium definierte Feld der Tabelle vorhanden sind.

 **Anmerkung:**

Auch bei Feldern vom Typ "Aufzählung" erscheinen in dem nach Klicken auf das Symbol  angezeigten Fenster nur die Werte, die in der Datenbank verwendet werden.

## Syntax der Programmiersprache AQL

Für die Programmierung in AQL sind gute SQL-Kenntnisse erforderlich. Eine ausführliche Beschreibung der SQL-Sprache würde den Rahmen dieses Handbuchs sprengen. Wir empfehlen Ihnen daher das Lesen der entsprechenden Referenzliteratur.

## Konventionen

In der nachstehenden Tabelle finden Sie die beim Schreiben der AQL-Syntax gültigen Konventionen:

**Tabelle 3.1. AQL - Syntaxkonventionen**

[ ]	Diese Klammern umschließen ein optionales Element. Sie müssen nicht manuell eingegeben werden.
< >	Diese Klammern umschließen ein logisches Element. Sie müssen nicht manuell eingegeben werden.
	Der vertikale Strich verweist auf exklusive Auswahlen.

...

Weist darauf hin, das sich der davorstehende Text ein oder mehrere Male wiederholen kann.

FROM

Wörtliche Ausdrücke stehen in Großbuchstaben.

## Abfragesyntax

### Einfache Abfragen

**SELECT** [DISTINCT] [FIRST\_ROWS] <Auswahlliste>

[Klausel FROM [Seite 54]]

[Klausel WHERE [Seite 55]]

[Klausel GROUP BY [Seite 56]]

[Klausel HAVING [Seite 57]]

[Klausel ORDER BY [Seite 58]]

### Nebenabfragen

AQL unterstützt die Verwendung von Nebenabfragen anstelle von Feldern.

 Anmerkung:

In den Nebenabfragen ist zusammen mit dem Befehl SELECT nur ein einziger Ausdruck zulässig.

**(SELECT** [DISTINCT] <Ausdruck>

[Klausel FROM [Seite 54]]

[Klausel WHERE [Seite 55]]

[Klausel GROUP BY [Seite 56]]

[Klausel HAVING [Seite 57]]

)

 **WARNUNG:**

Die Nebenabfragen müssen unbedingt in Klammern gesetzt sein.

Verwendungsbeispiel:

```
SELECT Self FROM amAsset WHERE mPrice >= (SELECT Max(mPrice)/2 FROM amAsset)
```

## Abfragen vom Typ UNION

Mit der Klausel **UNION** können Sie die Ergebnisse mehrerer Abfragen zusammenfassen:

```
SELECT <Auswahlliste>
[Klausel FROM [Seite 54]]
[Klausel WHERE [Seite 55]]
[Klausel GROUP BY [Seite 56]]
[Klausel HAVING [Seite 57]]
[ UNION | UNION ALL | INTERSECTS | MINUS]
SELECT <Auswahlliste>
[Klausel FROM [Seite 54]]
[Klausel WHERE [Seite 55]]
[Klausel WHERE [Seite 55]]
[Klausel GROUP BY [Seite 56]]
[Klausel HAVING [Seite 57]]...
[Klausel ORDER BY [Seite 58]]
```

## Elemente einer Abfrage

### Felder und Verknüpfungen

Bei Abfragen kommen Felder und Verknüpfungen der AssetCenter-Datenbank zur Anwendung.

Geben Sie den Namen eines Felds wie folgt ein:

- In Bezug auf die Ausgangstabelle der Abfrage. In diesem Fall ist es nicht notwendig, den Namen dieser Tabelle zu erwähnen:

```
[Verknüpfung. ...[Verknüpfung.]]<Feld>
```

Beispiele anhand der Tabelle der **Ausrüstungselemente (AmPortfolio)**:

```
Model
User.Name
User.Location.Name
```



- Den direkten Namen. In diesem Fall müssen Sie den Namen der Tabelle eingeben, aus der das Feld stammt:
  - Geben Sie die Tabelle in der Klausel **FROM** ein, und verwenden Sie dabei den Namen der Tabelle oder den entsprechenden Alias-Namen, sofern vorhanden:
    - <Tabelle.[Verknüpfung...]Feld>
    - <alias.[Verknüpfung...]Feld>
  - Geben Sie die Tabelle nicht in der Klausel **FROM** ein, und verwenden Sie ":" :
    - <Tabelle:[Verknüpfung...]Feld>
    - <Tabelle[\_Alias]:[Verknüpfung[\_Alias]...]Feld>

Diese beiden Notationen sind von besonderem Nutzen, wenn Sie nicht auf die Klausel **FROM** zugreifen können.

Ein Beispiel: Beim Schreiben einer Abfrage in AssetCenter können Sie nur auf die Klausel **WHERE** zugreifen. Die Ausgangstabelle der Abfrage ist implizit vorhanden (Tabelle, auf die ein Filter angewendet wird, Feld **Tabelle** (**TableName**) im Detailfenster einer Abfrage...). Dabei kann sich jedoch die Verwendung anderer Tabellen in der Abfrage als notwendig erweisen. In diesem Fall verwenden Sie die Notation ":".

## Konstanten

Im Folgenden finden Sie die gültige Syntax für Konstanten, die in Abfragen zur Anwendung kommen.

### Numerische Konstanten

Das Trennzeichen für Dezimalstellen ist ein Punkt.

Beispiele:

12

52.23

### Konstanten vom Typ Text

Der Text ist jeweils mit einem Apostroph abgegrenzt.

Beispiele:

'Rechner'

'Bildschirm'

## Konstanten vom Typ Datum oder Uhrzeit

Die Konstanten vom Typ Datum oder Uhrzeit sind mit dem Zeichen # abgegrenzt. Für das Format gelten folgende Regeln:

- Jahreszahlen werden mit 4 Ziffern ausgedrückt.
- Datumsangaben werden in der Reihenfolge Jahr-Monat-Tag ausgedrückt.
- Uhrzeiten werden in der Reihenfolge Uhrzeit-Minuten-Sekunden ausgedrückt.
- Die Stunden reichen von 0 bis 24 Uhr (und nicht von 0 bis 12 Uhr am/pm).
- Zum Trennen von Datumsangaben werden die Zeichen "/" oder "-" verwendet.
- Zum Trennen von Uhrzeiten wird das Zeichen ":" verwendet.
- Monaten, Tagen, Stunden, Minuten und Sekunden wird gegebenenfalls eine 0 vorangestellt.
- Wenn Sie sowohl das Datum als auch die Uhrzeit angeben, erscheint das Datum grundsätzlich vor der Uhrzeit. Die beiden Angaben sind durch ein Leerzeichen voneinander getrennt.

Beispiel:

#yyyy-mm-dd hh:mm:ss#

#yyyy-mm-dd#

#hh:mm:ss#

#2004-01-01 01:00:03#

## Ausdrücke

Die Ausdrücke setzen sich aus folgenden Elementen zusammen:

- Konstanten
- Felder
- Funktionen
- Nebenabfragen

Sie können diese Elemente mit Hilfe von Operatoren und Klammern zu komplexen Ausdrücken zusammensetzen.

Vergleichsoperatoren weisen folgende Struktur auf:

<Ausdruck> <Vergleichsoperator> <Ausdruck>

Logische Ausdrücke weisen folgende Struktur auf:

<Vergleichsausdruck> <AND | OR> <Vergleichsausdruck>

Die Klammern dienen zum Zusammenfassen mehrerer logischer Ausdrücke.

## Operatoren

### Logische Operatoren

Die logischen Operatoren dienen zum Verbinden zweier Bedingungen.

**Tabelle 3.2. AQL - Logische Operatoren**

Operator	Bedeutung
AND	Logisches UND
OR	Logisches ODER

Zum Optimieren einer Abfrage ist die Verwendung eines Vergleichsoperators dem Einsatz eines logischen Operators vorzuziehen. Das folgende Beispiel zeigt die Optimierung eines Abfragefilters bei der Auswahl von Ausrüstungsgegenständen, deren Feld **Zuordnung** (**seAssignment**) den Wert **Ausstehende Lieferung** oder **Rückgabe** (**Wartung**) aufweist. Die beiden Elemente der Systemaufzählung weisen respektive den Wert "3" und "4" auf. Sie können also den nachstehenden Ausdruck schreiben:

```
(seAssignment=3) OR (seAssignment =4)
```

Da jedoch der letzte Wert der Systemaufzählung dem Wert "4" entspricht, ist es vorzuziehen, die Abfrage wie folgt zu schreiben:

```
seAssignment >=3
```

### Vergleichsoperatoren

Diese Operatoren dienen zum Vergleichen zweier Ausdrücke.

**Tabelle 3.3. ASL - Vergleichsoperatoren**

Operator	Bedeutung
=	Gleich
<>	Ungleich
=!	

Operator	Bedeutung
>	Größer als
<	Kleiner als
>=	Größer gleich
=<	Kleiner gleich
=*	Externe Beziehung rechts. Aufgrund der Verfahrensweise von AQL bei der Verwaltung von Verknüpfungen ist der Gebrauch des Operators eingeschränkt.
*=	Externe Beziehung links. Aufgrund der Verfahrensweise von AQL bei der Verwaltung von Verknüpfungen ist der Gebrauch des Operators begrenzt.
LIKE	Die Funktionsweise ist identisch mit dem Operatoren = und ermöglicht den Einsatz von Jokerzeichen.
NOT LIKE	<p>Es stehen Ihnen folgende Jokerzeichen zur Verfügung:</p> <p>"%" ersetzt eine beliebige Zeichenfolge.</p> <p>"_" ersetzt ein beliebiges Zeichen.</p> <p>Je nach der Leistungsfähigkeit Ihrer Datenbank-Engine stehen Ihnen folgende Zeichen zur Verfügung. SQL Anywhere, SQL Server und Sybase unterstützen diese Zeichen, Oracle for WorkGroups unterstützt diese Zeichen nicht:</p> <p>[abc...] ermöglicht das Definieren einer Liste mit den für ein Zeichen möglichen Werten. Leerzeichen zwischen diesen Werten sind nicht zulässig.</p> <p>[a-c] ermöglicht das Definieren eines für ein Zeichen möglichen Wertebereichs.</p> <p>DB2 unterstützt die Verwendung des Operators LIKE X nicht, wenn X einen SQL-Spaltennamen enthält. Dieser Operator unterstützt nur Konstanten. Die folgende Abfrage ist daher beispielsweise für DB2 nicht korrekt:</p> <p><b>SELECT COL1, COL2 FROM TABLE1 WHERE COL1 LIKE COL2</b></p>
IS NULL	Mit diesem Operator stellen Sie fest, ob ein Feld dem Wert "NULL" entspricht.
IS NOT NULL	Achtung: AssetCenter erlaubt den Wert "NULL" nur in leeren Feldern vom Typ <b>Text</b> und für Felder vom Typ <b>Datum</b> bzw. <b>Datum + Uhrzeit</b> , für die keine Eingaben gemacht wurden.

 **Anmerkung:**

SQL Anywhere ist nicht in der Lage, Klauseln vom Typ "LIKE X" zu bearbeiten, wenn X mehr als 128 Zeichen umfasst. Ist dies der Fall, wird eine ODBC-Fehlermeldung angezeigt. Dieses Problem kann bei der Anzeige von Listen als Baumstruktur auftreten, da bei diesem Vorgang die Klausel "LIKE" zusammen mit dem Feld "FullName" verwendet wird.

## Spezifische Operatoren für Nebenabfragen

Mit Hilfe der nachstehenden Operatoren lassen sich die Werte mit dem Ergebnis einer Nebenabfrage vergleichen:

- = **ANY (Nebenabfrage).**
- = **ALL (Nebenabfrage).**
- = **SOME (Nebenabfrage).**

Beispiel:

- Mit der nachstehenden Abfrage stellen Sie die Liste der Ausrüstungselemente zusammen, deren Marke am Standort Hamburg verwendet wird:

```
SELECT lModelId, Model.Brand FROM amPortfolio WHERE Model.Brand = ANY
(SELECT Model.Brand FROM amPortfolio WHERE Location.FullName = '/Hambu
rg')
```

## Auswahlliste

Über die Auswahllisten definieren Sie die Elemente, die extrahiert oder angezeigt werden sollen. Zusätzlich legen Sie den Befehl SELECT in den Abfragen fest. Eine Auswahlliste besteht aus einem oder mehreren Ausdrücken, die durch ein Komma voneinander getrennt sind:

**<Ausdruck> [,<Ausdruck>...]**

Jeder Ausdruck kann mit einem Alias-Namen verknüpft sein. Beispiel:

```
SELECT MrMrs, (Name + FirstName) Identity FROM amEmplDept
```

Das ist besonders bei Exportabfragen von Nutzen, um den exportierten Spalten einen Namen zuzuordnen.



Anmerkung:

In einigen DBMS ist die Anzahl der Ausdrücke beschränkt, die im Befehl **SELECT** enthalten sein dürfen.

## Klausel FROM

Die Klausel **FROM** verweist auf eine oder mehrere Tabellen, für die der Befehl **SELECT** ausgeführt werden soll.

AQL ermöglicht die Verwendung von Alias-Namen für Tabellennamen.

### Syntax

```
FROM <Tabellenname> [Tabellenalias] [, <Tabellenname> [Tabellenalias]>
... ]
```

## Ausgangstabelle einer Abfrage

Die erste Tabelle, die in der Klausel **FROM** einer Abfrage erscheint, entspricht der Ausgangstabelle.

Wenn eine Abfrage auf ein Feld zugreift, für das keine Tabelle angegeben ist, geht AQL davon aus, dass es sich bei der Tabelle, aus der das Feld stammt, um die Ausgangstabelle der Abfrage handelt. Die AQL-Klausel **FROM** unterscheidet sich in diesem Sinn von der gleichnamigen SQL-Klausel.

In der nachfolgenden Abfrage sucht AQL das Feld **Interner Code (AssetTag)** in der Tabelle **Vermögen und Lose (amAsset)**:

```
SELECT AssetTag FROM amAsset
```

## Anzahl der Tabellen in einer Abfrage

Die Anzahl der Tabellen, die in einer Abfrage verwendet werden können, ist abhängig von der jeweils verwendeten Datenbank-Engine.

Beispiel:

- Oracle: Die Anzahl der verwendeten Tabellen ist keinerlei Einschränkungen unterworfen.
- Microsoft SQL Server oder Sybase SQL Server: Die Abfragen sind auf 16 Tabellen beschränkt.

**!** WARNUNG:

Wenn Sie die in einer Abfrage verwendeten Tabellen zählen, müssen Sie auch die Tabellen berücksichtigen, die nicht ausdrücklich genannt sind. Das ist insbesondere der Fall bei der Verwendung von Verknüpfungen. Achten Sie ebenfalls auf die Notation "fv\_" (Suche nach Werten von Merkmalen), mit der eine zusätzliche Beziehung für die SQL-Datenbank-Engine generiert wird. Mit Hilfe der Notation "cf\_" (berechnete Felder) können zusätzliche Beziehungen generiert werden.

## Beispiele

```
FROM amPortfolio
FROM amPortfolio a, amLocation l
```

Die nachfolgenden Abfragen sind identisch:

```
SELECT AssetTag FROM amAsset
SELECT a.AssetTag FROM amAsset a
SELECT amAsset.AssetTag FROM AmAsset
```

## Klausel WHERE

Die AQL-Klausel **WHERE** entspricht der gleichnamigen SQL-Klausel. Diese Klausel definiert die Suchbedingungen (zu extrahierende Elemente der Datenbank). Die Bedingungen können auch mit **HAVING**-Klauseln ausgedrückt werden.

### Syntax

**WHERE** <Suchbedingungen>

## Suchbedingungen schreiben

In den meisten Fällen müssen Sie die Bedingungen wie folgt schreiben:

```
<WHERE | HAVING> [NOT] <Ausdruck> <Vergleichsoperator> <Ausdruck>
<WHERE | HAVING> [NOT] <logischer Ausdruck>
<WHERE | HAVING> [NOT] <Feld> [NOT] LIKE 'xxxxx'
<WHERE | HAVING> [NOT] <logischer Ausdruck> <AND | OR> <logischer Ausdruck>
<WHERE | HAVING> [NOT] <Feld> IS [NOT] NULL
```

In den anderen Fällen müssen Sie komplexere Bedingungen schreiben, wie z. B.:

```
<WHERE | HAVING> [NOT] EXISTS (<Nebenabfrage>)
<WHERE | HAVING> [NOT] <Ausdruck> [NOT] IN (<Liste der Werte> | <Nebenabfrage>)
<WHERE | HAVING> [NOT] <Ausdruck> <Vergleichsoperator> <ANY | ALL> (<Nebenabfrage>)
```

## Klausel GROUP BY

Die AQL-Klausel **GROUP BY** entspricht der gleichnamigen SQL-Klausel.

### Syntax

**GROUP BY** <Ausdruck ohne Agregate> [, <Ausdruck ohne Agregate>]...

## Ratschläge für das Verfassen

Mit der Klausel **GROUP BY** legen Sie die Nebengruppen der Tabelle fest. Die Nebengruppen werden in der Klausel **GROUP BY** über einen Ausdruck definiert, bei dem es sich beispielsweise um einen Feldnamen handelt.

Wenn die Aggregatfunktionen in der Auswahlliste mit dem Befehl **SELECT** enthalten sind, sucht die Klausel **GROUP BY** den Wert, der sich dabei für jede Nebengruppe ergibt. Diese Werte können dann in einer **HAVING**-Klausel verwendet werden.

Wenn eine Abfrage die Klausel **GROUP BY** verwendet, muss jeder Ausdruck der Auswahlliste einen einzigen Wert für die Nebengruppe liefern.

## GROUP BY - Beispiele

Mit der nachstehenden Abfrage erhalten Sie die Anzahl der insgesamt in der Datenbank vorhandenen Marken. Für jeden Vermögensgegenstand, dem eine Marke zugeordnet ist, gibt AssetCenter den Eintrag der Marke zurück.

```
SELECT Count (Model.Brand.Name) FROM amAsset
```

Wenn Sie die Klausel **GROUP BY** verwenden, erhalten Sie die Liste der Marken sowie die Anzahl der Vermögensgegenstände für jede Marke:

```
SELECT Model.Brand.Name, count (lAstId) FROM amAsset GROUP BY Model.Brand
```



## Klausel HAVING

Die AQL-Klausel **HAVING** entspricht der gleichnamigen SQL-Klausel.

### Syntax

**HAVING** <Suchbedingungen>

## Unterschiede zur Klausel WHERE

Wie für die Klausel **WHERE** werden auch mit der Klausel **HAVING** Suchbedingungen festgelegt. Die beiden Bedingungen weisen jedoch folgende Unterschiede auf:

- Die Klausel **HAVING** definiert die Einschränkungen, die auf die Aggregatfunktionen der Auswahlliste angewendet werden sollen. Die Einschränkungen betreffen die Anzahl der Ergebniszeilen, wirken sich jedoch nicht auf die mit den Aggregatfunktionen verknüpften Berechnungen aus.
- Wird eine Abfrage die Klausel **WHERE** verwendet, gelten die Einschränkungen durch die Suchbedingungen für die Zeilen, mit denen die Aggregatfunktionen rechnen, wirken sich jedoch nicht auf die Ergebniszeilen aus.

## Beispiele

Beispiel einer Abfrage, bei der die Klausel **WHERE** der Klausel **HAVING** entspricht:

Die nachstehende Abfrage gibt die Liste der Marken zurück, deren Namen mit einem Buchstaben nach dem Buchstaben **B** beginnt, sowie die Anzahl der Vermögensgegenstände für jede dieser Marken:

```
SELECT Model.Brand.Name, count(lAstId) FROM amAsset GROUP BY Model.Brand.Name HAVING Model.Brand.Name > 'B'
```

Sie können dieselbe Bedingung auch mit Hilfe der Klausel **WHERE** ausdrücken:

```
SELECT Model.Brand.Name, count(lAstId) FROM amAsset WHERE Model.Brand.Name > 'B' GROUP BY Model.Brand.Name
```

Beispiel für eine Abfrage mit der Klausel **HAVING**:

Die Klausel **HAVING** ermöglicht im Gegensatz zur Klausel **WHERE** die Verwendung von Aggregatfunktionen, wie **Count**. Damit werden bei der

nachstehenden Abfrage alle Marken gesucht, für die die Anzahl der Vermögensgegenstände größer ist als 1:

```
SELECT Model.Brand.Name, count(lAstId) FROM amAsset GROUP BY Model.Brand.Name HAVING count(Model.Brand) > 1
```

## Klausel ORDER BY

Die AQL-Klausel **ORDER BY** entspricht der gleichnamigen SQL-Klausel.

Die Elemente können wie folgt sortiert werden:

- In aufsteigender Reihenfolge: **ASC**. Dabei handelt es sich um den Standardwert.
- In absteigender Reihenfolge: **DESC**.

### Syntax

**ORDER BY** <Ausdruck> [**ASC** | **DESC**] [, <Ausdruck> [**ASC** | **DESC**]...]

## Klausel INSERT

Diese Klausel dient dazu, einen Datensatz oder mehrere Datensätze in eine Tabelle der Datenbank einzufügen.

### Syntax

**INSERT INTO** <Tabellenname> [Tabellenalias] (<Feldname> [, <Feldname>]...) **VALUES** (<Ausdruck> [, Ausdruck]...) | **AQL-Unterabfrage**)

Die Klausel ist im AssetCenter-API **AmDbExecAql** enthalten.

Weitere Informationen zu den AssetCenter-APIs finden Sie im Handbuch Programmier's Referenz, Kapitel **Alphabetical Reference**.

## Beispiel

Die Klausel **INSERT** kann den Code des Assistenten **Zusätzliche Empfangsinformationen** vereinfachen.

### Assistentencode ohne die Klausel INSERT

```
hrAlarm = AmCreateRecord("amDateAlarm")
lErr = AmSetFieldLongValue(hrAlarm, "bSecondLevel", 0)
```

```

        lErr = AmSetFieldLongValue(hrAlarm, "dtTrig1", AmGetFieldLongValue(hrAsset, 2) - lDaysBefore*86400)
        lErr = AmSetFieldLongValue(hrAlarm, "lAction1Id", lActionId)
        lErr = AmSetFieldLongValue(hrAlarm, "lMonitObjId", lAstId)
        lErr = AmSetFieldStrValue(hrAlarm, "MonitoredField", "dWarrEnd")
        lErr = AmSetFieldStrValue(hrAlarm, "MonitoredTable", "amAsset")
        lErr = AmSetFieldLongValue(hrAlarm, "sDaysBefore1", lDaysBefore)
        lErr = AmInsertRecord(hrAlarm)

```

## Assistentencode mit der Klausel INSERT

```

lErr = AmDbExecAql("insert into amDateAlarm (bSecondLevel, dtTrig1, lActionId, lMonitObjId, MonitoredField, MonitoredTable, sDaysBefore1) values ( 0, " & AmGetFieldLongValue(AmGetFieldLongValue(hrAsset, 2) - lDaysBefore*86400 & ", " & lAstId & ", 'dWarrEnd', 'amAsset', " & lDaysBefore & ")")

```

## Klausel UPDATE

Mithilfe dieser Klausel können ein oder mehrere Datensatzfelder in einer Tabelle der Datenbank aktualisiert werden.

### Syntax

**UPDATE** <Tabellenname> [Tabellenalias] **SET** (<Feldname> [, <Feldname>...]) [[Klausel FROM](#) [Seite 54]] [[Klausel WHERE](#) [Seite 55]]

### Beispiel

Durch Verwendung der Klausel **UPDATE** kann der Code einer Aktion zum Auslösen einer Befehlsaktion vereinfacht werden.

## Aktionscode ohne die Klausel UPDATE

```

hr = AmGetRecordFromMainId("amPOrder", [lPordId])
    lErr = AmSetFieldLongValue(hr, "seStatus", "$(IDS_POSTSTATUS_ORDERED) ")
    lErr = AmUpdateRecord(hr)

```

## Aktionscode mit der Klausel UPDATE

```
lErr = AmDbExecAql("update amPOrder set seStatus = 21 where lPordId =
" &
[lPordId])
```

## Klausel DUPLICATE

Diese Klausel dient dazu, einen vorhandenen Datensatz in einer Tabelle der Datenbank zu duplizieren.

Diese Funktion ist AssetCenter eigen.

Weitere Informationen dazu finden Sie im Handbuch **Einführung in AssetCenter**, Abschnitt **Mit Datensätzen arbeiten**, Unterabschnitt **Duplizieren von Datensätzen**.

### Syntax

**DUPLICATE** <Tabellenname> [Tabellenalias] **SET** (<Feldname> [, <Feldname>...]) [[Klausel FROM](#) [Seite 54]] [[Klausel WHERE](#) [Seite 55]]

## Klausel DELETE

Diese Klausel dient zum Löschen der Felder eines Datensatzes in einer Tabelle der Datenbank.

### Syntax

**DELETE** [[Klausel FROM](#) [Seite 54]] [[Klausel WHERE](#) [Seite 55]]

## AQL-Funktionen

Im Folgenden finden Sie eine Liste der allgemein verwendeten AQL-Funktionen, die Sie in Abfragen und Formeln verwenden können:

- AQL-Funktionen vom Typ Aggregat
- AQL-Funktionen vom Typ String
- AQL-Funktionen vom Typ Datum
- AQL-Funktionen vom Typ Numerisch

- AQL-Funktionen vom Typ Test



Anmerkung:

Sie können auch die SQL-Funktionen Ihrer Datenbank-Engine verwenden. Dabei geht jedoch die Portabilität zwischen den Datenbank-Engines verloren.

## AQL-Funktionen vom Typ Aggregat

**Tabelle 3.4. AQL - Funktionen vom Typ Aggregat**

<b>Funktion</b>	<b>Beschreibung</b>
Avg( <Spalte> )	Gibt den mittleren Wert einer Spalte vom Typ "Zahl" zurück. Gibt 0 zurück, wenn die Spalte keinen Datensatz enthält.
Count( <Spalte> )	Zählt die Werte in einer Spalte, die ungleich Null sind.
Countdistinct( <Spalte> )	Zählt die unterschiedlichen Werte in einer Spalte, die ungleich Null sind.
Max( <Spalte> )	Gibt den maximalen Wert einer Spalte vom Typ "Zahl", "Text" oder "Datum" zurück. Ist in der Spalte kein Datensatz vorhanden, wird der Wert 0 (Spalte vom Typ "Zahl"), "Leerer String" (Spalte vom Typ "Text"), oder "Leeres Datum" (Spalte vom Typ "Datum") zurückgegeben.
Min( <Spalte> )	Gibt den Mindestwert einer Spalte vom Typ "Zahl", "Text" oder "Datum" zurück. Ist in der Spalte kein Datensatz vorhanden, wird der Wert 0 (Spalte vom Typ "Zahl"), "Leerer String" (Spalte vom Typ "Text"), oder "Leeres Datum" (Spalte vom Typ "Datum") zurückgegeben.
Sum( <Spalte> )	Gibt die Summe der Werte einer Spalte vom Typ "Zahl" zurück. Gibt den Wert 0 zurück, wenn in der Tabelle kein Datensatz vorhanden ist.

Diese Funktionen kommen zusammen in den Klauseln "GROUP BY" und "HAVING" zur Anwendung.

## AQL-Funktionen vom Typ String

**Tabelle 3.5. AQL - Funktionen vom Typ String**

<b>Funktion</b>	<b>Beschreibung</b>
Ascii( <String> )	Gibt den ASCII-Wert des ersten Zeichens der <Folge> zurück.
Char(<n>)	Gibt das ASCII-Codezeichen n zurück.
Length( <Zeichenfolge> )	Gibt die Länge der <Zeichenfolge> zurück.
Left( <String>, <n> )	Gibt die "n" ersten Zeichen der <Folge> zurück.
Lower( <String> )	Gibt die <Folge> in Kleinbuchstaben zurück.
Ltrim( <String> )	Entfernt die Leerstellen links neben der <Folge>.
Right( <String>, <n> )	Gibt die "n" letzten Zeichen der <Folge> zurück.
Rtrim( <String> )	Entfernt die Leerstellen rechts neben der <Folge>.
Substring( <String>, <n1>, <n2> )	Extrahiert die Nebenkette, die mit dem Zeichen "n1" der <Folge> beginnt und die Länge "n2" aufweist (das 1. Zeichen der <Folge> wird mit einer 1 numeriert).
Upper( <String> )	Gibt die <Folge> in Großbuchstaben zurück.

## AQL-Funktionen vom Typ Datum

**Tabelle 3.6. AQL - Funktionen vom Typ Datum**

<b>Funktion</b>	<b>Beschreibung</b>
Year( <Datum> )	Gibt die Zahl für das Jahr zurück, das im Feld vom Typ "Date" oder "Date and time" erscheinen soll (Bsp.: 2000).
Month( <Datum> )	Gibt die Zahl für den Monat zurück, der im Feld vom Typ "Date" oder "Date and time" erscheinen soll (1=Januar, ..., 12=Dezember).
Day( <Datum> )	Gibt die Zahl für den Tag des Monats zurück, der im Feld vom Typ "Date" oder "Date and time" erscheinen soll (1-31).
DayOfYear( <Datum> )	Gibt die Zahl für den Tag im Jahr zurück, der im Feld vom Typ "Date" oder "Date and time" erscheinen soll (1-366).

<b>Funktion</b>	<b>Beschreibung</b>
WeekDay( <Datum> )	Gibt die Zahl für den Tag in der Woche zurück, der im Feld vom Typ "Date" oder "Date and time" erscheinen soll.  Diese Zahl ist abhängig von der Konfiguration des Servers. Ein Beispiel: Die Standardkonfiguration unter Sybase oder Microsoft SQL Server lautet (1=Sonntag, 2=Montag, ..., 7=Samstag). Die Standardkonfiguration unter Oracle lautet (1=Montag, ..., 7=Sonntag).
Hour( <Uhrzeit> )	Gibt die Zahl für die Stunde zurück, die im Feld vom Typ "Time" oder "Date and time" erscheinen sollen (0-23).
Minute( <Uhrzeit> )	Gibt die Zahl für die Minuten zurück, die im Feld vom Typ "Time" oder "Date and time" erscheinen sollen (0-59).
Second( <Uhrzeit> )	Gibt die Zahl für die Sekunden zurück, die im Feld vom Typ "Time" oder "Date and time" erscheinen sollen (0-59).
Getdate()	Gibt das aktuelle Systemdatum des Servers zurück.
AddDays( <Datum>, <Anzahl> )	Fügt eine bestimmte Anzahl von Tagen in einem Feld vom Typ "Date" oder "Date and time" hinzu.
AddHours( <Datum>, <Anzahl> )	Fügt eine bestimmte Anzahl von Stunden in einem Feld vom Typ "Date" oder "Date and time" hinzu.
AddMinutes( <Datum>, <Anzahl> )	Fügt eine bestimmte Anzahl von Minuten in einem Feld vom Typ "Date" oder "Date and time" hinzu.
AddSeconds( <Datum>, <Anzahl> )	Fügt eine bestimmte Anzahl von Sekunden in einem Feld vom Typ "Date" oder "Date and time" hinzu.
DaysDiff( <Datum1>, <Datum2> )	Anzahl von Tagen zwischen den Datumsangaben Datum1 und Datum2 (Zahl mit Dezimalstellen)
HoursDiff( <Datum1>, <Datum2> )	Anzahl von Stunden zwischen den Datumsangaben Datum1 und Datum2 (Zahl mit Dezimalstellen)
MinutesDiff( <Datum1>, <Datum2> )	Anzahl von Minuten zwischen den Datumsangaben Datum1 und Datum2 (Zahl mit Dezimalstellen)
SecondsDiff( <Datum1>, <Datum2> )	Anzahl von Sekunden zwischen den Datumsangaben Datum1 und Datum2 (Zahl mit Dezimalstellen)
DbToLocalDate( <Datum> )	Konvertiert ein Datum aus der für die Datenbank definierten Zeitzone in ein Datum aus der für den Client definierten Zeitzone.
LocalToDbDate( <Datum> )	Konvertiert ein Datum aus der für den Client definierten Zeitzone in ein Datum aus der für die Datenbank definierten Zeitzone.

**Tabelle 3.7. AQL - Funktionen vom Typ Datum**

Beschreibung	AssetCenter-Abfragesprachen
Alle Datensätze, die in der vorangegangenen Woche geändert wurden.	AddDays( dtLastModif,7 )>=Getdate()
Alle Support-Unterlagen, die seit weniger als einer Stunde geöffnet sind.	HoursDiff( Getdate(), dtNotif ) <= 1 oder AddHours( dtNotif, 1 ) >= Getdate()
Alle Support-Unterlagen, die seit weniger als 30 Minuten geöffnet sind	MinutesDiff( Getdate(), dtActualFixed ) <= 30 oder AddMinutes( dtActualFixed, 30 ) >= Getdate()

Mit der nachstehenden Abfrage rufen Sie unter Berücksichtigung der für den Client gültigen Zeitzone alle Support-Unterlagen auf, die an demselben Tag geöffnet und wieder geschlossen wurden:

```
SELECT Self FROM amWorkorder WHERE DayOfYear(DbToLocalDate(dtActualFix
Start)) = DayOfYear(DbToLocalDate(dtActualFixed))
```

Mit der nachstehenden Abfrage rufen Sie alle Support-Unterlagen auf, die am aktuellen Tag geöffnet wurden:

```
SELECT Self FROM amWorkorder WHERE DayOfYear(DbToLocalDate(dtActualFix
Start)) = DayOfYear(DbToLocalDate(GetDate()))
```

## AQL-Funktionen vom Typ Numerisch

**Tabelle 3.8. AQL - Funktionen vom Typ Numerisch**

Funktion	Beschreibung
Abs( <Zahl> )	Gibt den absoluten Wert einer Zahl zurück.
Ceil( <Zahl> )	Gibt die kleinste ganze Zahl zurück, die größer oder gleich einer Zahl ist.
Floor( <Zahl> )	Gibt die größte ganze Zahl zurück, die kleiner oder gleich einer Zahl ist.
Mod( <a>, <b> )	Gibt den Rest einer Division von "a" geteilt durch "b" zurück (a = qb + r, mit q als ganzer Zahl und 0 < r < q).
Round( <a>, <n> )	Rundet "a" auf "n" Dezimalstellen auf.
Trunc( <a>, <n> )	Schneidet "n" Dezimalstellen von "a" ab.



Anwendungsbeispiele:

Abs (2.516) = 2.

Ceil (2.516) = 3.

Floor (2.516) = 2.

Mod (6,4) = 2.

Round (31.16, 1) = 31.20.

Round (31.16, 0) = 31.00.

Round (31.16, -1) = 30.00.

Trunc (31.16, 1) = 31.1.

## AQL-Funktionen vom Typ Test

**Tabelle 3.9. AQL - Funktionen vom Typ Test**

Funktion	Beschreibung
IsNull( <a>, <b> )	Wenn "a" gleich "Null" ist, wird "a" durch "b" ersetzt. Die Typen von "a" und "b" müssen kompatibel sein.

## Abfragebeispiele

Jedes Beispiel stellt einen bestimmten Aspekt bei der Erstellung von Abfragen dar. Zur Erstellung persönlicher Abfragen können Sie mehrere der aufgeführten Abfragebeispiele als Vorlage verwenden.

Die Beispiele veranschaulichen die vollständige Abfragesyntax. Um die Abfragen wie dargestellt zu testen, verwenden Sie AssetCenter Export-. Wenn Sie die beschriebenen Abfragen in einem Abfragefilter verwenden möchten, müssen Sie die Syntax entsprechend ändern.

Aus einer Abfrage, die vollständig ausgeschrieben wie folgt aussieht:

```
SELECT self FROM amAsset WHERE Model.Brand.Name= 'Compaq'
```

wird in einem auf die Tabelle der Vermögen und Lose angewendeten Abfragefilter Folgendes (nur die Klausel WHERE ist explizit):

```
Model.Brand.Name= 'Compaq'
```

Sie können auch die in der mit AssetCenter gelieferten Demodatenbank enthaltenen Abfragen als Grundlage nehmen.



Anmerkung:

Zum Anzeigen der Transkription einer Abfrage in die SQL-Sprache des DBMS öffnen Sie einfach die Registerkarte **Vorschau** im Abfragedetail und klicken auf die Schaltfläche.

## Ein Feld der Haupttabelle mit einem Wert vergleichen

Beispiel: Alle Ausrüstungsgegenstände der Marke "Compaq".

```
SELECT Self FROM amPortfolio WHERE Model.Brand.Name = 'Compaq'
```

## Eine Verknüpfung der Haupttabelle mit einer anderen Verknüpfung vergleichen

Beispiel: Alle Ausrüstungselemente, die sich an demselben Standort wie der übergeordnete Gegenstand befinden.

```
SELECT Self FROM amPortfolio WHERE Location = Parent.Location
```

## Eine Verknüpfung der Haupttabelle mit einem Wert vergleichen

Beispiel: Alle Abteilungen und Personen, die zur Niederlassung Köln gehören.

```
SELECT Self FROM amEmplDept WHERE Parent.Name = Niederlassung Köln
```

## Ein Feld einer mit der Haupttabelle verknüpften Tabelle mit dem Feld einer anderen Tabelle vergleichen

Beispiele: Alle Ausrüstungselemente mit demselben Standortnamen wie ihre übergeordneten Gegenstände.

```
SELECT Self FROM amPortfolio WHERE Location.Name = Parent.Location.Name
```

## Hierarchische Tabellen

### Verwendung des Felds "FullName"

Beispiel: Alle Nebenstandorte des Standorts "Meteor-Gebäude".

```
SELECT Self FROM amLocation WHERE FullName LIKE '/Meteor-Gebäude/%'
```

### Verwendung der Felder "FullName" und "sLvl"

Bei den Abfragen in den hierarchischen Tabellen kommen häufig die Felder "FullName" und "sLvl" zur Anwendung.

Ein Beispiel: Alle Nebenstandorte des Standorts "Meteor-Gebäude", die sich auf einer hierarchischen Ebene unter 3 befinden.

In den Baumstrukturen entspricht die Wurzel der hierarchischen Stufe 0.

```
SELECT Self FROM amLocation WHERE (FullName LIKE '/ Meteor-Gebäude /%' ) AND (sLvl < 3)
```

Achten Sie auf das Zeichen "/" am Anfang und am Ende der vollständigen Namen.

## Abfragen mit zwei Bedingungen

Beispiel: Alle Personen mit der Funktion "Vertriebsingenieur" und dem Standort "Köln", ausgehend von der Tabelle der Personen.

```
SELECT Self FROM amEmplDEpt WHERE (Title = 'Vertriebsingenieur') AND (Location.Name = Standort Köln)
```

## Vergleich eines Felds mit Zahlen, Datumsangaben oder Text

Beispiel: Alle Interventionen, die zwischen dem 01.01.2003 und dem 31.12.2003 ausgeführt wurden.

```
SELECT self FROM amWorkOrder WHERE (dtActualFixStart >= #2003-01-01 00:00:00#) AND (dtActualFixStart <= #2003-12-31 00:00:00#)
```

## Merkmale abfragen

Beispiel: Alle Ausrüstungsgegenstände, für die das Merkmal mit dem SQL-Namen **fv\_Size** einen Wert von mehr als 150 cm aufweist.

```
SELECT Self FROM amPortfolio WHERE fv_Size >= 150.00
```

## Datensätze unter Verwendung eines Ausdrucks suchen

Beispiel: Alle Vermögensgegenstände, deren Einkaufspreis dem maximalen Einkaufspreis der in der Datenbank enthaltenen Vermögensgegenstände entspricht. Achten Sie darauf, dass innerhalb der Hauptabfrage eine Nebenabfrage zur Identifizierung des maximalen Preises verwendet wird.

```
SELECT Self FROM amAsset WHERE mPrice = (SELECT max(mPrice) FROM amAsset)
```

## Leere Felder mit Hilfe einer Abfrage suchen

Beispiel: Alle Personen ohne Telefonnummer. Fehlende Zeichenfolgen werden mit einem doppelten (') gekennzeichnet.

```
SELECT Self FROM amEmplDept WHERE Phone=''
```

## Fehlende Verknüpfungen mit Hilfe einer Abfrage suchen

### Fall einer 1-Verknüpfung

Beispiel: Alle Ausrüstungselemente, die noch keinem Benutzer zugeordnet wurden. Fehlende Verknüpfungen werden mit einer 0 gekennzeichnet.

```
SELECT Self FROM amPortfolio WHERE User = 0
```

### Fall einer n-Verknüpfung

Beispiel: Alle Modelle, denen keine Vermögensgegenstände zugeordnet wurden.

```
SELECT self FROM amModel WHERE NOT ( EXISTS (SELECT A1.lAstId FROM amAsset A1 WHERE A1.lModelId = amModel.lModelId) )
```

Diese Abfrage greift auf die Tabelle der Modelle zurück, prüft nacheinander sämtliche Modelle und zählt in der Tabelle der Vermögen und Lose die Gegenstände, die dem jeweiligen Modell entsprechen.

## Beispiel für den Test einer 1-Verknüpfung und einer n-Verknüpfung

Beispiel: Alle Modelle, die weder ein übergeordnetes noch ein untergeordnetes Modell aufweisen.

```
SELECT self FROM amModel WHERE (NOT ( EXISTS (SELECT A1.lModelId FROM amModel A1 WHERE A1.lParentId = lModelId))) AND (Parent = 0)
```

Die Abfrage führt nachstehende Prüfungen durch:

- Prüfung der 1-Verknüpfung ("Parent = 0"), um die Modelle ohne übergeordneten Vermögensgegenstand auszuwählen.
- Prüfung der n-Verknüpfung ("0 = (SELECT COUNT(a.lModelId) FROM amModel a WHERE a.lParentId = lModelId)"), um die Modelle ohne Untermodelle auszuwählen. Bei der Prüfung der n-Verknüpfung wird für jedes einzelne Modell die Kennung "lModelId" ausgewählt. Gleichzeitig werden alle Modelle gezählt, die ein übergeordnetes Modell mit der Kennung "lParentId" gleich "lModelId" aufweisen.

## Weiteres Beispiel

Alle Modelle ohne Untermodelle der Art "Computer".

```
SELECT self FROM amModel p WHERE NOT ( EXISTS (SELECT lModelId FROM amModel WHERE (FullName LIKE (p.FullName + '%/') AND (Nature.Name = 'Computer'))))
```

 Anmerkung:

Wenn Sie diese Abfrage mit AssetCenter Export ausführen, wird eine Fehlermeldung angezeigt. Sie können diese Meldung ignorieren. Die Abfrage funktioniert ordnungsgemäß.

## Abfrage mit Alias

Beispiel: Alle Personen, die an einer Schulung mit der Bezeichnung 'Peregrine' und an einer Schulung mit der Bezeichnung 'Datenbank' teilgenommen haben.

Ausgangstabelle: Tabelle der Abteilungen und Personen.

Die Abfrage lautet wie folgt:

```
SELECT Self FROM amEmplDept WHERE (Trainings_1.Title = 'Peregrine') AND  
D (Trainings_2.Title = 'Datenbank')
```

Alias-Namen der Form "Training\_1" und "Training\_2" ermöglichen die Verwendung von 2 Bedingungen auf 2 verschiedene Datensätze, die durch die Verknüpfung "Training" verbunden sind.

Sie haben folgende Syntax eingegeben:

```
SELECT Self FROM amEmplDept WHERE (Trainings.Title = 'Peregrine') AND  
(Trainings.Title = 'Datenbank')
```

Mit dieser Abfrage wählen Sie alle Personen, die sowohl an der einen als auch an der anderen Schulung teilgenommen haben.

Nehmen wir einmal an, Sie hätte die nachstehende Syntax eingegeben:

```
SELECT Self FROM amEmplDept WHERE (Trainings.Title = 'Peregrine') AND  
(Trainings.Title = 'Datenbank')
```

Mit dieser Abfrage wählen Sie alle Personen, die an einer der beiden Schulungen teilgenommen haben.

# 4 | Formulare

## KAPITEL

In diesem Kapitel erfahren Sie, wie Sie Formulare mit AssetCenter erstellen. Der Zugriff auf die Liste der Formulare erfolgt über das Menü **Extras/Reporting/ Formulare**.

### Definition eines Formulars

Ein Formular ist das Modell eines Dokuments, mit dem Sie Daten ausdrucken. Im Gegensatz zu Crystal Reports-Berichten können Formulare direkt in AssetCenter bearbeitet werden.

### Vorkonfigurierte Formulare in der Datenbank mit den Lebenddaten installieren

Im Lieferumfang von AssetCenter sind vorkonfigurierte Formulare enthalten, die in der Demodatenbank bereits installiert sind. Sie müssen Sie also nur noch

bei der Erstellung bzw. Aktualisierung der Datenbank mit den Lebendaten importieren.

## Formulare bei der Erstellung einer Datenbank importieren

Zum Importieren von Formularen bei der Erstellung einer Datenbank gehen Sie wie folgt vor:

- 1 Starten Sie AssetCenter Database Administrator.
- 2 Wählen Sie **Datei/ Öffnen**.
- 3 Wählen Sie die Option **Datenbank beschreibungsdatei öffnen - neue Datenbank anlegen**.
- 4 Wählen Sie die Datei **gbbase.xml**, die im Unterordner **config** des Installationsordners von AssetCenter gespeichert ist.
- 5 Wählen Sie **Aktionen/ Datenbank anlegen**.
- 6 Geben Sie alle Daten ein, die zum Anlegen einer Datenbank erforderlich sind. Weitere Informationen finden Sie im Handbuch **Verwaltung**, Kapitel **Anlegen von AssetCenter-Datenbanken**.
- 7 Wählen Sie die Option **Zusatzdaten importieren**.
- 8 Wählen Sie im Rahmen **Zusatzdaten importieren** die Option **Formulare**.
- 9 Klicken Sie auf **Erstellen**, um das Anlegen der Datenbank zu bestätigen.

## Formulare in eine existierende Datenbank importieren

Zum Importieren von Formularen in eine existierende Datenbank gehen Sie wie folgt vor:

- 1 Starten Sie AssetCenter Database Administrator.
- 2 Wählen Sie **Datei/ Öffnen**.
- 3 Wählen Sie die Option **Datenbank beschreibungsdatei öffnen - neue Datenbank anlegen**.
- 4 Wählen Sie die Datei **gbbase.xml**, die im Unterordner **config** des Installationsordners von AssetCenter gespeichert ist.
- 5 Wählen Sie **Aktionen/ Datenbank anlegen**.
- 6 Wählen Sie die Datenbank im Feld **Datenbank**.
- 7 Deaktivieren Sie die Optionen **Datenbank anlegen** und **Systemdaten erstellen**.
- 8 Wählen Sie die Option **Zusatzdaten importieren**.



- 9 Wählen Sie im Rahmen **Zusatzdaten importieren** die Option **Formulare**.
- 10 Klicken Sie auf **Erstellen**, um den Import der Formulare zu bestätigen.

## Formulare erstellen

Der Zugriff auf die Liste der Formulare erfolgt über das Menü **Extras/Reporting/ Formulare**.

### Allgemeine Beschreibung

- 1 Geben Sie den Namen des Formulars ein.
- 2 Wählen Sie den Formulartyp (Liste oder Detail).  
Die beiden Formulartypen können sowohl vordefinierte Texte als auch Bilder enthalten.  
Unterschiede zwischen diesen beiden Formulartypen:
  - Liste: Drucken einer Datensatzliste, wie sie im aktiven Fenster erscheint (je nach den Spalten in der Liste und den angewendeten Filtern).
  - Detail: Drucken von Feldern in einem Detailfenster, beispielsweise dem Detailfenster eines Vermögensgegenstands, und der damit verknüpften Datensatzlisten, wie z. B. die Komponenten dieses Gegenstands.
- 3 Wählen Sie die Haupttabelle des Formulars.

---

 **WARNUNG:**

Das Feld **Tabelle** (SQL-Name: TableName) bestimmt, welche Formulare von AssetCenter beim Drucken für eine bestimmte Liste angezeigt werden.

---






## Formulare und Objekte bearbeiten

Beim Bearbeiten eines Formulars werden Objekte auf einer Seite positioniert und definiert.

Gehen Sie zum Einfügen eines neuen Objekts auf eine Seite wie folgt vor:

- 1 Wählen Sie die Registerkarte **Formular**.

- Klicken Sie auf das Symbol des Objekts, das sich am linken Rand der Seite befindet.

Symbol	Funktion
	<b>Auswahl eines Formularobjekts, beispielsweise um es zu ändern.</b>
	Hinzufügen eines festen Textes und unabhängiger Variablen aus den gedruckten Datensätzen, z. B. das Tagesdatum.
	Hinzufügen eines Bildes.
	Einfügen einer Formel, die die Werte für Felder und Zeichenfolgen des Standardtextes enthält.
	Einfügen einer Datensatzliste. Programm zum Positionieren der Liste auf der Seite. Ermöglicht das Definieren der verknüpften Tabelle mit den Datensätzen und den zu druckenden Listen in Formulardetails.

- Setzen Sie den Cursor auf die gewünschten Einfügestelle.
- Drücken Sie die linke Maustaste.
- Ziehen Sie mit der Maus einen Rahmen auf: Er stellt die Begrenzungslinie des Objekts dar.
- Doppelklicken Sie auf den Rahmen. Es erscheint ein Fenster zur Definition der Objekteigenschaften.
- Definieren Sie die Objekteigenschaften.
- Klicken Sie auf **Ändern**.

Formulare können die folgenden Objekte aufnehmen.

## Feste Texte

Texte, deren Inhalt von den gedruckten Datensätzen unabhängig ist. Diese Texte können Zeichen und Variablen enthalten:

\$D: Datum des Drucks

\$U: Login des AssetCenter-Benutzers, der den Druck anfordert

\$C: Seitenzahl

\$N: Gesamtanzahl der gedruckten Seiten

**WARNUNG:**

Setzen Sie den Text nicht in Anführungszeichen.

Beispiel:

Dokument gedruckt am \$D von \$U

## Formeln

Formeln können nur in Formularen vom Typ "Detail" verwendet werden.

In den Formeln stehen nebeneinander:

- Die Werte der Felder der AssetCenter-Datenbank
- Die in Anführungszeichen gesetzten festen Texte

Beispiel:

"Vermögen:" CodInt " / " Marke

Berechnungen können mit diesen Formeln jedoch nicht durchgeführt werden.

## Listen

- Formulare vom Typ "Liste" können nur eine einzige Liste aufweisen. Bei Auswahl des Menüs **Datei/ Drucken** wird die Liste dann durch die Liste des zum Zeitpunkt des Druckens aktiven Bildschirms ersetzt.
- Formulare vom Typ "Detail" schränken die Anzahl der verwendeten Listen in keiner Weise ein. Die Listen stellen alle Datensätze dar, die eine Beziehung zum aktuellen Datensatz aufweisen, z. B. alle Teile eines Vermögensgegenstands.

## Bilder

Sie haben auch die Möglichkeit, Bilder einzufügen (Logos...).

---

 **Anmerkung:**

Beim Bearbeiten von Formularen vom Typ "Liste" haben Sie nicht die Möglichkeit, die Felder zu wählen, die angezeigt werden sollen. AssetCenter druckt die Felder, die in Form von Spalten in den Listen erscheinen.

Bei Formularen des Typs Detail dagegen können Sie die Felder, die Sie drucken möchten, einzeln auswählen.

---

## Eigenschaften von Formularobjekten

### Position und Größe

Durch Ziehen und Verschieben der Greifpunkte bzw. des Objekts können Sie die Größe bzw. die Position eines Objekts ändern.

---

 **Anmerkung:**

Es ist möglich, mehrere Objekte gleichzeitig zu verschieben oder in der Größe zu ändern. Ziehen Sie dazu (durch Ziehen und Verschieben der Maus) ein Auswahlfeld auf. In diesem Fall sind alle in diesem Feld enthaltenen Objekte ausgewählt. Eine Alternative besteht im Markieren der Objekte bei gedrückter Strg-Taste.

---

### Eigenschaften

Durch Doppelklicken auf ein Objekt oder über das Menü **Interne Formulare/ Eigenschaften...** zeigen Sie eine Palette mit den Eigenschaften des markierten Objekts an.

Die Palette enthält zwei Spalten: In der ersten Spalte steht der Name der Eigenschaft, in der zweiten können Sie den Wert der Eigenschaft festlegen.

Um eine Eigenschaft zu ändern, doppelklicken Sie auf die zweite Spalte.

Einfache Eigenschaften (Text, Formel, Liste, Hintergrundfarbe, Textfarbe, Textausrichtung, Objektausrichtung) können unmittelbar bearbeitet werden.

Die komplexeren Eigenschaften (Rahmen, Text- oder Formelschriftart, Listeninhalt, Bild) werden über zusätzlich angezeigte Bildschirme geändert.

## Text

Geben Sie hier den gewünschten Text ein.

## Formel

Zum Erstellen der Formel können Sie eine Baumstruktur einblenden, die alle verfügbaren Felder für das gewählte Formular enthält. Durch Klicken auf einen der Knotenpunkte der Baumstruktur wird die aktuelle Wahl durch das gewählte Feld ersetzt, das in Anführungszeichen gesetzt wird. Zwischen den Feldern können Sie Text einfügen, der ebenfalls in Anführungszeichen gesetzt werden muss.

## Hintergrundfarbe, Textfarbe

Sie haben die Wahl zwischen 16 Farben.

## Textausrichtung


Ausrichtung des Texts im Rahmen. Die einzelnen Ausrichtungsmöglichkeiten (zentriert, linksbündig, rechtsbündig) werden in einer Dropdown-Liste ausgewählt.

## Seitenausrichtung


Horizontale Ausrichtung des Rahmens auf der Seite. Die gewünschte Ausrichtung wird in einer Dropdown-Liste gewählt:

- **Linksbündig**
- **Rechtsbündig**
- **Zentriert**
- **Relative Ausrichtung:** Das Objekt behält die Ausrichtung bei, die Sie auf dem Bildschirm zur Erstellung eines Formulars festgelegt haben.

## Schriftart

Klicken Sie in der bearbeiteten Zelle auf die Schaltfläche , um die Schriftart und den Schriftgrad für das Objekt zu wählen.

## Rahmen

Klicken Sie im Bearbeitungsbereich auf die Schaltfläche , und definieren Sie auf dem nun angezeigten Bildschirm die Begrenzungslinie für das Objekt.

Mit der Option **3D-Effekt** wird der Rahmen reliefartig dargestellt.

Für alle Rahmen ohne 3D-Effekt können Sie die Linienfarbe und -stärke festlegen. Geben Sie außerdem an, welche Ränder gezeichnet dargestellt werden sollen.

## Bild

Gehen Sie zum Einfügen eines Bildes wie folgt vor:

- 1 Klicken Sie auf die Schaltfläche im Bearbeitungsfeld.
- 2 Wählen Sie die gewünschte Grafikdatei im nun erscheinenden Dialogfenster.

## Verknüpfung - Liste

Um die im Formular anzuzeigende Liste zu wählen, blenden Sie eine Baumstruktur mit den verfügbaren Listen ein und klicken auf die gewünschte Liste (z. B. die Liste der von einer Person benutzten Vermögensgegenstände).

## Listeninhalt

---



**Anmerkung:**

Die Konfiguration des Listeninhalts ist nur möglich, wenn das Detailfenster zum Bearbeiten des Formulars angezeigt wird.

---

Klicken Sie auf die Schaltfläche im Bearbeitungsfeld, um den Inhalt einer Liste zusammenzustellen. Es erscheint ein Konfigurationsbildschirm mit sämtlichen Spalten der Liste.

Definieren Sie auf dem nun angezeigten Bildschirm folgendes:

- Die Überschriften der einzelnen Spalten
- Die Formel zum Zusammenstellen des Spalteninhalts
- Die Spaltenbreite (prozentual im Verhältnis zur Seite)
- Die Schriftart der Überschrift und des Inhalts der einzelnen Spalten
- Die Ausrichtung der einzelnen Überschriften und des Spalteninhalts
- Die Art der horizontalen und vertikalen Begrenzungslinien

Drücken Sie die Taste "Entf", um eine Spalte aus der Liste zu löschen.

Wenn Sie eine Spalte in die Liste einfügen möchten, bearbeiten Sie die letzte Listenzeile.

Alle Zellen werden auf die gleiche Art und Weise wie die Eigenschaftenpalette bearbeitet.

## Eingaberaster

Das Eingaberaster besteht aus horizontalen und vertikalen Linien.

Führen Sie nachstehende Vorgänge über die Option **Interne Formulare/Raster** im spezifischen Menü:

- Ein- und Ausblenden des Rasters
- Definieren der Abstände zwischen den horizontalen und vertikalen Linien

Das Raster wird in Form von Kreuzungspunkten der Linien angezeigt. Der Abstand zwischen den einzelnen Linien legt folglich fest, mit welcher Genauigkeit die Objekte auf einer Seite verschoben werden können.

## Formularseiten einrichten

Über die Option **Interne Formulare/ Seite einrichten** im spezifischen Menü definieren Sie folgende Elemente:

- Druckformat
- Ausrichtung in **Hochformat** oder **Querformat**
- Dokumentränder
- Eventuell vorhandene Kopf- und Fußzeilen



**Anmerkung:**

Die Dokumentränder sowie die Kopf- und Fußzeilen können direkt in der Eingabezone geändert werden. Ziehen und verschieben Sie dazu einfach mit der Maus die als gepunktete Linien dargestellten Randmarkierungen.

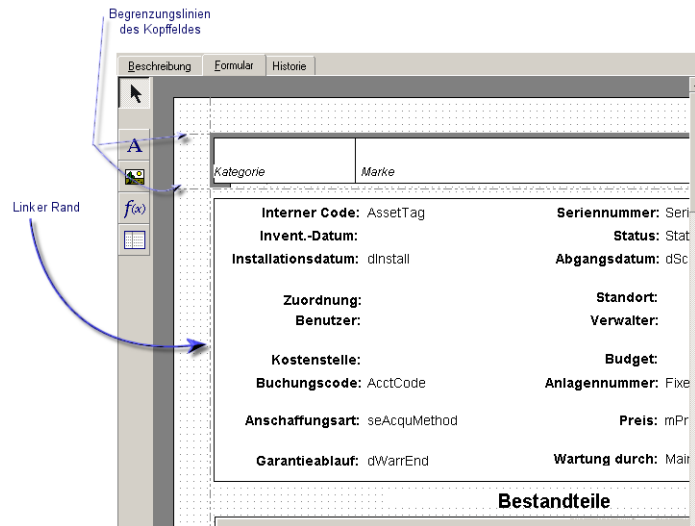
---

Gehen Sie zum Einfügen von Text in Kopf- und Fußzeilen wie folgt vor:

- 1 Wählen Sie das Menü **Interne Formulare/ Seite einrichten**.
- 2 Markieren Sie die Kontrollkästchen **Kopfzeile** und **Fußzeile**.
- 3 Klicken Sie auf **OK**.
- 4 Setzen Sie den Mauszeiger nun in den Kopf- bzw. Fußzeilen des Formular-Bearbeitungsfensters. Die Kopf- und Fußzeilen sind durch gepunktete Linien begrenzt.

- 5 Fügen Sie die gewünschten Objekte ein, oder verschieben Sie die Objekte im Hauptfeld.
- 6 Klicken Sie auf **Ändern**, um die Änderungen zu bestätigen.

**Abbildung 4.1. Formulare - Kopffeld**



**Anmerkung:**

Es ist nicht möglich, ein Objekt aus dem Hauptfeld in die Kopf- oder Fußzeilen der Seite zu verschieben oder umgekehrt.

## Einfache Berichte mühelos erstellen

Zum Generieren regelmäßig verwendeter Berichte empfehlen wir Ihnen die nachstehende Vorgehensweise:

- 1 Erstellen Sie eine "Ansicht" mit den Parametern, die Ihren Anforderungen entsprechen.
- 2 Ordnen Sie diese Ansicht einem Druckformular zu.

Definieren Sie folgende Elemente für die Ansicht:



- Das Sortierkriterium
- Den anzuwendenden Filter sowie die Filterwerte
- Die Liste der sichtbaren Spalten

Mit Hilfe des Formulars können Sie die Darstellung der Informationen auf dem Bericht festlegen.

Gehen Sie beim Drucken des Berichts wie folgt vor:

- 1 Zeigen Sie die zuvor erstellte Ansicht über das **Extras/ Ansichten** an.
- 2 Drucken Sie den Bericht unter Verwendung der angezeigten Ansicht (Menü **Datei/ Drucken**): Wählen Sie den für den Druck geeigneten "Typ" und das entsprechende "Formular".

## Formulare eines bestimmtes Moduls identifizieren

Zum Identifizieren von Formularen eines bestimmten Moduls gehen Sie wie folgt vor:

- 1 Starten Sie AssetCenter.
- 2 Zeigen Sie die Liste der Formulare an (**Extras/ Reporting/ Formulare**).
- 3 Klicken Sie mit der rechten Maustaste auf die Liste.
- 4 Wählen Sie **Liste konfigurieren**.
- 5 Fügen Sie die Verknüpfung **Bereich** (Domain) in den Spalten der Liste hinzu.
- 6 Klicken Sie auf **OK**.
- 7 Sortieren Sie die Liste anhand der Spalte **Bereich**.
- 8 Die Formulare des Moduls werden anhand des Bereichsnamens identifiziert.  
Beispiel: **/Beschaffung/Formulare/**.

## Formulare und Aktionsschaltflächen zuordnen

Es bestehen mehrere Möglichkeiten, einem Formular eine Aktionsschaltfläche zuzuordnen.

Weitere Informationen dazu finden Sie an folgenden Stellen:

- Handbuch **Einführung in AssetCenter**, Kapitel **Anpassung eines Clients**, Abschnitt **Anpassung der Schaltflächen**

- Handbuch **Verwaltung**, Kapitel **Anpassung der Datenbank**, Abschnitt **Vorhandene Objekte anpassen/ Objekte anpassen / Details anpassen / Schaltflächen erstellen**
- Handbuch **Verwaltung**, Kapitel **Anpassung der Datenbank**, Abschnitt **Neue Objekte erstellen/ Aktionsschaltflächen erstellen**

# 5 | Crystal-Berichte

---

## KAPITEL

In diesem Kapitel wird das Bearbeiten von Berichten mit AssetCenter erläutert. Der Zugriff auf die Liste der Berichte erfolgt über das Menü **Extras/ Reporting/ Berichte**.

---

 Anmerkung:

Für die UNIX-Version der Anwendung AssetCenter stehen die Crystal-Berichte nicht zur Verfügung.

---

## Funktionsprinzip und Installation des Berichtsgenerators

### Allgemeines Funktionsprinzip

AssetCenter setzt die Software Crystal Reports zum Drucken von Berichten ein. Berichte weisen die Dateierweiterung **.rpt** auf.



Anmerkung:

Die kommentierte Liste der verfügbaren Berichte befindet sich in der Datei **reports.txt**, die im Unterordner **datakit\standard\reports** des Installationsordners von AssetCenter gespeichert ist.

Die Crystal-Berichte sind im Ordner **datakit\standard\reports\rpt** gespeichert. Die Struktur der Datei **reports.txt** darf nicht verändert werden, da das Skript für den Berichtimport auf diese Datei zugreift. Wenn Sie jedoch nur eine Auswahl von Berichten in die Datenbank importieren möchten, können Sie die **kompletten** Zeilen in dieser Datei vor dem Ausführen des Skripts löschen. Darüber hinaus können Sie auch ihre eigenen Berichte hinzufügen.

## Installation von Crystal Reports zum Drucken vorgegebener Berichte nicht erforderlich

AssetCenter umfasst eine Grundversion von Crystal Reports. Sie wird installiert, wenn Sie bei der Installation die entsprechende Option ankreuzen.

Diese Grundversion reicht aus, um die vorgegebenen Berichte mit den aktuellen Daten aus der AssetCenter-Datenbank in der Vorschau einzusehen und zu drucken.

## Installation von Crystal Reports zum Ändern vorgegebener Berichte oder Erstellen neuer Berichte erforderlich

Die Erstellung von Crystal Reports-Berichten ist mit AssetCenter allein natürlich nicht möglich.

Sie müssen zunächst eine Vollversion von Crystal Reports installieren. Die damit einhergehenden Einschränkungen sind in der folgenden Tabelle beschrieben:

<b>Unterstützte Crystal Reports-Versionen</b>	Version 7.x und 8.0
	Version in japanischer Sprache: Version 7.0

**Unterstützte Crystal Reports-Sprachen**

- Französisch
- Englisch
- Deutsch
- Spanisch
- Italienisch
- Japanisch
- Polnisch

Sie können für Crystal Reports und AssetCenter unterschiedliche Sprachen verwenden.

**Von Crystal Reports unterstützte ODBC-Treiber**

ODBC-Treiber von AssetCenter.

## Crystal Reports-Vollversion oder Laufzeitversion installieren

Das AssetCenter-Installationsprogramm bietet Ihnen die Möglichkeit, eine Laufzeitversion von Crystal Reports zu installieren, die zum Drucken der bereits vorhandenen Berichte ausreicht. Sie müssen dazu lediglich die entsprechende Option bei der Installation von AssetCenter markieren.

Wenn Sie die Vollversion von Crystal Reports installieren möchten oder bereits installiert haben, ist das Markieren dieser Option in AssetCenter selbstverständlich überflüssig.

## Vorkonfigurierte Crystal Reports-Berichte in der Datenbank installieren

Im Lieferumfang von AssetCenter sind vorkonfigurierte Berichte enthalten, die in der Demodatenbank bereits installiert sind. Sie brauchen sie also nur noch in der Datenbank mit den Lebendaten zu importieren.

**Anmerkung:**

In den nachstehenden Versionen von AssetCenter sind die vorkonfigurierten Berichte jedoch nicht vorhanden:

- Japanische Version
- Polnische Version

Zum Importieren einzelner Berichte gehen Sie wie folgt vor:

- 1 Starten Sie AssetCenter.

- 2 Öffnen Sie die Datenbank.
- 3 Verwenden Sie das Menü **Extras/ Reporting/ Berichte**.
- 4 Erstellen Sie einen neuen Bericht.
- 5 Wählen Sie die Registerkarte **Datei** im Berichtdetail.
- 6 Klicken Sie auf **Importieren**.
- 7 Wählen Sie im Unterordner `\datakit\standard\reports\rpt` des Installationsordners von AssetCenter die Datei mit der Erweiterung **.rpt**, die Ihren Anforderungen entspricht.

## Berichte beim Anlegen einer neuen Datenbank importieren

Zum Importieren von Berichten beim Anlegen einer Datenbank gehen Sie wie folgt vor:

- 1 Starten Sie AssetCenter Database Administrator.
- 2 Wählen Sie **Datei/ Öffnen**.
- 3 Wählen Sie die Option **Datenbank beschreibungsdatei öffnen - neue Datenbank anlegen**.
- 4 Wählen Sie die Datei **gbase.xml**, die im Unterordner **config** des Installationsordners von AssetCenter gespeichert ist.
- 5 Wählen Sie **Aktionen/ Datenbank anlegen**.
- 6 Geben Sie alle Daten ein, die zum Anlegen einer Datenbank erforderlich sind. Weitere Informationen finden Sie im Handbuch **Verwaltung**, Kapitel **Anlegen von AssetCenter-Datenbanken**.
- 7 Wählen Sie die Option **Zusatzdaten importieren**.
- 8 Wählen Sie im Rahmen **Zusatzdaten importieren** die Option **Crystal Reports-Berichte**.
- 9 Klicken Sie auf **Erstellen**, um das Anlegen der Datenbank zu bestätigen.

## Berichte in eine existierende Datenbank importieren

Zum Importieren von Berichten in eine existierende Datenbank gehen Sie wie folgt vor:

- 1 Starten Sie AssetCenter Database Administrator.
- 2 Wählen Sie **Datei/ Öffnen**.
- 3 Wählen Sie die Option **Datenbank beschreibungsdatei öffnen - neue Datenbank anlegen**.
- 4 Wählen Sie die Datei **gbase.xml**, die im Unterordner **config** des Installationsordners von AssetCenter gespeichert ist.

- 5 Wählen Sie **Aktionen/ Datenbank anlegen**.
- 6 Wählen Sie die Datenbank im Feld **Datenbank**.
- 7 Deaktivieren Sie die Optionen **Datenbank anlegen** und **Systemdaten erstellen**.
- 8 Wählen Sie die Option **Zusatzdaten importieren**.
- 9 Wählen Sie im Rahmen **Zusatzdaten importieren** die Option **Crystal Reports-Bericht**.
- 10 Klicken Sie auf **Erstellen**, um den Import der Berichte zu bestätigen.

## Detail eines Berichts

Der Zugriff auf die Liste der Berichte erfolgt über das Menü **Extras/ Reporting/ Berichte**.

## Berichte verwalten

 **Anmerkung:**

Das nachstehend genannte Feld **Datei** (SQL-Name: FileName) kann nicht direkt bearbeitet werden. Es enthält den Namen der Berichtdatei (mit Dateierweiterung und relativem Zugriffspfad auf den entsprechenden Ablageordner), die über die Schaltfläche **Importieren** importiert wurde.

Folgende Schaltflächen können bei der Arbeit mit Berichten eingesetzt werden:

- **Importieren** : Schaltfläche zum Importieren eines externen Berichts (beim ersten Mal zum Erstellen, anschließend zum Bearbeiten des Berichts). Externe Berichte weisen die Dateierweiterung **.rpt** auf. Beim Importieren des Berichts wird das Feld **Datei** (SQL-Name: FileName) im Detail des AssetCenter-Berichts aktualisiert.
- **Exportieren**: Schaltfläche zum Erstellen einer **.RPT-Datei** über einen in der AssetCenter-Datenbank enthaltenen Bericht. Standardmäßig zeigt das nun erscheinende Fenster als Dateinamen den im Feld **Datei** angegebenen Namen an. Dieser Name kann jedoch geändert werden. Bei diesem Vorgang haben Sie die Möglichkeit, den Bericht mit dem Editor für externe Berichte zu bearbeiten.

- **Seitenansicht:** Diese Schaltfläche steht ebenfalls über das Menü **Datei/ Drucken** zur Verfügung und dient zum Anzeigen des Berichts vor dem Druck.
  - **Drucken :** Diese Schaltfläche steht über das Menü **Datei/ Drucken** zur Verfügung und dient zum Drucken des Berichts
- 

 **Anmerkung:**

Wenn Sie auf die Schaltfläche **Seitenansicht** oder **Drucken** klicken, erstellt AssetCenter von dem in der Datenbank enthaltenen Bericht eine temporäre Datei. Diese Datei wird von der Crystal Reports-Druck-Engine bearbeitet und anschließend gelöscht. Die angezeigten bzw. gedruckten Daten sind die aktuellen Daten der offenen Datenbank.

---

## Crystal Reports-Berichte ändern

Das Ändern eines in der AssetCenter-Datenbank enthaltenen Berichts ist nur über Crystal Reports möglich.

Gehen Sie zum Ändern eines solchen Berichts wie folgt vor:

- 1 Zeigen Sie das Berichtdetail über das Menü **Extras/ Reporting/ Berichte** an.
- 2 Klicken Sie auf die Schaltfläche **Exportieren**, um eine Datei mit der Erweiterung **.rpt** zu erstellen.
- 3 Ändern Sie den **.rpt**-Bericht unter Crystal Reports und speichern Sie die neue Version.
- 4 Zeigen Sie das Berichtdetail erneut über das Menü **Extras/ Reporting/ Berichte** an.
- 5 Importieren Sie die **.rpt**-Datei, um die Datei zu aktualisieren und die gespeicherte Version zu ändern.

## Crystal Reports-Statistiken

Verwenden Sie das Menü **Extras/ Reporting/ Crystal Reports-Statistiken**, um die automatisch aktualisierten Crystal Reports-Berichte anzuzeigen.




Sie können dieselben Berichte anzeigen, auf die Sie über das Menü **Extras/ Reporting/ Berichte** zugreifen.

## Art


Geben Sie die Art des Berichts an, der angezeigt werden soll. Im Feld rechts neben der Art des Berichts können Sie einen Bericht wählen. Die angezeigten Berichte ergeben sich aus der "Art" für die Sie sich entschieden haben.

## Schaltfläche Automatisch auffrischen

Die Schaltfläche wird durch das Symbol  dargestellt.

- Drücken Sie die linke Maustaste, um den Bericht unmittelbar zu aktualisieren.
- Drücken Sie die rechte Maustaste zum Festlegen der Häufigkeit, mit der die Berichte aktualisiert werden sollen.

## Schaltfläche Zoom

Die Schaltfläche wird durch das Symbol  dargestellt.

Ändern Sie den Zoom-Faktor (3 Stufen).

## Detailberichte erstellen

Unter einem "Detailbericht" ist ein Bericht zu verstehen, der die Detailinformationen für die in der Liste ausgewählten Datensätze druckt.

## Anwendungsbeispiel

- 1 Zeigen Sie die Liste der Vermögensgegenstände an.
- 2 Wählen Sie einen Vermögensgegenstand.
- 3 Wählen Sie das Menü **Datei/ Drucken**.
- 4 Ordnen Sie dem Feld **Typ** den Wert "Detaillierter Bericht (Crystal Reports)" zu.

- 5 Wählen Sie den Bericht.
- 6 Starten Sie den Druck.

Daraufhin wird ein Detailbericht für jeden ausgewählten Datensatz gedruckt.

## Berichtparameter unter Crystal Reports definieren

Zum Erstellen eines Detailberichts müssen Sie den nachstehenden Anweisungen folgen (Beispiel mit Crystal Reports Professionnel 5.0, englischsprachige Version):

- 1 Erstellen Sie ein Formelfeld über das Menü **Insert/ Formula Field**. Der Name muss die folgende Syntax aufweisen:

```
<SQL-Name der Tabelle, für die ein kontextbezogener Bericht erstellt werden soll>Id
```

 **Anmerkung:**

In den Tabellen müssen Sie unbedingt die Groß- und Kleinschreibung der SQL-Namen berücksichtigen.

Die Formel zum Erstellen eines kontextbezogenen Berichts für die Tabelle der Vermögen und Lose lautet beispielsweise:

```
amAssetId
```

 **Anmerkung:**

Verwechseln Sie nicht die Syntax des Formelnamens mit dem SQL-Namen des Felds, das als Primärschlüssel dient. Der Primärschlüssel der Tabelle der Vermögen und Lose lautet beispielsweise "lAstId", und nicht "amAssetId".

Die Formel "CurrentUserId" (beachten Sie die Groß- und Kleinschreibung) identifiziert den Benutzer, der den Bericht druckt. Beim Druckvorgang entspricht diese Formel der Feld-Id, und damit dem Wert des Felds mit dem SQL-Namen: "lEmplDeptId" für das aktuelle Login des Benutzers, der mit der AssetCenter-Datenbank verbunden ist.

Wenn Sie das Ergebnis eines Berichts für einen bestimmten Datensatz der Kontexttabelle anzeigen möchten, müssen Sie das Formelfeld bearbeiten. Ordnen Sie diesem Feld dann den Wert des Primärschlüssels der Tabelle für einen in der AssetCenter-Datenbank vorhandenen Datensatz zu.

Beispiel:

512

 **Anmerkung:**

Die Bearbeitung des Formelfelds erfolgt in dem Fenster, das automatisch nach dem Bestätigen des Formelfeldnamens erscheint. Wenn das Formelfeld bereits existiert, klicken Sie auf **Edit**, um es zu bearbeiten.

- 2 Bearbeiten Sie die Auswahlformel über das Menü **Report/ Edit Selection Formula/ Record**. Die Formel muss die nachstehende Struktur aufweisen:

```
{<SQL-Name der Kontexttabelle>.<SQL-Name des Felds, das als Primärschlüssel dient>} = @<Name des Formelfelds>}
```

Der Groß- und Kleinschreibung der SQL-Namen von Tabellen und Feldern wird keine Rechnung getragen. Beispiel:

```
{amAsset.lAstId} = {@amAssetId}
```

Wenn Sie die oben genannten Anweisungen befolgen, identifiziert AssetCenter den Bericht beim Import in die Datenbank automatisch als kontextbezogen. Dies wird beim Ausführen folgender Vorgänge deutlich:

- 1 Greifen Sie über das Menü **Extras/ Reporting/ Berichte** auf die Liste der Berichte zu.
- 2 Erstellen Sie einen neuen Bericht.
- 3 Klicken Sie auf die Schaltfläche **Importieren**, um die Crystal Reports-Datei **.rpt** zu importieren.
- 4 Sobald Sie die Datei hinzugefügt haben, sehen Sie im Feld **Tabelle** (SQL-Name: TableName) den SQL-Namen der Kontexttabelle. Ist das nicht der Fall, prüfen Sie das Formelfeld und die Auswahlformel im Crystal Reports-Bericht.

# Berichte drucken

## Detailberichte

---

 **Anmerkung:**

Eine Definition und Anweisungen zur Erstellung von Detailberichten mit Crystal Reports finden Sie in: ► [Detailberichte erstellen](#) [Seite 89].

---

So drucken Sie einen Detailbericht in AssetCenter:

- 1 Zeigen Sie die Datensatzliste an, deren Detail gedruckt werden soll (beispielsweise **Ausrüstung/ Vermögen und Lose**).
- 2 Markieren Sie alle zu druckenden Datensätze.
- 3 Wählen Sie das Menü **Datei/ Drucken**.
- 4 Ordnen Sie dem Feld **Typ** den Wert **Detaillierter Bericht (Crystal Reports)** zu.
- 5 Wählen Sie den Bericht.
- 6 Klicken Sie auf **Drucken**.

Daraufhin wird ein Detailbericht für jeden ausgewählten Datensatz gedruckt.

## Kontextunabhängige Berichte

---

 **Anmerkung:**

Listen- und Graphenberichte werden als **kontextunabhängige Berichte** bezeichnet.

Sie beziehen sich im Gegensatz zu **Detailberichten** nicht auf eine Tabelle als Kontext.

---

So drucken Sie kontextunabhängige Berichte in AssetCenter:

- 1 Wählen Sie das Menü **Datei/ Drucken**.
- 2 Ordnen Sie dem Feld **Typ** den Wert **Kontextunabhängig. Bericht (Crystal)** zu.
- 3 Wählen Sie den Bericht.
- 4 Klicken Sie auf **Drucken**.

Daraufhin wird ein Listen- oder Graphenbericht gedruckt.

## Crystal-Berichte eines bestimmten Moduls identifizieren

Zum Identifizieren der Crystal-Berichte eines bestimmten Moduls gehen Sie wie folgt vor:

- 1 Starten Sie AssetCenter.
- 2 Zeigen Sie die Liste mit den Berichten an (**Extras/ Reporting/ Berichte**).
- 3 Klicken Sie mit der rechten Maustaste auf die Liste.
- 4 Wählen Sie im Kontextmenü den Befehl **Liste konfigurieren**.
- 5 Fügen Sie die Verknüpfung **Bereich [Domain]** in den Spalten der Liste hinzu.
- 6 Klicken Sie auf **OK**.
- 7 Sortieren Sie die Liste anhand der Spalte **Bereich**.
- 8 Die Berichte des Moduls werden anhand des Bereichsnamens identifiziert.  
Beispiel: `/Ausrüstungsverwaltung/EDV/Berichte/`.

## Berichte und Aktionsschaltflächen zuordnen

Es bestehen mehrere Möglichkeiten, einem Bericht eine Aktionsschaltfläche zuzuordnen.

Weitere Informationen dazu finden Sie an folgenden Stellen:

- Handbuch **Einführung in AssetCenter**, Kapitel **Anpassung eines Clients**, Abschnitt **Anpassung der Schaltflächen**
- Handbuch **Verwaltung**, Kapitel **Anpassung der Datenbank**, Abschnitt **Vorhandene Objekte anpassen/ Objekte anpassen / Details anpassen / Schaltflächen erstellen**
- Handbuch **Verwaltung**, Kapitel **Anpassung der Datenbank**, Abschnitt **Neue Objekte erstellen/ Aktionsschaltflächen erstellen**



# 6 | Aktionen

## KAPITEL

In diesem Kapitel wird das Definieren von Aktionen mit AssetCenter erläutert. Die Definition von Aktionen erfolgt über das Menü **Extras/ Aktionen/ Bearbeiten**.

Die Ausführung von Aktionen erfolgt über das Menü Extras/ Aktionen oder die Kontextliste "Aktionen" in der Symbolleiste.

## Definition einer Aktion

Eine Aktion ist ein Vorgang, bei dem ein ausführbares Programm direkt über AssetCenter ausgeführt wird.

Aktionen unterscheiden sich nach ihrem Typ:

- Programm
- DDE
- Nachricht
- Skript: Änderung eines Objekts der AssetCenter-Datenbank
- Assistent
- Druck

- Einrichtung
- Aktion

Sie müssen die verschiedenen Aktionen vordefinieren, um sie jederzeit durch eine einfache Auswahl in einer Liste ausführen zu können.

---

 **Anmerkung:**

Durch die Definition von Funktionsbereichen können Sie die Aktionen nach ihrem Anwendungsbereich gruppieren.

---

## Funktionsbereiche

AssetCenter ermöglicht die Definition von Bereichen zur Gruppierung der Softwarefunktionen. Bestimmte Funktionsbereiche sind standardmäßig in der Anwendung vorhanden. Dabei handelt es sich um die Module, die über das Menü Datei/ Module aktivieren aktiviert und deaktiviert werden.

Funktionsbereiche dienen zur Erstellung und Klassifizierung der im Funktions- und Favoritenausschnitt angezeigten Daten. Wenn Sie eine Aktion einem Funktionsbereich zuordnen, wird sie im Ausschnitt unter diesem Bereich dargestellt.

---

 **Anmerkung:**

Der Inhalt des Funktions- und Favoritenausschnitts wird je nach Kontext geändert. Kontextuelle Aktionen (beispielsweise Aktionen, die ausgeführt werden können, wenn das Fenster der Personen geöffnet ist) werden beispielsweise nur angezeigt, wenn sie dem aktuellen Kontext entsprechen (in unserem Beispiel die Tabelle der Personen).

---

So definieren Sie einen Funktionsbereich:

- 1 Klicken Sie auf **Neu**.
- 2 Wählen Sie das Menü **Verwaltung/ Funktionsbereiche**.
- 3 Geben Sie einen **Namen** für den Funktionsbereich ein. Er wird später im Funktions- und Favoritenausschnitt angezeigt. Standardmäßig gibt AssetCenter den Funktionsbereichen einen **SQL-Namen**, der vom Benutzer geändert werden kann.
- 4 Wählen Sie ggf. einen **übergeordneten Bereich**.
- 5 Schließen Sie die Erstellung durch Klicken auf **Erstellen** ab.



 **WARNUNG:**

Wenn der Zugriff auf einen Funktionsbereich einem Benutzer vollständig untersagt ist (Lesen und Schreiben), hat dieser keinen Zugang zu den Aktionen, Ansichten und Berichten des Bereichs. In der Praxis ist es ratsam, bei der Definition eines Funktionsbereichs den Lesezugriff auf die folgenden Tabellen zu erlauben:

- amFuncDomain
- amViewDef
- amReport
- amForm
- amAction
- amScriptLibrary

## Aktionen erstellen

In diesem Abschnitt wird die Erstellung von Aktionen erläutert:

- Aktionstypen
- Vorgehensweise
- Informationen auf der Registerkarte Ausführb. Programm oder DDE eingeben
- Informationen auf der Registerkarte Nachrichtensystem eingeben

## Aktionstypen

Mit AssetCenter können Sie verschiedene Aktionstypen definieren.

 **Anmerkung:**

In AssetCenter können Sie nur Aktionen vom Typ **Programm**, **Nachricht** und **Druck** erstellen. Aktionen des Typs **DDE**, **Skript** und **Assistent** gelten als reserviert und können vom Benutzer lediglich ausgeführt werden.

## Aktion vom Typ Programm

Eine Aktion vom Typ **Programm** führt ein externes Programm aus.

Die Aktion startet ein Programm vom Typ **.exe**, **.com**, **.bat**, **.pif**. Zusätzlich besteht die Möglichkeit, eine Beziehung zu den unterschiedlichsten Dokumenten herzustellen, sofern ihre Erweiterung über den Windows Programm-Manager einer Anwendung zugeordnet wurde.

## Aktion vom Typ DDE

Eine Aktion vom Typ **DDE** startet eine DDE-Abfrage mit der Anwendung, die die DDEs verwaltet.

DDE steht für "Dynamic Data Exchange" und bezeichnet die dynamischen Austauschmechanismen zwischen den Anwendungen. In AssetCenter kommen die DDEs zur Anwendung, um die Befehle von einer anderen Anwendung ausführen zu lassen.

Beispiel: Über den DDE können Sie beispielsweise festlegen, dass ein Microsoft Word-Programm eine Datei mit einem spezifischen Inhalt unter einem bestimmten Namen öffnet.

## Aktion vom Typ Nachricht

Eine Aktion vom Typ **Nachricht** ermöglicht das Senden von Nachrichten:

- Über das interne AssetCenter-Nachrichtensystem
- Über ein externes Nachrichtensystem, das unter dem VIM-Standard läuft (Lotus Notes, Lotus cc:Mail usw.)
- Über ein externes Nachrichtensystem, das unter dem MAPI-Standard läuft (Microsoft Exchange, Microsoft Outlook usw.)
- Über ein Nachrichtensystem, das unter dem Internet-Standard SMTP läuft



### WARNUNG:

Die Nachrichten können nur über die Systeme gesendet werden, für die Sie über eine Verbindung verfügen.

Zum Senden einer Nachricht unter Verwendung der Standards VIM, MAPI oder SMTP verwendet AssetCenter folgende Elemente:

- Die Felder **Konto** (SQL-Name: MailLogin) und **Kennwort** (SQL-Name: MailPassword) der Registerkarte **Profil** im Detailfenster der Person, die die AssetCenter-Datenbank geöffnet hat (Tabelle der Abteilungen und Personen), um den Absender zu identifizieren.
- Das Feld **E-Mail** (SQL-Name: EMail) auf der Registerkarte **Allgemein** im Personendetail, um den Absender zu ermitteln.

Zum Senden einer internen AssetCenter-Nachricht verwendet AssetCenter die Felder **Login** (SQL-Name: UserLogin) und **Kennwort** auf der Registerkarte **Profil** im Absender- oder Empfängerdetail.

---

 **Anmerkung:**

Die Adresse des internen Nachrichtensystems eines AssetCenter-Benutzers entspricht seinem **Login**.

---

 **WARNUNG:**

Der Administrator muss einen Benutzer mit dem Namen "Admin" erstellen und die erforderlichen Informationen in die Felder **Konto**, **E-Mail** und **Kennwort** eingeben, um ein externes Nachrichtensystem nutzen und das einwandfreie Funktionieren von AssetCenter Server zu gewährleisten.

---

## Aktion vom Typ Skript

Eine Aktion vom Typ **Skript** kann jeden beliebigen Vorgang in der AssetCenter-Datenbank ausführen. Mit diesem Aktionstyp hat ein fortgeschrittener Benutzer eine besonders gute Kontrolle über die Datenbank, und er kann Operationen ausführen, die mit den anderen Aktionstypen nicht möglich sind. Dazu gehören u. a.:

- Erstellen eines Datensatzes
- Löschen eines Datensatzes
- Duplizieren eines Datensatzes
- Ändern eines oder mehrerer Objekte der AssetCenter-Datenbank, sowie sämtlicher Datensätze einer Tabelle, eines Felds oder einer Verknüpfung.

Die von diesem Aktionstyp ausgeführten Vorgänge werden mit Hilfe eines BASIC-Skripts beschrieben, mit denen die Verwendung komplexer, den AssetCenter API- ähnlichen Funktionen möglich ist.

---

 **Anmerkung:**

Aufgrund der Komplexität der Aktionen vom Typ **Skript** und der Möglichkeit, tiefgreifende Änderungen in der Datenbank vorzunehmen, stellt dieser Aktionstyp eine Gefahr für die Integrität der Datenbank dar und ist daher fortgeschrittenen Benutzern vorbehalten.

---

Je nach Kontext, in der die Aktion zum Einsatz kommt, werden unterschiedliche Funktionen zur Änderung des Werts eines Datenbankobjekts verwendet:

- Wenn für die Aktion kein Kontext vorliegt, müssen Sie die von den AssetCenter API- abgeleiteten Funktionen, wie z. B. **AmSetFieldStrValue()** oder **AmSetFieldLongValue()** verwenden.
- Wenn für die Aktion ein Kontext in Form einer Tabelle vorliegt, ist die Funktion **Set()** mit der nachstehenden Syntax verwendbar:

```
Set [<Verknüpfung.Verknüpfung.Feld>]=<Wert>
```

## Aktion vom Typ Assistent

Die Assistenten führen Sie schrittweise durch komplexe oder systematisch durchzuführende Aufgaben. Für die Realisierung eines Assistenten benötigen Sie eine dedizierte Programmiersprache.



Anmerkung:

Assistenten sind komplexe Aktionen. Ausführliche Informationen zu den Assistenten finden Sie in diesem Handbuch im Kapitel **Assistenten**.

## Aktion vom Typ Druck

Eine Aktion vom Typ **Druck** ermöglicht das Drucken eines Berichts oder Formulars.

Im Folgenden sind die für diesen Aktionstyp zu definierenden Felder aufgeführt:

- Feld **Typ** (seFormType): Ermöglicht das Definieren des zu druckenden Dokuments (Bericht oder Formular).
- Feld **Bericht** oder **Formular**: Ermöglicht das Definieren des Berichts bzw. Formulars, das verwendet werden soll.



Anmerkung:

Der Kontext der Aktion wird über den für den Bericht bzw. das Formular definierten Kontext festgelegt.

## Aktion vom Typ Einrichtung

Eine Aktion vom Typ **Einrichtung** ermöglicht das Starten einer Einrichtung für eine Liste von Computern, die in der Tabelle der Computer definiert sind.

Dabei handelt es sich um eine kontextuelle Aktion in der Tabelle der Computer. Im Folgenden sind die für diesen Aktionstyp zu definierenden Felder aufgeführt:

- **Einrichtungs-Workflow:** Wählen Sie den Einrichtungs-Workflow, den Sie erstellt oder importiert haben.
- **Einrichtungsserver:** Wählen Sie die Anwendung für den Einrichtungsserver.

Weitere Informationen zur Ausführung und Konfiguration der Einrichtung finden Sie im Handbuch **Desktop Administration**.

## Aktion vom Typ Aktion

Eine Aktion vom Typ **Aktion** ermöglicht das Ausführen einer anderen Aktion unter Berücksichtigung der entsprechenden Abfrage.

Im Folgenden sind die für diesen Aktionstyp zu definierenden Felder aufgeführt:

- Feld **Auszuführende Aktion:** Ermöglicht das Definieren der Aktion, die ausgeführt werden soll.
- Feld **Auswahlanfrage:** Ermöglicht das Eingeben des Abfrageskripts, das den Kontext für die Anwendung der Aktion festlegt.

## Vorgehensweise

Gehen Sie zum Erstellen einer Aktion wie folgt vor:

- 1 Wählen Sie das Menü **Extras/ Aktionen/ Bearbeiten**.
- 2 Klicken Sie auf **Neu**.
- 3 Geben Sie der Aktion einen Namen.
- 4 Geben Sie im Feld **Art** (SQL-Name: seActionType) den Aktionstyp an, den Sie erstellen möchten. Je nach dem von Ihnen gewählten Aktionstyp erscheint einer der nachstehenden Registerkarten:
  - **Programm**
  - **DDE**.
  - **Nachricht**
  - **Skript**
  - **Assistent**
  - **Druck**
- 5 Geben Sie die eventuell erforderlichen Informationen im Feld **SQL-Name** (SQL-Name: SQLName) des Aktionsdetails ein. Mit diesem SQL-Namen wird die Aktion eindeutig identifiziert und vor allem bei der Ausführung

einer Aktion mit Hilfe eines DDE-Befehls verwendet. Das ist beispielsweise der Fall, wenn AssetCenter als DDE-Befehlsserver verwendet wird.

---

 **Anmerkung:**

Wenn Sie im Feld **SQL-Name** keine Eingaben machen, generiert AssetCenter einen standardmäßigen SQL-Namen.

---

- 6 Geben Sie die erforderlichen Informationen im Feld Kontext (**SQL-Name:** ContextTable) ein:
    - Wenn Sie eine Tabelle in der Dropdown-Liste wählen, ist die Aktion kontextbezogen und wird daraufhin nur dann vorgeschlagen, wenn Sie die Liste der Datensätze dieser Tabelle oder das entsprechende Detail anzeigen.
    - Wenn die Aktion keiner Tabelle zugeordnet ist, wählen Sie die Option (**Keine Tabelle**) im oberen Teil der Dropdown-Liste.
  - 7 Füllen Sie ggf. das Feld **Bereich** aus, und geben Sie den Funktionsbereich an, zu dem die Aktion gehören soll. Die Aktion wird anschließend im Funktions- und Favoritenausschnitt unter diesem Bereich angezeigt.
  - 8 Ordnen Sie der Aktion gegebenenfalls ein Symbol zu.  
Verwenden Sie dazu das Viereck oben links im Detailfenster der Aktion. Das Bild erscheint anschließend in der Kontextliste Aktionen der Symbolleiste. Bei dem aktiven Symbol dieser Liste, also dem standardmäßig auf dem Bildschirm erscheinenden Symbol, handelt es sich um die zuletzt über die Symbolleiste ausgeführte Aktion.
  - 9 Geben Sie die erforderlichen Informationen in den Feldern der Registerkarte **Beschreibung** sowie in den Feldern der Registerkarten ein, die dem von Ihnen erstellten Aktionstyp entsprechen.
  - 10 Klicken Sie auf **Erstellen**.
- 

 **Anmerkung:**

Der AssetCenter-Administrator kann alle Aktionen anzeigen, wobei es keine Rolle spielt, ob sie gemeinsam genutzt werden oder nicht, und vom wem sie geschrieben wurden.

---

## Informationen auf der Registerkarte Ausführb. Programm oder DDE eingeben

Die Informationen für Aktionen vom Typ DDE erscheinen auf der Registerkarte Ausführb. Programm oder DDE des Aktionsdetails.

Diese Registerkarte wird nur angezeigt, wenn Sie im Feld **Art** (SQL-Name: seActionType) mit den grundlegenden Informationen zu der Aktion den Wert **DDE** eingegeben haben.

Die DDE-Mechanismen stützen sich auf die von der Software gebotenen Dienste. Zum Ausführen eines DDE-Mechanismus müssen Sie ein Thema definieren, das den Kontext beschreibt, in dem die Befehle ausgeführt werden müssen.

Geben Sie folgende Informationen ein:

- Geben Sie im Feld **Service** (SQL-Name: DDEService) den Namen des von Ihnen gewünschten DDE-Dienstes an, der von dem ausführbaren Programm vorgeschlagen werden soll. In den meisten Fällen handelt es sich dabei um einen für das ausführbare Programm eindeutigen Dienst. Eine Liste mit den verfügbaren Diensten finden Sie in der Dokumentation des ausführbaren Programms.
- Geben Sie im Feld **Thema** (SQL-Name: DDETopic) den Kontext an, in dem die Aktion ausgeführt werden soll.
- Geben Sie im Feld **Befehl** (SQL-Name: DDECommand) die Befehle ein, die von der externen Anwendung ausgeführt werden sollen.

Im Fall von Word kann es sich dabei um einen Befehl vom Typ Word Basic oder Visual Basic handeln.

Sofern dies bei dem angeforderten DDE-Dienst möglich ist, können Sie mehrere Befehle nebeneinander plazieren.

Dabei müssen Sie die von der externen Anwendung geforderte Syntax respektieren.

- Wenn der Dienst nicht verfügbar ist, geben Sie im Feld **Datei** (SQL-Name: ActionFile) die Datei ein, mit der die Anwendung gestartet wird, die den Dienst aktiviert. Es handelt sich dabei um die Hauptanwendung, die auf die DDE-Befehle reagiert.

## Wichtiger Hinweis

Die an die externe Anwendung übergebenen Befehle werden in eckige Klammern ("[" , "]" gesetzt. Ein Word-Befehl sieht beispielsweise folgendermaßen aus:

```
[FileOpen("c:\tmp\test.txt")]
```

- Ist die Aktion kontextbezogen, können Sie Variablen einsetzen, die auf den Wert eines Feldes der Datenbank verweisen. Diese Variablen stehen ebenfalls in eckigen Klammern. Da AssetCenter Variablen und Befehle nicht unterscheiden kann, müssen Sie den Klammern das Zeichen "\" voranstellen, um die Befehle abzugrenzen. Das vorstehende Beispiel sieht für eine kontextbezogene Aktion wie folgt aus:

```
\[FileOpen("c:\tmp\test.txt")\]
```

Sie haben die Möglichkeit, Befehle und Variablen wie nachstehend gezeigt zu kombinieren (der Kontext ist die Tabelle der Vermögen und Lose).

```
\[FileOpen("c:\tmp\"+[AssetTag]+".txt")\]\[FileClose()\]\[FileExit()\]
```

- Ist die Aktion nicht kontextbezogen, stellt sich das Problem nicht. Die von eckigen Klammern umschlossenen Texte werden stets als Befehle betrachtet und an die externe Anwendung übergeben.

## Informationen auf der Registerkarte Nachrichtensystem eingeben

Die Informationen für Aktionen vom Typ Nachricht erscheinen auf der Registerkarte Nachrichtensystem des Aktionsdetails.

Diese Registerkarte wird nur angezeigt, wenn Sie im Feld **Art** (SQL-Name: seActionType) mit den grundlegenden Informationen zu der Aktion den Wert **Nachricht** eingegeben haben.

### WARNUNG:

Wenn das Nachrichtensystem einwandfrei funktioniert, muss die Umgebungsvariable PATH Ihres Systems auf das Verzeichnis mit den DLLs für VIM (VIM.DLL für die 16 Bit-Version von AssetCenter und VIM32.DLL für die 32 Bit-Version) und MAPI (MAPI.DLL für die 16 Bit-Version von AssetCenter und MAPI32.DLL für die 32 Bit-Version) zeigen.



## Verwendung des Felds Referenzobjekt (SQL-Name: RefObject)

Über dieses Feld können Sie eine Verknüpfung ab der im Feld **Kontext** (SQL-Name: ContextTable) ausgewählten Tabelle festlegen.

Das Feld kommt nur bei Nachrichten zur Anwendung, die an das interne AssetCenter-Nachrichtensystem gerichtet sind. Es ermöglicht den direkten Zugriff auf das Senderobjekt: Klicken Sie einfach auf die Schaltfläche **Referenzobjekt** im Nachrichtendetail. Wenn es sich bei dem Objekt um den Datensatz handelt, der die Aktion auslöst, brauchen Sie im Feld **Referenzobjekt** (SQL-Name: RefObject) keine Angaben zu machen.

## Empfangsbestätigungen anfordern

Aktivieren Sie das Kästchen **Empfangsbestätigung** (SQL-Name: bAcknowledgment), wenn der Absender in seinem üblichen Nachrichtensystem eine Empfangsbestätigung erhalten soll.

Die Empfangsbestätigung wird an die Adresse gesendet, die im Feld **E-Mail** (SQL-Name: EMail) auf der Registerkarte **Allgemein** im Detailfenster der Person erscheint, die die AssetCenter-Datenbank geöffnet hat (Tabelle der Abteilungen und Personen).



**Anmerkung:**

Für Nachrichten, die vom internen AssetCenter-Nachrichtensystem oder von einem MAPI- oder SMTP-Nachrichtensystem gesendet wurden, können keine Empfangsbestätigungen empfangen werden.

## Adressen angeben

Bei der Angabe einer Adresse haben Sie die Wahl zwischen verschiedenen Vorgehensweisen.

### Adressen im Format

#### <Nachrichtensystem-Engine>:<Nachrichtensystemadresse>

Folgende <Nachrichtensystem-Engines> stehen zur Auswahl:

- AM: Erzwingen der Verwendung des AssetCenter internen Nachrichtensystems
- MAPI: Erzwingen der Verwendung eines Nachrichtensystems mit dem Standard MAPI (Internet Mail, Microsoft Outlook usw.)

- VIM: Erzwingen der Verwendung eines Nachrichtensystems mit dem Standard VIM (Lotus Notes usw.)
- SMTP: Erzwingen der Verwendung eines Nachrichtensystems mit dem Standard SMTP (Internet-Standard)

Die <Nachrichtensystemadresse> weist dasselbe Format auf, wie die herkömmlichen, im Nachrichtensystem verwendeten Adressen. Bei den Adressen des internen Nachrichtensystem handelt es sich um Logins.

Beispieladressen:

- AM:Admin
- MAPI:CathyBernard@taltek.com
- VIM:Cathy Bernard / TALTEK
- SMTP:cbernard@taltek.com

## Adressen im Format <AssetCenter-Login>

In diesem Fall wird das Nachrichtensystem verwendet, das im Feld **E-Mail** (SQL-Name: EMail) der Registerkarte **Allgemein** im Detailfenster der Person eingegeben wurde, deren **Login** (Registerkarte **Profil** im Personendetail) in der Adresse enthalten ist.

Wenn im Feld **E-Mail** keine Angaben gemacht wurden, wird die Nachricht über das interne Nachrichtensystem weitergeleitet.

Beispiel:

- 1 Eine Nachricht wird an die AssetCenter-Logins "Cathy", "Gerald" und "Philip" gesendet.
- 2 In den Feldern **E-Mail** steht "MAPI:CathyBernard@taltek.com" für Cathy und "VIM:Gerald Colombo / Taltek" für Gerald. Das Feld **E-Mail** (SQL-Name: EMail) für "Philip" ist leer.
- 3 Wenn der Sender über ein MAPI-Konto verfügt, wird die Nachricht über MAPI an Cathy und über das interne Nachrichtensystem von AssetCenter an die anderen beiden Empfänger gesendet.
- 4 Wenn der Sender über ein VIM-Konto verfügt, wird die Nachricht an Gerald über VIM und an die anderen beiden Empfänger über das interne Nachrichtensystem von AssetCenter gesendet.

## Adresse mit kontextbezogenen Variablen

Bei einer kontextbezogenen Aktion können Sie die in eckige Klammern [ ] gesetzten Variablen verwenden. Diese Variablen rufen den Wert eines Felds in die AssetCenter-Datenbank.

Beispiel: Geben Sie die Adresse [**Benutzer.Email**] an, um eine Nachricht an den Benutzer des in der Tabelle der Vermögen und Lose ausgewählten Gegenstands zu senden.

## Beispiele für Aktionen

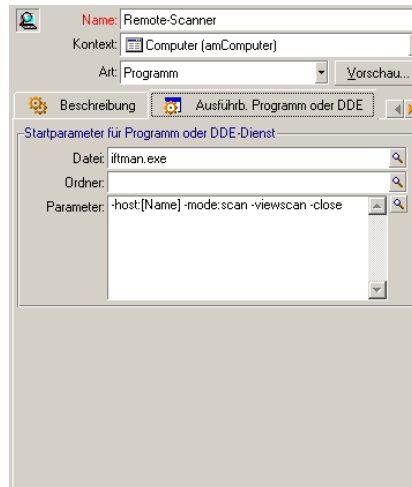
In diesem Abschnitt stellen wir Ihnen einige Beispiele für AssetCenter-Aktionen vor:

- Beispiel einer Aktion vom Typ Programm
- Beispiel einer Aktion vom Typ DDE
- Beispiel einer Aktion vom Typ Nachricht
- Beispiel einer Aktion vom Typ Skript

## Beispiel einer Aktion vom Typ Programm

In der nachstehenden Abbildung ist eine kontextabhängige Aktion dargestellt.

**Abbildung 6.1. Aktion vom Typ Programm - Detailfenster**



## Beispiel einer Aktion vom Typ DDE

Sie haben die Wahl zwischen zahlreichen Anwendungen der Aktionen vom Typ **DDE**:

- Einfügen von AssetCenter-Daten in ein Microsoft-Excel-Tabellenblatt.
- Einfügen von Informationen, die sich auf einen Befehl einer Buchhaltungssoftware beziehen.
- Automatisches Senden einer Telekopie zur Bestätigung der Benachrichtigung bzw. des Abschlusses eines Ereignisses.
- Automatisches Senden einer Telekopie zum Anfordern einer Intervention.
- Sonstige

In diesem Abschnitt wird ein einfaches Beispiel für eine Aktion vom Typ **DDE** gegeben.

### Ziel der Aktion

Sie möchten einen Brief zur Bestätigung einer Beschaffungsanforderung senden. Diese Aktion wird über das Detailfenster einer Anforderung ausgelöst.

Die Aktion verwendet eine DDE-Verknüpfung zwischen Microsoft Word 7 und AssetCenter und fügt die Informationen aus dem Verzeichnis in ein

Word-Dokument ein (Anschrift des Anforderers und Anforderungsnummer), um sie anschließend auszudrucken.

## Vorbereitung des Word-Briefes

Sie müssen zunächst ein Word-Dokument mit dem Titel **Standardbrief.doc** erstellen, das gedruckt werden soll.

Der Standardbrief weist folgendes Format auf:

**Abbildung 6.2. "Standardbrief.doc"**

```

I

                                TALTEK SERVICES
                                San Mateo Site

                                IT Department

<Mr Mrs> <FirstName> <Name>
<Adr1>
<Adr2>
<Zip> <City>

                                San Mateo. { DATE }

Dear <Mr Mrs>,

Following our conversation, I am pleased to confirm that your request has been attributed the
following request number: <ReqNo>.

Regards,

                                Procurement Manager

```

Das Dokument **Standardbrief.doc** wird im AssetCenter-Installationsordner gespeichert.

Die Datei **Normal.dot** wird an die Datei **Standardbrief.doc** angehängt. Diese Datei enthält das Makro **mymacro.bas**:

```

Attribute VB_Name = "MyMacro"
Sub DruckenStandardbrief(MrMrs, FirstName, Name, Adr1, Adr2, Zip, City, ReqNo)
'
' PrintLettreType Macro
'
Application.WindowState = wdWindowStateMinimize 'Run Winword in the back end
Documents.Open ("Standardbrief.doc") 'Open letter pattern
Documents ("Standardbrief.doc").Activate

Selection.Find.ClearFormatting 'Clear parameters for Find function
Selection.Find.Replacement.ClearFormatting 'Clear parameters for Replace function

```

```
With Selection.Find
    .Text = "<MrMrs>"
    .Replacement.Text = MrMrs
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll 'Execute replacement

With Selection.Find
    .Text = "<FirstName>"
    .Replacement.Text = FirstName
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
    .Text = "<Name>"
    .Replacement.Text = Name
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
    .Text = "<Adr1>"
    .Replacement.Text = Adr1
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
    .Text = "<Adr2>"
    .Replacement.Text = Adr2
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
    .Text = "<Zip>"
    .Replacement.Text = Zip
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
    .Text = "<City>"
    .Replacement.Text = City
End With
Selection.Find.Execute Replace:=wdReplaceAll

With Selection.Find
    .Text = "<ReqNo>"
    .Replacement.Text = ReqNo
```

```

End With
Selection.Find.Execute Replace:=wdReplaceAll

ActiveDocument.PrintOut 'Print document
ActiveWindow.Close (wdDoNotSaveChanges) 'Close document w/o updating
MsgBox ("Your document is being printed.") 'Notify user
End Sub

```

## Schritt 1: Erstellen der AssetCenter-Aktion

Gehen Sie zum Erstellen der AssetCenter-Aktion wie folgt vor:

- 1 Öffnen Sie die Tabelle der Aktionen (Menü **Extras/ Aktionen/ Bearbeiten**).
- 2 Klicken Sie auf **Neu**, um eine neue Aktion zu erstellen.
- 3 Geben Sie die erforderlichen Informationen im Feld **Kontext** (SQL-Name: ContextTable) ein, so dass das Feld auf die Tabelle der Anforderungen verweist.
- 4 Legen Sie als Art **DDE** fest.
- 5 Geben Sie die erforderlichen Informationen auf der Registerkarte **Ausführb. Programm oder DDE** ein, so dass die DDE-Verknüpfung zwischen AssetCenter und Microsoft Word 7 definiert wird.

Auf der Registerkarte **Ausführb. Programm oder DDE**:

- 1 Im Feld **Service** (SQL-Name: DDEService) erscheint der Wert "Winword".
- 2 Im Feld **Thema** (SQL-Name: DDETopic) erscheint der Wert "System".
- 3 Im Rahmen **Startparameter für Programm oder DDE-Dienst** erscheint das ausführbare Programm **Winword.exe** mit dem entsprechenden Zugriffspfad.
- 4 Im Feld **Befehl** (SQL-Name: DDECommand) erscheint das zu startende Makro mit den entsprechenden Parametern:

```

\[MyMacro.DruckenStandardbrief "[Requester.MrMrs]", "[Requester.FirstName]", "[Requester.Name]", "[Requester.Location.Address1]", "[Requester.Location.Address2]", "[Requester.Location.ZIP]", "[Requester.Location.City]", "[ReqNumber]" \]

```

Klicken Sie auf die Schaltfläche **Erstellen**, um die Aktion zu bestätigen.

## Schritt 2: Starten einer Aktion

Gehen Sie zum Starten einer Aktion wie folgt vor:

- 1 Öffnen Sie die Tabelle der Anforderungen.
- 2 Wählen Sie eine Beschaffungsanforderung aus.
- 3 Starten Sie die Aktion über das Menü **Extras/ Aktionen**.

Sobald die Aktion gestartet ist, geschieht folgendes:

- 1 Microsoft Word startet und lädt **Standardbrief.doc**.
- 2 Die Anschrift des Anforderers sowie die Anforderungsnummer werden im Brief eingefügt.
- 3 Der Brief wird gedruckt.

## Beispiel einer Aktion vom Typ Nachricht

Sie möchten dem Benutzer eines Vermögensgegenstands über die Liste der Vermögensgegenstände eine Nachricht senden, um ihn auf das Ablaufdatum des entsprechenden Leasingvertrags hinzuweisen. Der Vermögensgegenstand muss als Beschaffungsart den Wert "Leasing" aufweisen und einem Leasingvertrag zugeordnet sein (Registerkarte Beschaffungen). Legen Sie im Aktionsdetail Folgendes fest, damit der Leasingvertrag das Referenzobjekt darstellt:

**Abbildung 6.3. Aktion vom Typ Nachricht mit einem Referenzobjekt - Detailfenster**

The screenshot shows a configuration window for a message action. The fields are as follows:

- Name: Beispiel
- Kontext: Vermögen (amAsset)
- Art: Nachricht
- Referenzobjekt: AcquContract
- Priorität: Normal
- Empfangsbestätigung:
- An: [Portfoliitem.User.Email]
- Cc: [Empty]
- Bcc: [Empty]
- Betreff: [AcquContract.cf\_self]
- Text: [AcquContract.dEnd]

## Beispiel einer Aktion vom Typ Skript

Die Erstellung einer Aktion vom Typ **Skript** beschränkt sich hauptsächlich auf das Schreiben eines BASIC-Skripts, mit dem die AssetCenter-Datenbank geändert wird.




 **Anmerkung:**

Innerhalb dieser Skripte können aktionsspezifische Funktionen verwendet werden. Einen Index dieser Funktionen finden Sie im Handbuch "Programmer's Reference", Kapitel "Function indexes per application fields, Abschnitt "Builtin functions".

## Vorbereitung

Gehen Sie bei der Vorbereitung zur Erstellung der Aktion wie folgt vor:

- 1 Wählen Sie das Menü **Extras/ Aktionen/ Bearbeiten** und klicken Sie auf die Schaltfläche **Neu** im Detailfenster der Aktionen.
- 2 Geben Sie der Aktion, die Sie erstellen möchten, einen Namen, beispielsweise "Test", und definieren Sie für das Feld **Art** (SQL-Name: seActionType) den Wert **Skript**. Wählen Sie keinen Kontext für die Aktion. Klicken Sie auf **Erstellen**.
- 3 Klicken Sie auf der Registerkarte **Skript** auf die Schaltfläche , um das Fenster mit dem Skripteditor anzuzeigen. Die für diese Aktionen verwendete programmierbare Funktion **Success()** erfordert keinen ausdrücklichen Rückgabewert. Im nun folgenden Beispiel erstellen wir einen neuen Datensatz in der Tabelle der Kategorien, und zwar auf der Grundlage der in der nachstehenden Tabelle enthaltenen Informationen.

Feldbezeichnung	SQL-Name des Felds	Wert des Felds
<b>Name</b>	Name	Mikrocomputer
<b>Erstellt</b>	seBasis	Ausrüstungselement
<b>Kann verbunden werden</b>	bIsCnxClient	Kontrollkästchen markiert

## Schreiben des Skripts

Geben Sie das nachstehende Skript ein:

```
Dim lrec As Long
Dim lres As Long
  lrec=AmCreateRecord("amNature")
  lres=AmSetFieldStrValue(lrec, "Name", "Mikrocomputer")
  lres=AmSetFieldStrValue(lrec, "seBasis", 1)
  lres=AmSetFieldStrValue(lrec, "bIsCnxClient", 1)
AmInsertRecord(lrec)
```



Anmerkung:

Diese Aktion erstellt die gewünschte Art ohne Zutun des Benutzers.

## Beispiel einer Verwendung der Funktion "Set()"

Wir werden jetzt dieselbe Art anhand einer Aktion vom Typ **Skript** erstellen, und dabei die Tabelle der Arten als Kontext für die Aktion definieren. Das Skript liest sich wie folgt:

```
Set [Name]="Mikrocomputer"
Set [seBasis]=1
Set [bIsCnxClient]=1
```



Anmerkung:

Zum Ausführen dieser Aktion muss der Benutzer die Tabelle der Kategorien öffnen und auf die Schaltfläche **Neu** klicken. Sobald die Aktion ausgeführt ist, muss der Benutzer auf die Schaltfläche **Erstellen** klicken, um die Erstellung zu bestätigen.

## Tipp


Wenn Sie die Ausführung einer Aktion innerhalb eines Skripts aufheben möchten, geben Sie für den Rückgabewert einfach einen Wert ungleich 0 ein (z. B. 12001). Der Wert wird als Fehlercode angesehen. Der nächste Befehl unterbricht die Aktion und annulliert die bereits ausgeführten Änderungen.

```
RetVal=12001
```

## Arbeiten mit Variablen

In den Registerkarten **Programm**, **Ausführb. Programm** oder **DDE** oder **Nachrichtensystem** im Detailfenster einer kontextbezogenen Aktion können Sie Variable verwenden, die sich auf den Inhalt von Feldern oder Merkmale der Datenbank beziehen.

Diese Variablen erscheinen im Format [**Verknüpfung.Verknüpfung.Feld**].

Wenn Sie bei der Eingabe dieser Variablen Unterstützung brauchen, klicken Sie auf die Schaltfläche  rechts neben dem auszufüllenden Feld.

Alles, was nicht in eckige Klammern [] gefasst ist, gilt als Text.

Beispiel: [**Verknüpfung.Verknüpfung.Feld**].doc ruft den Wert aus dem Feld **Feld** in die Tabelle, die mit der Haupttabelle verknüpft ist, und greift dabei auf die Verknüpfungen **Verknüpfung.Verknüpfung** zurück.

---

 **WARNUNG:**

Damit die Variablen einwandfrei funktionieren, muss im Feld **Kontext** (SQL-Name: ContextTable) des Aktionsdetails eine AssetCenter-Tabelle erscheinen, und Sie müssen einen Datensatz in der Datensatzliste der Tabelle ausgewählt haben, bevor Sie die Aktion ausführen.

---

## Sonderfall der Sybase SQL Anywhere-Engine

Wenn es sich bei der Datenbank-Engine für AssetCenter um Sybase SQL Anywhere handelt, ist das Schreiben von {"d" oder "m" am Anfang eines Felds im Detailfenster einer Aktion nicht möglich.

Wenn ein Feld im Detailfenster einer Aktion mit {"d" oder "m" beginnen soll, empfehlen wir Ihnen, diesen Zeichenfolgen ein Leerzeichen voranzustellen.

## Aktionen testen

Wenn Sie eine Aktion bei ihrer Erstellung testen möchten, klicken Sie auf die Schaltfläche **Vorschau** oben rechts im Detailfenster der entsprechenden Aktion.

## Schaltfläche Berechnen





Sobald Sie den Kontext ausgewählt haben, klicken Sie auf die Schaltfläche **Berechnen**. Daraufhin werden die Daten in die Felder der Registerkarten **Ausführb. Programm oder DDE** bzw. **Nachrichtensystem** gerufen. Vergewissern Sie sich, dass die Variablen ordnungsgemäß aus dem im Feld **Kontext** (SQL-Name: ContextTable) ausgewählten Datensatz extrahiert wurden.

## Schaltfläche Ausführen

Durch Klicken auf diese Schaltfläche können Sie die Aktion direkt über diesen Bildschirm ausführen.

## Aktionen ausführen

Beim Ausführen einer Aktion haben Sie die Wahl zwischen verschiedenen Möglichkeiten:

- Verwendung der Dropdownliste  in der Symbolleiste:
  - Anstelle der Schaltfläche  erscheint das Symbol, das der zuletzt auf der Arbeitsstation durchgeführten Aktion entspricht, sofern dieses Symbol existiert. Wenn bereits eine Aktion ausgeführt wurde, klicken Sie auf die Schaltfläche  oder auf das an derselben Stelle erscheinende Symbol, um die Aktion erneut auszuführen.
  - Über die Schaltfläche  rufen Sie die Liste der verfügbaren Aktionen ab.
  - Verwenden Sie das Menü Extras/ Symbolleiste anpassen, um diese Dropdown-Liste in die Symbolleiste einzufügen: Die Liste gehört zur Kategorie "Extras".
- Verwendung des Menüs Extras/ Aktionen: Wählen Sie die gewünschte Aktion im Untermenü.
- Verwendung der Schaltfläche **Seitenansicht** oben rechts im Aktionsdetail:
  - Wenn es sich um eine kontextbezogene Aktion handelt, müssen Sie den Kontext durch Auswahl eines Datensatzes in der Referenztable der Aktion auswählen.
  - Klicken Sie auf **Ausführen**, um die Aktion auszuführen.
- Verwendung des Kontextmenüs, das Sie mit einem Druck auf die rechte Maustaste aufrufen. Wenn für die offene Tabelle wenigstens eine Aktion zur Verfügung steht, erscheint der Eintrag Aktionen im Kontextmenü.

## Mehrere Datensätze in einer Liste auswählen

Sie können mehrere Datensätze in einer Liste auswählen, und eine Aktion für sie durchführen.

Das bedeutet, dass Sie beispielsweise mehrere Vermögensgegenstände auswählen und eine Nachricht an alle Benutzer dieser Gegenstände schicken können.

## Aktionen vom Typ Assistent

Assistenten setzen sich aus mehreren Seiten zusammen. Auf jeder dieser Seiten erscheinen Informationen, und der Benutzer muss Daten eingeben oder Auswahlen treffen.

Das Navigieren zwischen den verschiedenen Seiten eines Assistenten ist denkbar einfach:

- Sobald Sie alle erforderlichen Daten auf einer Seite eingegeben haben, gehen Sie durch Klicken auf die Schaltfläche **Weiter** zur nächsten Seite: man spricht hierbei von einem Übergang. Auf der letzten Seite des Assistenten fehlt ein solcher Übergang.
- Durch Klicken auf die Schaltfläche **Zurück** können Sie um eine Seite zurückgehen, um eventuell Korrekturen vorzunehmen.
- Durch Klicken auf die Schaltfläche **Fertig stellen** lässt sich jederzeit die letzte Aktion eines Assistenten ausführen. Sollten die verfügbaren Informationen nicht ausreichen, um die geforderte Aufgabe auszuführen, wird der Benutzer zur entsprechenden Seite zurückgeführt.

Durch Klicken auf die Schaltfläche **Abbrechen** können Sie die Ausführung eines Assistenten und der entsprechenden Aktion abbrechen.

## Aktionen und Aktionsschaltflächen zuordnen

Es bestehen mehrere Möglichkeiten, einer Aktion eine Aktionsschaltfläche zuzuordnen.

Weitere Informationen dazu finden Sie an folgenden Stellen:

- Handbuch **Einführung in AssetCenter**, Kapitel **Anpassung eines Clients**, Abschnitt **Anpassung der Schaltflächen**
- Handbuch **Verwaltung**, Kapitel **Anpassung der Datenbank**, Abschnitt **Vorhandene Objekte anpassen/ Objekte anpassen / Details anpassen / Schaltflächen erstellen**
- Handbuch **Verwaltung**, Kapitel **Anpassung der Datenbank**, Abschnitt **Neue Objekte erstellen/ Aktionsschaltflächen erstellen**



# 7 | Nachrichtensystem

---

## KAPITEL

AssetCenter bietet die Möglichkeit der Verwaltung von zwei Arten von Nachrichten:

- Nachrichten, die von AssetCenter über das interne Nachrichtensystem an die AssetCenter-Datenbank gesendet werden
- Nachrichten, die in AssetCenter erstellt und an das externe Nachrichtensystem des Absenders weitergeleitet werden

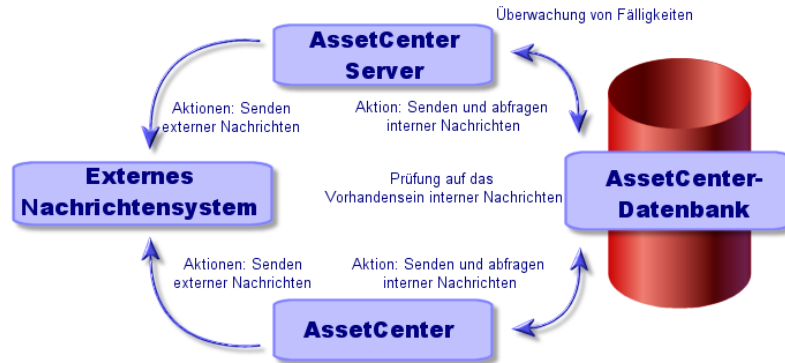
## Allgemeine Funktionsweise des Nachrichtensystems

AssetCenter unterstützt folgende Protokolle zum Senden von Nachrichten:

- AM (AssetCenter)
- SMTP
- MAPI
- VIM

Beim Nachrichtenempfang werden Nachrichten des Typs AM (AssetCenter) unterstützt.

Abbildung 7.1. Nachrichtensystem - Allgemeine Funktionsweise



## Nachrichten senden

Die Generierung von Nachrichten erfolgt über Aktionen vom Typ **Nachricht**. Bevor eine Nachricht gesendet werden kann, muss die entsprechende Aktion erstellt werden.

Die Aktion kann auf verschiedene Arten ausgelöst werden:

- Manuell durch Auswahl der Aktion in der Liste des Menüs **Extras/ Aktionen**
- Automatisch über AssetCenter Server
- Automatisch über AssetCenter

Die Erstellung von Aktionen des Typs Nachricht wird im Abschnitt **Informationen auf der Registerkarte Nachrichtensystem eingeben** [Seite 104] beschrieben.

## Nachrichten abfragen

### An das interne Nachrichtensystem gesendete Nachrichten abfragen

Ein Agent sucht im System nach neuen internen Nachrichten und weist ggf. darauf hin, dass neue Nachrichten vorliegen.

Beim Abfragen von Nachrichten haben Sie die Wahl zwischen zwei Vorgehensweisen:

- Über das Menü **Extras/ Nachrichten**



- Über das Dialogfeld, das Sie auf das Vorhandensein neuer Nachrichten aufmerksam macht

## An das externe Nachrichtensystem gesendete Nachrichten abfragen

Der Empfänger kann die Nachrichten wie in einem herkömmlichen Mail-Programm abfragen.

## Empfangsbestätigungen

---

 Anmerkung:

Die Empfangsbestätigung kann nicht direkt als Nachricht empfangen werden, die über das interne AssetCenter Nachrichtensystem oder über ein Nachrichtensystem unter Verwendung des Standards MAPI oder SMTP gesendet wurde.

---

Im Detailfenster von Aktionen des Typs **Nachricht** können Sie im Feld **Empfangsbestätigung** (SQL-Name: bAcknowledgment) festlegen, dass Sie eine Empfangsbestätigung erhalten möchten.

Die Empfangsbestätigung wird an die Adresse gesendet, die im Feld **E-Mail** (SQL-Name: EMail) auf der Registerkarte **Allgemein** im Detail der Person erscheint, die die AssetCenter-Datenbank geöffnet hat (Tabelle der Abteilungen und Personen).



# 8 | Workflows

## KAPITEL

In diesem Kapitel wird die Definition und Verwaltung von Workflow-Modellen mit AssetCenter erklärt:

### Definition der Konzepte im Rahmen der Workflow-Modelle

In diesem Abschnitt werden die Grundlagen des Workflows erklärt:

- Definition des Workflows
- Definition einer Workflow-Aktivität
- Definition eines Workflow-Ereignisses
- Definition eines Workflow-Übergangs
- Definition einer Workflow-Aufgabe
- Definition des Bearbeiters einer Workflow-Aktivität
- Definition einer Workflow-Ausführungsgruppe

## Definition des Workflows

Unter dem Workflow ist eine Formalisierung bzw. Automatisierung von Geschäftsprozessen zu verstehen.

So können beispielsweise die nachstehenden Verfahren mit Hilfe eines Workflow-Modells dargestellt und automatisiert werden:

- Prozesse zur Bestätigung von Beschaffungsanforderungen
- Umzug von Vermögensgegenständen
- Andere

AssetCenter ermöglicht das Definieren von Workflow-Modellen und das Verwalten ihres Ablaufs.

## Workflow-Modelle

Bei der Erstellung eines Workflow-Modells in AssetCenter definieren Sie die nachstehenden Elemente:

- Aktivitäten
- Ausgabeereignisse von Aktivitäten, mit denen die entsprechenden Übergänge aktiviert werden
- Übergänge, die Aktivitäten auslösen
- Einen Kontext
- Fristen und Warnungen

**Abbildung 8.1. Workflows - Vereinfachtes Modell**



Der Zugriff auf die Tabelle der Workflow-Modelle (SQL-Name: "amWfScheme") erfolgt über das Menü Extras/ Workflow/ Workflow-Modelle.

## Workflow-Instanz

Auf den folgenden Seiten bezeichnen wir den Ablauf eines Workflow-Modells als "Workflow-Instanz".

## Definition einer Workflow-Aktivität

Eine Workflow-Aktivität setzt sich aus folgenden Elementen zusammen:

- Einer auszuführenden Arbeit. Diese Arbeit kann sowohl von einem Benutzer als auch automatisch von AssetCenter Server ausgeführt werden.
- Ausgabeereignisse, mit denen der Übergang zu anderen Aktivitäten ausgelöst wird.

Die Workflow-Aktivitäten werden in der entsprechenden Tabelle unter dem SQL-Namen "amWfActivity" gespeichert.

Der Zugriff auf das Detail einer Workflow-Aktivität erfolgt über den Grafikeditor auf der Registerkarte Aktivitäten des Workflow-Details.

## Definition eines Workflow-Ereignisses

Workflow-Ereignisse entstehen im Anschluß an eine Aktivitätsausgabe und ermöglichen das Aktivieren von Übergängen, die wiederum andere Aktivitäten auslösen.

Die Ereignisse, die zu einem Workflow-Modell gehören, werden in der Tabelle der Workflow-Ereignisse unter dem SQL-Namen "amWfEvent" gespeichert.

Wenn diese Ereignisse dann tatsächlich stattfinden, können sie in der Tabelle mit den grundlegenden Ereignissen einer Workflow-Instanz unter dem SQL-Namen "amWfOccurEvent" gespeichert werden.

Der Zugriff auf das Detail eines Workflow-Ereignisses erfolgt über den Grafikeditor, der auf der Registerkarte Aktivitäten eines Workflow-Details enthalten ist.

## Definition eines Workflow-Übergangs

Ein Workflow-Übergang ermöglicht den Wechsel von einer Aktivität zur nächsten. Ein Übergang wird von einem Ereignis ausgelöst.

Ein Ereignis kann mehreren Übergängen zugeordnet sein.

Die Übergänge, die zu einem bestimmten Workflow-Modell gehören, werden in der Tabelle mit den Workflow-Übergängen unter dem SQL-Namen "amWfTransition" gespeichert.

Der Zugriff auf das Detail eines Workflow-Übergangs erfolgt über die Registerkarte Aktivitäten eines Workflow-Details.

## Definition einer Workflow-Aufgabe

Eine Workflow-Aufgabe ist eine Arbeit, die aufgrund einer ausgelösten Aktivität durchgeführt werden muss.

Wenn Sie eine Workflow-Aufgabe in der entsprechenden Tabelle unter dem SQL-Namen "amWfWorkItem" speichern wollen, müssen Sie die Option **Aufgabe speichern** (SQL-Name: bLogWorkItem) auf der Registerkarte **Allgemein** im Detailfenster der Aktivität bestätigen, die zur Ausführung der Aufgabe geführt hat.

Der Zugriff auf die Liste mit den Workflow-Aufgaben erfolgt über das Menü Extras/ Laufende Aufgaben.

## Definition des Bearbeiters einer Workflow-Aktivität

Die Bearbeiter müssen die Aufgaben durchführen, die aus den Aktivitäten vom Typ **Frage** oder **Benutzeraktion** heraus entstanden sind. Für die Aktivitäten vom Typ **Automatische Aktion** oder **Test / Skript** sind keine Bearbeiter vorhanden.

Die Bearbeiter werden in der Tabelle der Workflow-Rollen (SQL-Name: amWfOrgRole) gespeichert. Der Zugriff auf diese erfolgt über das Menü Extras/ Workflow/ Rollen.

## Definition einer Workflow-Ausführungsgruppe

Die von Ihnen definierten Workflow-Modelle lassen sich mit Hilfe von Workflow-Ausführungsgruppen klassifizieren. Die zu den einzelnen Workflow-Modellen gehörenden Ausführungsgruppen sind dem Feld **Gruppe** (SQL-Name: GroupName) der Registerkarte **Allgemein** im Detailfenster des Workflows zu entnehmen.

## Allgemeine Funktionsweise

Der erste Schritt bei der Verwaltung von Workflows mit AssetCenter ist die Definition eines Workflow-Modells mit Hilfe des Grafikeditors, auf den Sie über das Menü **Extras/ Workflow/ Workflow-Modelle** zugreifen.

Workflow-Modelle dienen zur Definition von Aktivitäten, Ereignissen und Übergängen. Dabei können sich die Modelle sowohl auf AssetCenter-Aktionen als auch auf Workflow-Bearbeiter beziehen.

Die Workflow-Modelle werden von Workflow-Engines interpretiert. Bei den Workflow-Engines von AssetCenter handelt es sich entweder um AssetCenter Server, oder um AssetCenter-Agenten.

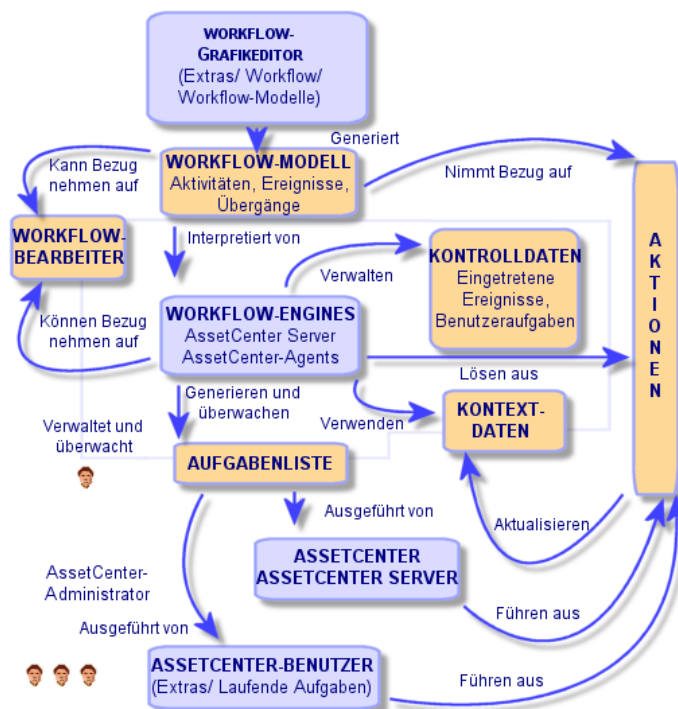
Beim Eintreten eines Ereignisses lösen die Workflow-Engines eine Workflow-Instanz aus, und steuern den anschließenden Prozess:

- Die Workflow-Engines generieren die auszuführenden Aufgaben.
- Die Workflow-Engines überwachen die Ausführung dieser Aufgaben sowie das Eintreten von Ereignissen, die Aktivitäten auslösen.
- Die Workflow-Engines können den Ablauf des Workflow-Prozesses protokollieren, indem sie die eingetretenen Ereignisse und die auszuführenden Benutzeraufgaben speichern.

Die Workflow-Aufgaben werden entweder von den Workflow-Engines oder von einem AssetCenter Benutzer ausgelöst. Sobald die Aufgaben ausgeführt worden sind, aktivieren sie Ereignisse, die von den Workflow-Engines berücksichtigt wurden.

Das nachstehende Schema gibt Ihnen einen Überblick über den Workflow, wie er in AssetCenter implementiert worden ist:

Abbildung 8.2. AssetCenter-Workflow - Allgemeine Übersicht



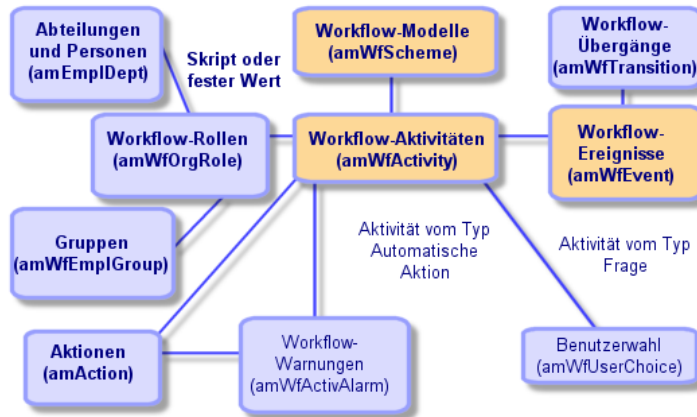
Je nach den Parametern, die für Aktivitäten und Ereignisse definiert worden sind, und der Konfiguration der Workflow-Engines, die diese Aktivitäten und Ereignisse steuern, sind verschiedene Bearbeitungsweisen der Workflow-Instanzen zu unterscheiden.

## Für den Workflow hauptsächlich verwendete Tabellen

Aus dem nachstehenden Schema entnehmen Sie die Tabellen, die zur Definition von Workflow-Modellen herangezogen werden, sowie die Verknüpfungen, die diesen Tabellen zugeordnet sind. Die Identifizierung dieser Tabellen erfolgt anhand ihrer Bezeichnung und ihres SQL-Namens:

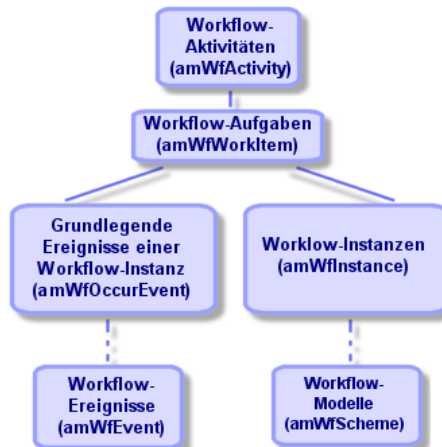


**Abbildung 8.3. Workflow - Zum Definieren eines Workflow-Modells hauptsächlich verwendete Tabellen**



Das nachstehende Schema zeigt die beim Ablauf einer Workflow-Instanz hauptsächlich verwendeten Tabellen:

**Abbildung 8.4. Workflow - Im Rahmen des Ablaufs einer Workflow-Instanz hauptsächlich verwendete Tabellen**



## Arbeiten mit dem Workflow-Grafikeditor

Der Zugriff auf die Liste und das Detailfenster mit den Workflow-Modellen von AssetCenter erfolgt über das Menü Extras/ Workflow/ Workflow-Modelle. Nur die AssetCenter-Administratoren können dieses Menü aktivieren.


Die Registerkarte **Aktivitäten** im Detailfenster eines Workflow-Modells zeigt einen Grafikeditor, der die Erstellung von Workflow-Modellen besonders benutzerfreundlich gestaltet.

Im nun folgenden Abschnitt wird das Erstellen, Ändern und Löschen von Workflow-Elementen mit dem Grafikeditor definiert:


- Aktivitäten
- Ereignisse
- Übergänge
- Weitere Funktionalitäten

### Aktivitäten

Gehen Sie zum Erstellen einer Aktivität wie folgt vor:

- Klicken Sie mit der rechten Maustaste auf einen leeren Bereich auf der Registerkarte **Aktivitäten**, und wählen Sie in dem daraufhin erscheinenden Kontextmenü die Option **Aktivität hinzufügen**. Es erscheint das Detailfenster der Aktivitäten; oder
- Klicken Sie auf die Schaltfläche  und dann auf die Grafikzone. Der Detailbereich der Aktivität erscheint.

Gehen Sie zum Löschen einer Aktivität wie folgt vor:

- Klicken Sie mit der linken Maustaste auf die gewünschte Aktivität, und drücken Sie dann auf die Taste "Entf" der Tastatur; oder
- Klicken Sie mit der rechten Maustaste auf die gewünschte Aktivität, und wählen Sie in dem daraufhin erscheinenden Kontextmenü die Option **Löschen**; oder
- Wählen Sie eine Aktivität, und klicken Sie dann auf die Schaltfläche .

Gehen Sie zum Ändern des Details einer Aktivität wie folgt vor:

- Setzen Sie den Cursor auf die Aktivität, drücken Sie die rechte Maustaste und wählen Sie in dem daraufhin erscheinenden Kontextmenü die Option **Aktivitätendetail**, oder


- Wählen Sie eine Aktivität und drücken Sie die linke Maustaste.




## Ereignisse

### Ereignisse vom Typ **Bei Datenbankänderung** oder **In regelmäßigen Abständen**

Gehen Sie zum Hinzufügen eines solchen Ausgabeereignisses einer Aktivität wie folgt vor:

- Setzen Sie den Cursor auf die Aktivität, und drücken Sie die rechte Maustaste. Wählen Sie in dem daraufhin erscheinenden Kontextmenü die Option Ereignis hinzufügen oder
- Wählen Sie die Aktivität, und klicken Sie auf die Schaltfläche .

Gehen Sie zum Löschen eines Ereignisses vom Typ **Bei Datenbankänderung** oder **In regelmäßigen Abständen** einer Aktivität wie folgt vor:

- Klicken Sie mit der linken Maustaste auf das gewünschte Ereignis, und drücken Sie dann auf die Taste "Entf" der Tastatur; oder
- Klicken Sie mit der rechten Maustaste auf das gewünschte Ereignis, und wählen Sie in dem daraufhin erscheinenden Kontextmenü die Option **Löschen**; oder
- Wählen Sie ein Ereignis, und klicken Sie dann auf die Schaltfläche .

Gehen Sie zum Ändern des Details eines Ereignisses vom Typ **Bei Datenbankänderung** oder **In regelmäßigen Abständen** wie folgt vor:

- Wählen Sie das Ereignis mit einem Druck auf die linke Maustaste und öffnen Sie dann das Fenster zum Ändern des Ereignisses mit einem Doppelklick.
- Klicken Sie mit der rechten Maustaste auf das gewünschte Ereignis, und wählen Sie in dem daraufhin erscheinenden Kontextmenü die Option **Ereignisdetail**.

## Ereignis vom Typ System

Das Erstellen und Löschen der Ereignisse vom Typ **System** erfolgt über das Detailfenster der Aktivitäten, aus denen Sie stammen.

Wenn Sie das Detail eines Ereignisses vom Typ **System** ändern möchten, wählen Sie eine der beiden nachstehend beschriebenen Vorgehensweisen:


- Zum Ändern der Vorgehensweise bei der Bearbeitung eines Ereignisses (Feld **Bearbeitung** (SQL-Name: seProcessingMode) im Ereignisdetail), gehen Sie vor, als ob Sie ein Ereignis vom Typ **Bei Datenbankänderung** oder **In regelmäßigen Abständen** ändern würden.
- Ansonsten bearbeiten und ändern Sie das Detail der Aktivität, das zu dem Ereignis geführt hat.

## Übergänge

Gehen Sie zum Erstellen eines Übergangs wie folgt vor:

- 1 Wählen Sie durch Drücken der linken Maustaste das Ausgangsereignis für den Übergang.
- 2 Halten Sie die linke Maustaste niedergedrückt und ziehen Sie sie bis zur Zielaktivität.

Gehen Sie zum Löschen eines Übergangs wie folgt vor:

- Klicken Sie mit der linken Maustaste auf den gewünschten Übergang, und drücken Sie dann auf die Taste "Entf"; oder
- Klicken Sie mit der rechten Maustaste auf den gewünschten Übergang, und wählen Sie in dem daraufhin erscheinenden Kontextmenü die Option **Löschen**; oder
- Wählen Sie einen Übergang, und klicken Sie dann auf die Schaltfläche .

Gehen Sie zum Ändern eines Übergangsdetails wie folgt vor:


- 1 Wählen Sie den Übergang durch Klicken mit der rechten Maustaste.
- 2 Wählen Sie in dem daraufhin erscheinenden Kontextmenü die Option **Übergangsdetail**.

Gehen Sie zum Ändern des Ausgangs bzw. des Ziels eines Übergangs wie folgt vor:

- 1 Wählen Sie den Übergang.
- 2 Ziehen Sie den Cursor auf die gewünschte Position.

## Weitere Funktionalitäten

Der Grafikeditor bietet Ihnen weitere Möglichkeiten:

- Ziehen und Ablegen (Drag & Drop) der Aktivitäten und der damit verknüpften Übergänge.
- Vergrößern bzw. Verkleinern des Schemas über den Regler **Zoom** oder die Schaltfläche .

## Workflow-Modelle einrichten

Die Workflow-Modelle in AssetCenter ermöglichen Ihnen das Automatisieren von Geschäftsprozessen. Folgen Sie dazu den nachstehenden Anweisungen:

- 1 Analysieren Sie die Geschäftsprozesse, für die Sie Modelle definieren wollen.
- 2 Erstellung:
  - 1 Workflow-Rollen
  - 2 Aktionen
- 3 Beim Erstellen der Workflow-Modelle definieren Sie folgende Elemente:
  - 1 Aktivitäten, Ereignisse und Übergänge
  - 2 Warnungen
- 4 Definieren Sie die für die Workflow-Modelle geeigneten Ausführungsgruppen. Ordnen Sie die einzelnen Workflow-Modelle einer Ausführungsgruppe zu.
- 5 Starten Sie AssetCenter Server auf einem oder mehreren Rechnern. Definieren Sie für jedes AssetCenter Server-Programm die Ausführungsgruppen des Workflow-Modells, die vom Programm überwacht werden sollen, und legen Sie die entsprechenden Parameter fest.

## Beispiel eines Workflow-Modells für die Bewilligung von Anforderungen

In diesem Abschnitt stellen wir Ihnen ein Workflow-Modell anhand eines einfachen Beispiels vor:

- 1 [Ziel und Zweck](#) [Seite 134]
- 2 [Voraussetzungen](#) [Seite 135]

- 3 Aktivitäten erstellen [Seite 140]
- 4 Parameter für die mit den Aktivitäten erstellten Ereignisse festlegen [Seite 145]
- 5 Ausgangsereignisse erstellen [Seite 146]
- 6 Übergänge erstellen [Seite 147]
- 7 Beispiele zum Auslösen des Workflows [Seite 148]

## Ziel und Zweck

Die Bewilligung von Beschaffungsanforderungen soll anhand des nachstehenden Workflow-Modells automatisiert werden:

**Abbildung 8.5. Workflow - Modell zur Bewilligung von Anforderungen**



Im Folgenden sind die einzelnen Schritte des Workflows aufgeführt:

- 1 Die Workflow-Instanz beginnt mit dem Vorliegen einer Anforderung, die bewilligt werden soll, also sobald im Feld **Anforderungsstatus** (SQL-Name: seStatus) im Anforderungsdetail der Wert **Zu bewilligen** erscheint.
- 2 Die Anforderung durchläuft zunächst den Schritt der funktionellen Bewilligung. In diesem Schritt wird der Leiter der Abteilung, in der der Benutzer arbeitet, zur Bewilligung oder Ablehnung der Anforderung aufgefordert. Der Abteilungsleiter wird durch eine Nachricht auf das Vorliegen einer Anforderung aufmerksam gemacht. Wenn der Bewilliger bis zum Ende des darauffolgenden Tages nicht reagiert hat, sendet das System eine Warnmeldung.
- 3 Wenn der Abteilungsleiter die Beschaffungsanforderung bewilligt, besteht der nächste Schritt in der finanziellen Bewilligung. Ansonsten ist die Anforderung abgelehnt (3b).

- 4 Bei der finanziellen Bewilligung wird der Finanzprüfer des Unternehmens, Herr Gerald Colombo, zur Bewilligung der Anforderung aufgefordert. Auch der Finanzprüfer wird mittels einer Nachricht auf die erforderliche Bewilligung aufmerksam gemacht, und das System sendet eine Warnmeldung, wenn die Anforderung nicht innerhalb einer bestimmten Frist bearbeitet worden ist.
- 5 Wenn der Finanzprüfer die Anforderung bewilligt, gilt die Anforderung als genehmigt.  
Ansonsten ist die Anforderung abgelehnt (5b).
- 6 Wenn die Anforderung bewilligt ist, ändert AssetCenter das Feld **Anforderungsstatus** im Anforderungsdetail, so dass der Wert **Bewilligt** erscheint.  
Wenn die Anforderung abgelehnt wird, ändert AssetCenter den Wert **Anforderungsstatus** (SQL-Name: seStatus) im Anforderungsdetail in den Wert **Abgelehnt**.

## Voraussetzungen

Sie müssen über das Login **Admin** eine Verbindung mit der Datenbank herstellen und das Nachrichtensystem konfigurieren (► [Nachrichtensystem](#) [Seite 119]).

## Workflow-Bearbeiter erstellen

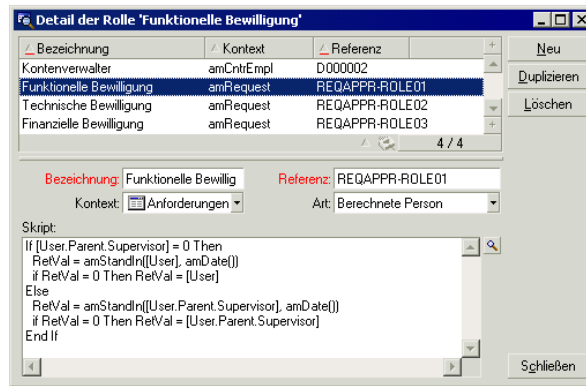
Bei den Workflow-Bearbeitern, die im Workflow-Modell bestimmte Aufgaben übernehmen, handelt es sich um folgende Personen:

- Die Person, die in der Hierarchie über dem Anforderer steht.
- Den Finanzprüfer des Unternehmens, Herrn Gerald Colombo.

Die Workflow-Bearbeiter sind in der Tabelle der Workflow-Rollen (SQL-Name: amWfOrgRole) gespeichert. Zum Erstellen der Bearbeiter wählen Sie das Menü **Extras/ Workflow/ Rollen** und klicken dann auf **Neu**.

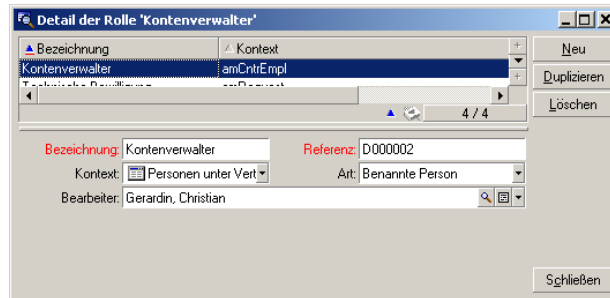
## Leiter der Abteilung, in der der Anforderer tätig ist

Der Leiter wird mit Hilfe eines Skripts berechnet. Geben Sie zum Definieren des Leiters folgende Informationen im Detailbildschirm ein:



## Finanzprüfer

Es handelt sich um eine bestimmte Person, Herrn Gerald Colombo. Zum Definieren des Finanzprüfers als Bearbeiter geben Sie folgende Informationen im Detailbildschirm ein:



## Aktionen erstellen

Das Workflow-Modell greift auf zahlreiche Aktionen zurück. Verwenden Sie zum Erstellen der Aktionen das Menü **Extras/ Aktionen/ Bearbeiten**.



## Aufforderung zur funktionellen Bewilligung an den Abteilungsleiter des Anforderers

Diese Aktion kommt im Rahmen der funktionellen Bewilligung zur Anwendung. Sie benachrichtigt den für die funktionelle Bewilligung zuständigen Mitarbeiter.

The screenshot shows a dialog box titled "Detail der Aktion 'Anforderung der funktionellen Bewilligung'". It contains the following fields and controls:

- Name:** Anforderung der funktionellen Bewilligung
- Kontext:** Workflow-Aufgaben (amWfWorkItem)
- Art:** Nachricht
- Buttons:** Neu, Duplizieren, Vorschau...
- Navigation:** Beschreibung (selected), Nachrichtensystem, Historie
- Referenzobjekt:** (empty dropdown)
- Priorität:** Normal
- Empfangsbestätigung:** (checkbox, unchecked)
- An:** [Assignee.E-Mail]
- Cc:** (empty)
- Bcc:** (empty)
- Betreff:** (empty)
- Text:** Warnung: Bewilligung für Anforderung
- Buttons:** Schließen

## Aufforderung zur finanziellen Bewilligung an den Finanzprüfer des Unternehmens

Diese Aktion kommt im Rahmen der funktionellen Bewilligung zur Anwendung. Sie benachrichtigt den für die funktionelle Bewilligung zuständigen Mitarbeiter.

The screenshot shows a dialog box titled "Detail der Aktion 'Erinnerung zur Anforderung der finanziellen Bewilligung'". It contains the following fields and controls:

- Name:** Erinnerung zur Anforderung der finanziellen Bewilligung
- Kontext:** Workflow-Aufgaben (amWfWorkItem)
- Art:** Nachricht
- Buttons:** Vorschau...
- Navigation:** Beschreibung (selected), Nachrichtensystem, Historie
- Referenzobjekt:** (empty dropdown)
- Priorität:** Normal
- Empfangsbestätigung:** (checkbox, unchecked)
- An:** [Assignee.E-Mail]
- Cc:** (empty)
- Bcc:** (empty)
- Betreff:** (empty)
- Text:** Warnung: Bewilligung für Anforderung

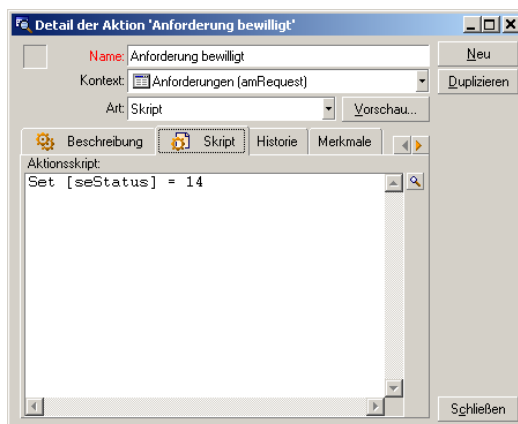
## Bewilligung der Beschaffungsanforderung

Diese Aktion kommt in der Aktivität **Anforderung bewilligt** zur Anwendung. Über die Aktion wird der Wert im Feld **Anforderungsstatus** (SQL-Name: seStatus) im Anforderungsdetail in den Wert **Bewilligt** geändert. Dabei handelt es sich um eine Aktion vom Typ **Skript**.

Das Feld **Anforderungsstatus** ist eine Systemaufzählung. Zum Zugreifen auf die verschiedenen Werte können Sie die Direkthilfe anzeigen:

- 1 Setzen Sie den Cursor im Anforderungsdetail auf das Feld **Anforderungsstatus**, und rechtsklicken Sie.
- 2 Wählen Sie im Kontextmenü die Option **Quickinfo**: Der angezeigte Wert **Bewilligt** wird in der Datenbank als **14** gespeichert.

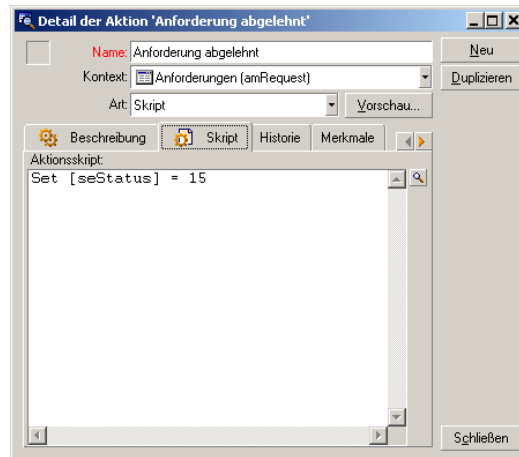
Die Aktion erscheint in folgendem Format:



## Ablehnung der Beschaffungsanforderung

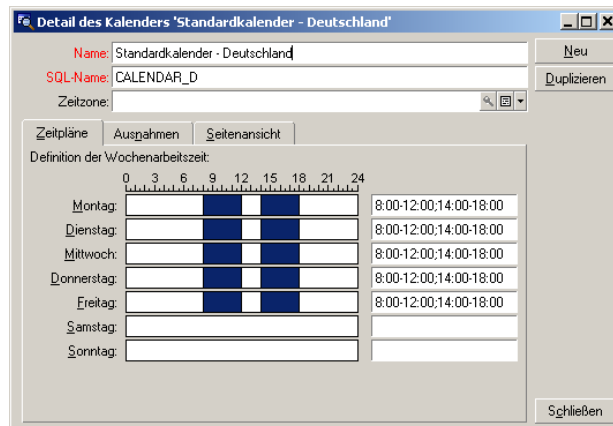
Diese Aktion kommt in der Aktivität **Anforderung abgelehnt** zur Anwendung, die an späterer Stelle erstellt wird.

Diese Aktion ähnelt der Aktion **Anforderung bewilligt**, wobei jedoch im Feld **Anforderungsstatus** (SQL-Name: seStatus) auf der Registerkarte **Allgemein** im Anforderungsdetail der Wert **Abgelehnt** erscheint.



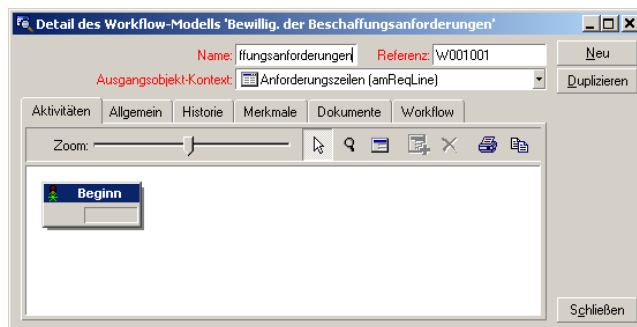
## Kalender erstellen

Der Zugriff auf die Liste der Kalender erfolgt über das Menü **Extras/ Kalender**. Der Kalender wird den Aktivitäten des Workflow-Modells zugeordnet. Über ihn lassen sich die Fälligkeiten für die Instanzen des Workflow-Modells festhalten:



## Workflow-Modell vorbereiten

- 1 Wählen Sie das Menü **Extras/ Workflow/ Workflow-Modelle**.
- 2 Klicken Sie auf **Neu**.
- 3 Nennen Sie den Workflow **Anforderungsbewilligung**.
- 4 Geben Sie den Kontext des Ausgangsobjekts ein. Dieser Kontext gilt standardmäßig für alle Aktivitäten, aus denen sich das Workflow-Modell zusammensetzt. In unserem Beispiel handelt es sich um die Tabelle der Anforderungen (SQL-Name: amReqLine).
- 5 Klicken Sie auf **Erstellen**: AssetCenter erstellt die Ausgangsaktivität **Beginn** im Grafikeditor auf der Registerkarte **Aktivitäten**.



## Aktivitäten erstellen

Die grafische Erstellung von Aktivitäten erfolgt auf der Registerkarte **Aktivitäten** im Workflow-Modelldetail:

- 1 Zeigen Sie mit dem Mauszeiger auf eine Stelle außerhalb der Workflow-Objekte.
- 2 Rechtsklicken Sie.
- 3 Wählen Sie in dem daraufhin erscheinenden Kontextmenü die Option **Aktivität hinzufügen**, um den Detailbildschirm der Aktivität anzuzeigen.

## Die Aktivität Funktionelle Bewilligung erstellen

- 1 Geben Sie als Aktivitätsnamen **Funktionelle Bewilligung** ein.

- 2 Die Aktivität fordert den Leiter der Abteilung, in der der Anforderer tätig ist, auf, die Anforderung zu bestätigen oder abzulehnen. Wählen Sie den Wert **Frage** in der Systemaufzählung des Felds **Art** (SQL-Name: seType).
- 3 Das Feld **Kontext** (SQL-Name: ContextTable) auf der Registerkarte **Allgemein** bleibt unverändert.

## Die Aktivität Funktionelle Bewilligung konfigurieren

- 1 Geben Sie die erforderlichen Informationen auf der Registerkarte **Parameter** ein, wie in der nachstehenden Abbildung zu sehen ist:

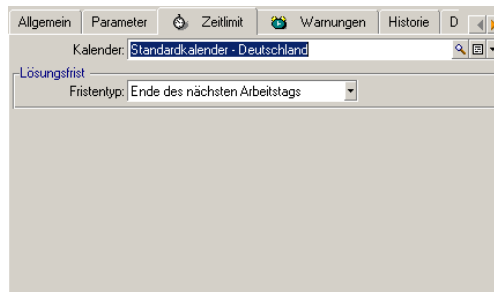
- 2 Legen Sie den genauen Wortlaut der Frage fest:
  - 1 Der Text der Frage bezieht sich auf die Nummer der Anforderung, die bewilligt werden soll.
  - 2 Sie haben die Wahl zwischen 2 möglichen Antworten: Ablehnung oder Bewilligung. Wenn Sie eine Unterregisterkarte mit der Antwort auf die Frage hinzufügen möchten, setzen Sie den Cursor auf den Unterregisterkartennamen, und drücken die rechte Maustaste, um das Kontextmenü anzuzeigen. Wählen Sie die Option **Verknüpfung hinzufügen** oder **Verknüpfung duplizieren**.
- 3 Geben Sie im Feld **Bearbeiter** (SQL-Name: Assignee) die Person an, die die Frage beantworten soll. In unserem Beispiel ist der Bearbeiter der Leiter der Abteilung, in der der Anforderer tätig ist. Bei dem Leiter handelt es sich um eine Person, die im vorherigen Schritt in der Tabelle der Workflow-Rollen erstellt wurde.
- 4 Wenn der Workflow-Bearbeiter automatisch über zu bewilligende Anforderungen informiert werden soll, gehen Sie wie folgt vor:

- 1 Wählen Sie rechts neben dem Feld **Person benachrichtigen** (SQL-Name: bNotifAssignee) den Wert **Ja**.
- 2 Legen Sie die auszuführende Aktion fest: In unserem Fall handelt es sich um die im vorherigen Schritt erstellte Aktion **Funktionelle Bewilligung**. Die Aktion wird automatisch ausgelöst, sobald eine Anforderung zur funktionellen Bewilligung vorgelegt wird.

## Ausführungsfrist für die Aktivität Funktionelle Bewilligung definieren

Auf der Registerkarte **Fristen** des Aktivitätendetails geben Sie folgende Informationen ein:

- 1 Legen Sie fest, welcher Arbeitstagekalender der Aktivität zugeordnet werden soll. Der Kalender wird bei der Berechnung der Workflow-Fälligkeiten herangezogen. Wählen Sie den Kalender **Standardkalender Deutschland**, den Sie an früherer Stelle erstellt haben.
- 2 Legen Sie die Frist für die Entscheidungsfindung fest, d. h. der Zeitpunkt, zu dem die Aktivität ausgelöst wird. In unserem Beispiel muss der Workflow-Bearbeiter die Frage vor Ende des nächsten Arbeitstages beantwortet haben.



## Warnung für die Aktivität Funktionelle Bewilligung definieren

Definieren Sie auf der Registerkarte **Warnungen** des Aktivitätendetails eine Warnmeldung für den Fall, dass vor dem Ablauf der auf der Registerkarte **Fristen** definierten **Frist** keine Entscheidung gefallen ist.

Um das Beispiel so einfach wie möglich zu gestalten, führt die Warnung zum Auslösen der Aktion **Funktionelle Bewilligungsanforderung**:

Wenn Sie im Kontextmenü die Option **Verknüpfung hinzufügen** wählen, können Sie zusätzliche Warnungen definieren.

## Ereignisse

Sobald Sie die Aktivität erstellt haben, generiert AssetCenter die Systemereignisse **Bewilligen** und **Ablehnen**, die den beiden auf die Frage möglichen Antworten entsprechen.



Wenn diese Ereignisse eintreten, werden sie von einem AssetCenter-Agenten in der Tabelle der grundlegenden Ereignisse von Workflow-Instanzen gespeichert (SQL-Name: amWfOccurEvent).

Die folgenden Aktivitäten werden je nach der Konfiguration der Workflow-Ereignisse entweder direkt von AssetCenter oder von AssetCenter Server ausgelöst:

- Wenn das Feld **Bearbeitung** (SQL-Name: seProcessingMode) die Werte **Ereignis speichern und unmittelbar bearbeiten** oder **Ereignis unmittelbar bearbeiten, nicht speichern** aufweist, löst AssetCenter die nächste Aktion aus.

- Weist das Feld **Bearbeitung** (SQL-Name: seProcessingMode) den Wert **Ereignis speichern und vom Server bearbeiten** auf, löst AssetCenter Server die nächste Aktion aus.

Standardmäßig wird im Feld **Bearbeitung** (SQL-Name: seProcessingMode) auf der Registerkarte **Allgemein** im Ereignisdetail der Wert **Ereignis speichern und vom Server bearbeiten** angezeigt.

## Die Aktivität **Finanzielle Bewilligung erstellen**

Diese Aktivität ähnelt der vorherigen.

Der Workflow-Bearbeiter ist jedoch eine andere Person: Es handelt sich um den Finanzprüfer des Unternehmens, Gerald Colombo. Diese Person wurde in einem früheren Schritt in der Tabelle der Workflow-Rollen erstellt. Wählen Sie die Aktion **Finanzielle Bewilligungsanforderung**, um ihn zu benachrichtigen.

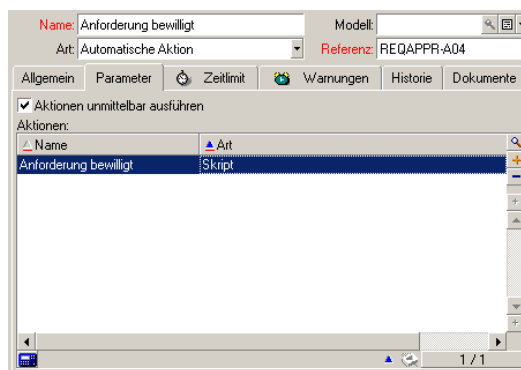
## Die Aktivität **Bewilligte Anforderung erstellen**

Wenn die Anforderung die ersten beiden Schritte erfolgreich durchlaufen hat, gilt sie als bewilligt.

Die Aktivität **Anforderung bewilligt** ist einer der Endpunkte des Workflow-Modells.

Diese Aktivität muss das Anforderungsdetail dahingehend ändern, dass die Anforderung als bewilligt erscheint.

Es handelt sich dabei um eine Aktion vom Typ **Automatische Aktion**. Die an früherer Stelle erstellte Aktion **Anforderung bewilligt** wird ausgeführt.

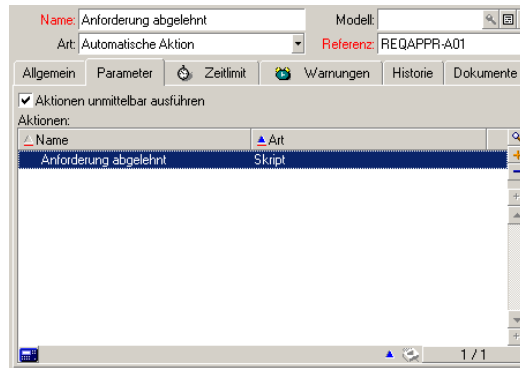




## Die Aktivität **Anforderung abgelehnt** erstellen

Die Aktivität **Anforderung abgelehnt** ähnelt der Aktivität **Anforderung bewilligt**.

In diesem Fall muss das Anforderungsdetail dahingehend geändert werden, dass es die Anforderung als abgelehnt ausweist. Bei der auszuführenden Aktion handelt es sich um die Aktion **Anforderung abgelehnt**, die an früherer Stelle erstellt wurde.



## Parameter für die mit den Aktivitäten erstellten Ereignisse festlegen

Gemeinsam mit den Aktivitäten wurden in den oben beschriebenen Schritten auch die folgenden Ereignisse erstellt:

- Aktivität **Funktionelle Bewilligung**:
  - Ereignis **Bewilligen**
  - Ereignis **Ablehnen**
- Aktivität **Finanzielle Bewilligung**:
  - Ereignis **Bewilligen**
  - Ereignis **Ablehnen**
- Aktivität **Anforderung bewilligt**:
  - Ereignis **Ausgeführt**
- Aktivität **Anforderung abgelehnt**:
  - Ereignis **Ausgeführt**

Wählen Sie die einzelnen Ereignisse nacheinander aus, und nehmen Sie Folgendes vor, um Abhängigkeiten von AssetCenter Server zu vermeiden:

- 1 Rechtsklicken Sie auf den Namen des zu bearbeitenden Ereignisses.
- 2 Wählen Sie das Menü **Ereignisdetail**.
- 3 Zeigen Sie die Registerkarte **Allgemein** an.
- 4 Wählen Sie für das Feld **Bearbeitung** (SQL-Name: seProcessingMode) den Wert **Ereignis speichern und unmittelbar bearbeiten**.
- 5 Klicken Sie auf **Ändern**.

## Ausgangsereignisse erstellen

Die Ereignisse, die eine Workflow-Instanz auslösen, sind der Aktivität **Beginn** zugeordnet.

Um ein Ausgangsereignis zu erstellen, rechtsklicken Sie auf die leere Ereigniszone der Aktivität **Beginn** und wählen die Option **Ereignisdetail**.

- 1 In unserem Beispiel wird die Workflow-Instanz ausgelöst, wenn im Feld **Anforderungsstatus** (SQL-Name: seStatus) im Anforderungsdetail der Wert **Zu bewilligen** erscheint.

Das Ausgangsereignis entspricht dem Typ **Bei Datenbankänderung** und die entsprechenden Aktivierungsparameter sind auf der Registerkarte **Allgemein** beschrieben, wie in der nachstehenden Abbildung zu sehen ist:

The screenshot shows a configuration window for an event. At the top, there are fields for 'Name' (Neue Anforderung), 'Referenz' (REQ\_TD\_VALIDATE), and 'Art' (Bei Datenbankänderung). Below these are tabs for 'Allgemein', 'Parameter', 'Historie', and 'Dokumente'. The 'Allgemein' tab is selected, showing a 'Kontext' of 'amRequest' and a 'Bearbeitung' dropdown menu set to 'Ereignis speichern und unmittelbar bearbeiten'. Under the 'Aktivierungsbedingungen' section, there is a checked checkbox for 'Workflow neu starten, wenn bereits ein anderer Workflow läuft' and an AQL condition 'seStatus = 13'.

- 2 Auf der Registerkarte **Allgemein** des Ereignisses wählen Sie im Feld **Bearbeitung** (SQL-Name: seProcessingMode) den Wert **Ereignis speichern und unmittelbar bearbeiten**.
- 3 Auf der Registerkarte **Parameter** des Ereignisses:

- 1 Markieren Sie das Kontrollkästchen **Aktualisierung** (SQL-Name: bUpdate).
- 2 Geben Sie in das Feld **Kontrollierte Felder** (SQL-Name: MonitFields) die Informationen aus dem Feld mit dem SQL-Namen **seStatus** ein.

## Übergänge erstellen

Sobald Sie die Aktivitäten definiert haben, müssen Sie sie nur noch durch Erstellen von Übergängen miteinander verknüpfen.

Gehen Sie zum Erstellen eines Übergangs wie folgt vor:

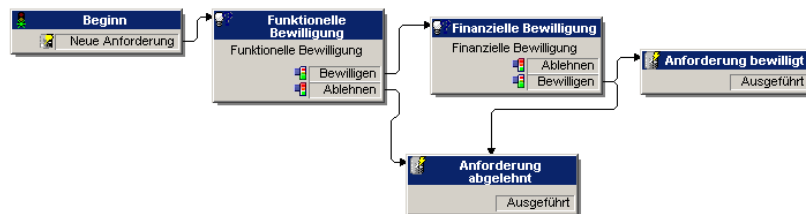
- 1 Klicken Sie mit der linken Maustaste auf das Ausgangsereignis des Übergangs.
- 2 Halten Sie die Maustaste niedergedrückt und ziehen Sie den Mauszeiger bis zur Zielaktivität.

In unserem Beispiel sind die nachstehenden Übergänge zu erstellen:

- Vom Ausgangsereignis des Workflows zur Aktivität **Funktionelle Bewilligung**.
- Vom Ereignis **Bewilligen** der Aktivität **Funktionelle Bewilligung** zur Aktivität **Finanzielle Bewilligung**.
- Vom Ereignis **Bewilligen** der Aktivität **Finanzielle Bewilligung** zur Aktivität **Anforderung bewilligt**.
- Von den Ereignissen **Ablehnen** der Aktivitäten **Funktionelle Bewilligung** und **Finanzielle Bewilligung** zur Aktivität **Anforderung abgelehnt**.

Daraufhin erhalten Sie das nachstehende Workflow-Modell:

**Abbildung 8.6. Workflow - Modell zur Bewilligung von Anforderungen**



## Beispiele zum Auslösen des Workflows

Jetzt müssen Sie nur noch das einwandfreie Funktionieren des erstellten Workflow-Modells prüfen.

Befolgen Sie dabei die nachstehenden Anweisungen:

- 1 Die Tabelle der Abteilungen und Personen ausfüllen [Seite 148]
- 2 Zu bewilligende Beschaffungsanforderungen erstellen [Seite 149]
- 3 Kontrolle des einwandfreien Ablaufs des Workflows [Seite 149]

### Die Tabelle der Abteilungen und Personen ausfüllen

Vor dem Erstellen einer zu bewilligenden Beschaffungsanforderung müssen Sie sowohl den Anforderer als auch den entsprechenden Abteilungsleiter in der Tabelle der Abteilungen und Personen erstellen. Der Abteilungsleiter muss über die zum Ausführen einer Reihe von Operationen erforderlichen Rechte verfügen:

- 1 Erstellen Sie den Anforderer **Jerome Carpenter**, der in der Abteilung **EDV-Leitung** arbeitet.
- 2 Ordnen Sie ihm das **Login Carpenter** sowie ein Kennwort und ein Benutzerprofil zu, so dass er Beschaffungsanforderungen eingeben kann, (Registerkarte **Profil** im entsprechenden Datensatzdetail). Sie können dazu auch das Profil **Anforderer** aus der Demodatenbank verwenden.
- 3 Geben Sie in das Feld **E-Mail** (SQL-Name: EMail) den Wert **AM:Carpenter** ein.
- 4 Der Leiter der Abteilung **EDV-Leitung** heißt **Philip Chavez**.
- 5 Um den Vorgang zu vereinfachen, räumen Sie Philip Chavez administrative Rechte an der Datenbank ein: Zeigen Sie hierzu die Registerkarte **Profil** im Detailbereich des entsprechenden Datensatzes an, und markieren Sie das Kontrollkästchen **Administrative Rechte** (SQL-Name: bAdminRight). Geben Sie das **Login** (SQL-Name: UserLogin) und das Kennwort von Philip Chavez ein.
- 6 Geben Sie in das Feld **E-Mail** (SQL-Name: EMail) den Wert **AM:Chavez** ein.
- 7 Wählen Sie die Person **Gerald Colombo**.
- 8 Geben Sie in das Feld **E-Mail** (SQL-Name: EMail) den Wert **AM:Colombo** und in das Feld **Login** (SQL-Name: UserLogin) den Wert **Colombo** ein.

## Zu bewilligende Beschaffungsanforderungen erstellen

Im nun folgenden Schritt geht es um die Bewilligung einer Beschaffungsanforderung:

- 1 Stellen Sie unter dem Login Wolfgang Heinemann eine Verbindung zur Demodatenbank her.
- 2 Wählen Sie das AssetCenter-Menü **Beschaffung/ Anforderungen**.
- 3 Klicken Sie auf **Neu**.
- 4 Markieren Sie die Option **Neue angepasste Anforderung erstellen**.
- 5 Klicken Sie auf **OK**.
- 6 Wählen Sie im Feld **Anforderer** (SQL-Name: Requester) auf der Registerkarte **Allgemein** im Anforderungsdetail den Datensatz **Carpenter**.
- 7 Wählen Sie den Wert **Zu bewilligen** für das Feld **Anforderungsstatus** (SQL-Name: seStatus) im Anforderungsdetail.
- 8 Bestätigen Sie die Erstellung der Beschaffungsanforderung. Damit ist das Workflow-Ausgangsereignis eingetreten. Es wird von einem AssetCenter-Agenten in der Tabelle der grundlegenden Ereignisse von Workflow-Instanzen (SQL-Name: amWfOccurEvent) gespeichert.

## Kontrolle des einwandfreien Ablaufs des Workflows

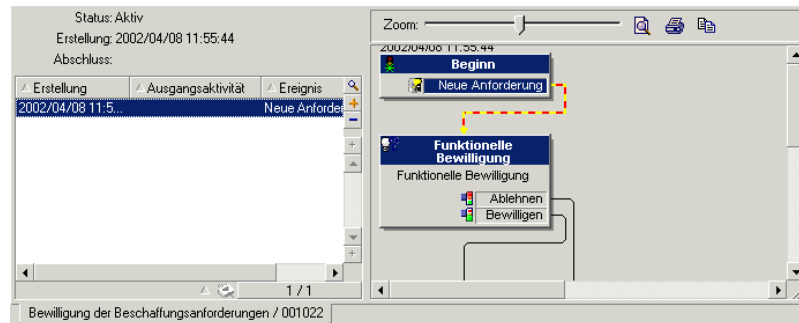
Zum Überprüfen des einwandfreien Ablaufs einer Workflow-Instanz müssen Sie AssetCenter starten und unter dem Login von Phillip Schumann, dem Abteilungsleiter von Wolfgang Heinemann, eine Verbindung zur Demodatenbank herstellen.

## Anzeige der Workflow-Instanz

Zeigen Sie das Detailfenster der Beschaffungsanforderung an, die Sie zuvor erstellt haben. Auf der Registerkarte **Workflow** werden die laufenden Workflow-Instanzen der Anforderung aufgeführt. Jede Workflow-Instanz wird in einer Unterregisterkarte beschrieben.

- In der linken Hälfte der Unterregisterkarte werden die Ereignisse aufgeführt, die eingetreten sind.

- In der rechten Hälfte erscheint der Status der Instanz. In unserem Beispiel blinkt die auszuführende Aufgabe **Funktionelle Bewilligung**.



## Anzeige der auszuführenden Aufgabe

- 1 Wählen Sie das Menü **Extras/ Laufende Aufgaben**. Der Abteilungsleiter Philip Chavez kann auf diese Weise das Detailfenster Aufgabe einsehen, die er ausführen soll:

Die Registerkarte **Allgemein** zeigt die Frage an, die Sie auf der Registerkarte **Fragen** der Aktivität **Funktionelle Bewilligung** definiert haben.

Die Registerkarte **Zuordnung** beschreibt die Person, die mit der Ausführung der Aufgabe betraut ist, sowie die Frist, innerhalb der die Aufgabe ausgeführt werden muss. Dieses Datum wird anhand der Informationen berechnet, die auf der Registerkarte **Fristen** im Aktivitätendetail **Funktionelle Bewilligung** enthalten sind, sowie des Datums, an dem die Aufgabe erstellt wurde (Übergangsaktivierungsdatum).

Sie können auf die Schaltfläche **Detail** klicken, um das Detailfenster der Anforderung zu öffnen, die Gegenstand der Aufgabe ist.

- 2 Klicken Sie auf die Schaltfläche **Bewilligen** oder **Ablehnen**, um die Aufgabe auszuführen. Es besteht die Möglichkeit, der auf der Registerkarte **Allgemein** getroffenen Entscheidung einen Kommentar hinzuzufügen.

## Kontext eines Workflow-Modells

Jede Workflow-Instanz läuft in einem spezifischen Kontext ab.

### Kontext eines Workflows definieren

Bei der Definition eines Workflow-Modells legen Sie den Kontext wie folgt fest:

- Ein Standardkontext in der oberen Hälfte des Workflow-Details.
- Ein Kontext für alle Ereignisse, Übergänge und Aktivitäten des Workflows (im Detailfenster eines Übergangs oder auf der Registerkarte **Allgemein** des Ereignis- oder Aktivitätendetails). Dieser Kontext ist mit dem Standardkontext des Workflows verknüpft.

In beiden Fällen handelt es sich bei dem Kontext um eine Tabelle.

### Referenzobjekt eines Workflow-Modells

Wenn ein Datensatz die Bedingungen für die Aktivierung eines Workflow-Modells erfüllt, wird eine Workflow-Instanz ausgelöst. Der Datensatz stellt ein Referenzobjekt für das Ausgangsereignis dar.

Beim Ablufen einer Workflow-Instanz macht das ursprüngliche Referenzobjekt eine Entwicklung im Vergleich zum ursprünglichen Referenzobjekt durch, und berücksichtigt dabei den Kontext, der für die Aktivitäten, Ereignisse und Übergänge definiert ist.

Ein Beispiel: Ein Workflow wird durch die Bewilligung einer Beschaffungsanforderung ausgelöst, und erstellt einen Auftrag unter Berücksichtigung dieser Anforderung. Wenn die Anforderung D1 bewilligt ist, stellt sie das Referenzobjekt für das Ausgangsereignis des Workflows dar. Anschließend wird das Ausgangsobjekt zum Auftrag C1, und damit zum Auftrag, der anhand der Beschaffungsanforderung generiert wurde.

## Aktuelle Workflow-Instanzen für ein Objekt einschränken

### Option Ein laufender Workflow pro Objekt (SQL-Name: bUniqueActive)

AssetCenter lässt Sie die aktuellen Workflow-Instanzen für ein Objekt mit Hilfe der Option **Ein laufender Workflow pro Objekt** auf der Registerkarte **Allgemein** eines Workflow-Details einschränken.

Wenn ein Ausgabereignis der Aktivität "Beginn" einen zweiten aktiven Workflow für ein Objekt erstellen muss, wird das Ereignis mit Hilfe der Optionen **Ein laufender Workflow pro Objekt** und **Workflow neu starten, wenn bereits ein anderer Workflow läuft** (SQL-Name: bReinitialize) (Registerkarte **Allgemein** des Ereignisdetails) festgelegt:

In der nachstehenden Tabelle finden Sie eine Übersicht über die verschiedenen Möglichkeiten:

		Option Ein laufender Workflow pro Objekt auf der Registerkarte Allgemein des Workflows.	
		Bestätigt	Nicht bestätigt
<b>Option Workflow neu starten, wenn bereits ein anderer</b>	<b>Bestätigt</b>	Wenn für das Objekt bereits ein aktueller Workflow vorliegt, wird er angehalten und ein neuer Workflow gestartet.	
<b>Workflow läuft auf der Registerkarte Allgemein des Ausgabereignisses der Aktivität Start.</b>	<b>Nicht bestätigt</b>	Wenn für das Objekt bereits ein aktueller Workflow vorliegt, wird das Ereignis ignoriert (kein neuer Workflow).	Ein neuer Workflow wird erstellt.

### Anwendungsbeispiele

Im Rahmen des Workflows zur Bewilligung von Beschaffungsanforderungen bieten sich folgende Möglichkeiten:

- Wählen Sie die Option **Ein laufender Workflow pro Objekt**, damit eine Beschaffungsanforderung nur Objekt eines einzigen Prozesses zur Bewilligung von Beschaffungsanforderungen ist.
- Wählen Sie die Option **Workflow neu starten, wenn bereits ein anderer Workflow läuft** für das Ereignis des Workflow-Ausgangs, um den Workflow neu zu starten, wenn sich die Zusammenstellung der Anforderung ändert.



## Workflow-Rollen

Die Aufgaben, die sich aus bestimmten Workflow-Aktivitäten ergeben, müssen von einem Bearbeiter durchgeführt werden.

 **Anmerkung:**

Die Bearbeiter kümmern sich nur um Aktivitäten vom Typ **Frage** oder **Benutzeraktion**. Es gibt keine Bearbeiter für Aktivitäten vom Typ **Automatische Aktion** oder **Test / Skript**.

Die Auswahl der Bearbeiter erfolgt in der Tabelle der Workflow-Rollen (SQL-Name amWfOrgRole). Der Zugriff auf diese erfolgt über das Menü Extras/ Workflow/ Rollen.

### Typ einer Workflow-Rolle


Bei den Workflow-Rollen sind verschiedene Typen zu unterscheiden (Feld **Art** (SQL-Name: seType) im Detailfenster einer Workflow-Rolle):

- **Benannte Person**
- **Berechnete Person**
- **Gruppe**

### Benannte Person

Der Bearbeiter wird direkt in der Tabelle der Abteilungen und Personen ausgewählt.

Beispiel:

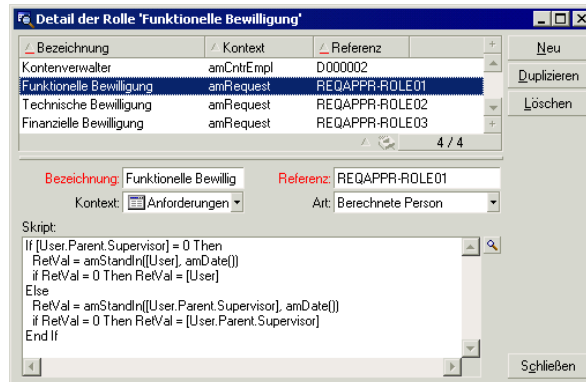


The screenshot shows a dialog box titled "Detail der Rolle 'Kontenverwalter'". It contains a table with two columns: "Bezeichnung" and "Kontext". The table has one row with "Kontenverwalter" in the "Bezeichnung" column and "amCntEmpl" in the "Kontext" column. Below the table, there are several fields: "Bezeichnung" (Kontenverwalter), "Referenz" (D000002), "Kontext" (Personen unter Vert), "Art" (Benannte Person), and "Bearbeiter" (Gerardin, Christian). On the right side, there are buttons for "Neu", "Duplizieren", and "Löschen". At the bottom right, there is a "Schließen" button.

## Berechnete Person

Der Bearbeiter erscheint in der Tabelle der Abteilungen und Personen, wird jedoch mit Hilfe eines Skripts ermittelt.

Beispiel:



## Gruppe

Der Bearbeiter wird in der Tabelle der Personengruppen ausgewählt (SQL-Name: "amEmplGroup").

## Bearbeiter definieren

Sie definieren den Bearbeiter einer Aktivität vom Typ **Frage** oder **Benutzeraktion** im Feld **Bearbeiter** (SQL-Name: Assignee) der Registerkarte **Fragen** (Aktivität vom Typ **Frage**) oder der Registerkarte **Aktionen** (Aktivität vom Typ **Benutzeraktion**).

## Workflow-Aktivitäten

Die Aktivitäten sind in zwei Gruppen unterteilt:

- Aktivitäten, für die der Eingriff eines AssetCenter-Benutzers erforderlich ist: Aktivitäten der "Art" (Feld **Art**) (SQL-Name: seType) **Frage** und **Benutzeraktion** in der oberen Hälfte des Aktivitätendetails.

- Aktivitäten, die automatisch ausgeführt werden: Aktivitäten vom Typ **Automatische Aktion** und **Test / Skript**

Der Wert des Felds **Art** einer Aktivität steuert die Anzeige bestimmter Registerkarten im Aktivitätendetail.

Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

- Aktivitäten vom Typ Frage
- Aktivität vom Typ Benutzeraktion
- Aktivitäten vom Typ Automatische Aktion
- Aktivitäten vom Typ Test / Skript
- Aktivität Beginn
- Aktivitätenmodelle
- Aktivitäten auslösen

## Aktivitäten vom Typ Frage

Bei diesen Aktivitäten ist der Eingriff eines Benutzers erforderlich, der in unserem Fall **Bearbeiter** (SQL-Name: Assignee) genannt wird.

Aktivitäten vom Typ **Frage** werden definiert wie folgt:

- Eine Frage oder Anweisungen
- Mögliche Antworten

Beispiele:

- Im Rahmen des Prozesses zur Bewilligung von Beschaffungsanforderungen, wird der Leiter einer Abteilung aufgefordert, die Beschaffungsanforderung der in seiner Abteilung tätigen Person zu bewilligen oder abzulehnen.
- Eine Aktivität vom Typ **Frage** kann auch der Kontrolle dienen, indem die Durchführung einer Aufgabe bestätigt und diese Bestätigung gespeichert wird. In diesem Fall gibt es beispielsweise nur eine einzige Antwort.

Wählen Sie den Typ **Frage** für eine Aktivität, die die Anzeige der Registerkarte **Fragen** steuert.

Geben Sie folgende Informationen ein:

- 1 Speichern Sie in der Tabelle der Workflow-Rollen den Bearbeiter, der mit der Beantwortung der Fragen und dem Durchführen der Anweisungen betraut ist. Dieser Bearbeiter wird über eine AssetCenter Aktion automatisch von der Notwendigkeit seines Eingreifens unterrichtet. Geben Sie dazu

einfach die entsprechenden Daten im Feld **Person benachrichtigen** (SQL-Name: bNotifAssignee) der Registerkarte **Fragen** ein.

---

 **Anmerkung:**

Die Aktion zur Benachrichtigung des Bearbeiters wird ausgelöst, sobald die auszuführende Aufgabe erstellt wird, d.h. bei Aktivierung des Übergangs, der zum Auslösen der Aktivität führt.

---

Der Zugriff auf das Detailfenster mit den auszuführenden Aufgaben erfolgt über das Menü **Extras/ Laufende Aufgaben**.

- 2 Es erscheint der Text mit der Frage bzw. den Anweisungen.
  - 3 Es erscheinen die möglichen Antworten. Jede einzelne Antwort ist in einer Unterregisterkarte beschrieben, und kann anhand ihrer Beschreibung und ihres SQL-Namens identifiziert werden. Zum Hinzufügen, Duplizieren oder Löschen einer Antwort klicken Sie mit der rechten Maustaste auf den Unterregisterkarten auf den Bereich mit den Namen, und wählen in dem daraufhin erscheinenden Kontextmenü die Option **Verknüpfung hinzufügen**, **Verknüpfung duplizieren** oder **Verknüpfung löschen**.
- 

 **Anmerkung:**

Jede Antwort führt automatisch zur Erstellung eines Ausgabeereignisses der Aktivität.

---

## Aktivität vom Typ Benutzeraktion

Bei diesen Aktivitäten ist der Eingriff eines Benutzers erforderlich, der in unserem Fall **Bearbeiter** (SQL-Name: Assignee) genannt wird.

Die Aktivitäten werden anhand der nachstehenden Elemente definiert:

- Anweisungen, die zu befolgen sind.
- Ein Assistent, der auszuführen ist.

Wählen Sie den Typ **Benutzeraktion** für eine Aktivität. Dieser Typ steuert die Anzeige der Registerkarte **Aktionen**.

Geben Sie folgende Informationen ein:

- Die Anweisungen, die zu befolgen sind
- Den Assistenten, der auszuführen ist

- Den Datensatz der Tabelle der Workflow-Rollen mit dem Bearbeiter, der für die Ausführung des Assistenten verantwortlich ist. Dieser Bearbeiter wird über eine AssetCenter-Aktion automatisch von der Notwendigkeit seines Eingreifens unterrichtet. Geben Sie dazu einfach die entsprechenden Daten im Feld **Person benachrichtigen** (SQL-Name: bNotifAssignee) der Registerkarte **Aktionen** ein.

---

 **Anmerkung:**

Die Aktion zur Benachrichtigung des Bearbeiters wird ausgelöst, sobald die auszuführende Aufgabe erstellt wird, d.h. bei Aktivierung des Übergangs der zum Auslösen der Aktivität führt.

---

Der Bearbeiter greift über das Menü **Extras/ Laufende Aufgaben** auf das Detailfenster mit den Aufgaben zu, die der ausführen soll.

---

 **Anmerkung:**

Als Ausgabeereignis einer Aktivität wird automatisch ein **ausgeführtes** Ereignis erstellt.

---

Beispiel: Im Rahmen der Verwaltung von Empfangsbestätigungen wird der Benutzer beim Durchführen des gesamten oder teilweisen Empfangs von Auftragszeilen unterstützt.

## Aktivitäten vom Typ Automatische Aktion

Diese Aktivitäten werden von AssetCenter oder AssetCenter Server automatisch durchgeführt.

### Beschreibung

Die Aktivitäten vom Typ **Automatische Aktion** führen automatisch die auszuführenden Aktionen durch.

Beispiel: Im Rahmen des Umzugs von Vermögensgegenständen ändert eine Aktivität vom Typ **Automatische Aktion** automatisch den Standort aller Vermögensgegenstände, für die der übergeordnete Gegenstand an einen anderen Ort gebracht wurde.

Wählen Sie den Typ **Automatische Aktion** für eine Aktivität, die die Anzeige der Registerkarte **Aktionen** steuert.

Geben Sie die Liste der auszuführenden Aktionen an.

---

 **Anmerkung:**

Als Ausgabeereignis einer Aktivität wird automatisch ein **ausgeführtes** Ereignis erstellt.

---

## Ausführung

Die Aktionen der Aktivität werden unmittelbar von der Workflow-Engine ausgeführt, die den entsprechenden Übergang aktiviert. Je nach dem Bearbeitungsmodus des Ereignisses, das den Übergang auslöst, werden die Aktionen entweder von AssetCenter Server oder von einem AssetCenter-Agenten ausgeführt.

- Wenn Sie das Kontrollkästchen **Aktionen unmittelbar ausführen** (SQL-Name: bExecImmediately) markieren, werden die Aktionen der Aktivität unmittelbar von der Workflow-Engine ausgeführt, die den entsprechenden Übergang aktiviert.
- Ansonsten werden die Aufgaben bei der nächsten Prüfung von AssetCenter Server ausgeführt.

## Aktivitäten vom Typ Test / Skript

Diese Aktivitäten werden von AssetCenter oder AssetCenter Server automatisch durchgeführt.

## Beschreibung

Diese Aktivitäten werden mit Hilfe eines Skripts und den daraufhin möglichen Vorgängen definiert.

Beispiel: Im Rahmen der Bestandsverwaltung und der Beschaffungsanforderungen stellt eine Aktivität vom Typ Test/ Skript für jede Anforderungszeile fest, ob das referenzierte Element lagerhaltig und nicht reserviert ist. In diesem Fall kann die Aktion eine Aktivität vom Typ **Frage** auslösen, bei der der Anforderer gefragt wird, ob er das lagerhaltige Element reservieren möchte oder nicht.

Wählen Sie den Typ **Test / Skript** für eine Aktivität, die die Anzeige der Registerkarte **Tests** steuert.

Geben Sie folgende Informationen ein:

- Das auszuführende Textskript.
- Mögliche Vorgänge. Die Beschreibungen der einzelnen Vorgänge befinden sich in einer Unterregisterkarte, wo sie mit Hilfe der Beschreibung und des SQL-Namens identifizierbar sind. Zum Hinzufügen, Duplizieren oder Löschen eines Ausgabevorgangs klicken Sie mit der rechten Maustaste in den Unterregisterkarten auf den Bereich mit den Namen, und wählen in dem daraufhin erscheinenden Kontextmenü die Option **Verknüpfung hinzufügen**, **Verknüpfung duplizieren** oder **Verknüpfung löschen**.



Anmerkung:

Der SQL-Name eines jeden Vorgangs muss mit dem Rückgabewert des Textskripts übereinstimmen.



Anmerkung:

Jeder Vorgang führt automatisch zur Erstellung eines Ausgabeereignisses der Aktivität.

## Ausführung

Die Aktionen der Aktivität werden unmittelbar von der Workflow-Engine ausgeführt, die den entsprechenden Übergang aktiviert. Je nach dem Bearbeitungsmodus des Ereignisses, das den Übergang auslöst, werden die Aktionen entweder von AssetCenter Server oder von einem AssetCenter-Agenten ausgeführt.

- Wenn Sie das Kontrollkästchen "Aktionen unmittelbar ausführen" (SQL-Name: bExecImmediately) markieren, werden die Aktionen der Aktivität unmittelbar von der Workflow-Engine ausgeführt, die den entsprechenden Übergang aktiviert. Je nach dem Bearbeitungsmodus des Ereignisses, das den Übergang auslöst, werden die Aktionen entweder von AssetCenter Server oder von einem AssetCenter-Agenten ausgeführt.
- Ansonsten werden die Aufgaben bei der nächsten Prüfung von AssetCenter Server ausgeführt.

## Aktivität Beginn

Bei der Aktivität **Beginn** handelt es sich um die Ausgangsaktivität eines Workflows.

Diese Aktivität ist obligatorisch und wird bei der Erstellung eines Workflows automatisch eingerichtet. Die Bearbeitung des Details ist nicht möglich.

Die anschließend auszuführende Aufgabe wird von dieser Aktivität nicht definiert.

Die Ausgabeereignisse der Aktivität **Beginn** lösen den Ablauf des Workflow-Modells aus.

## Aktivitätenmodelle

Die Aktivitätenmodelle erleichtern das Erstellen der Aktivitäten für Workflow-Modelle.

Die Speicherung dieser Modelle erfolgt in der Tabelle der Aktivitäten mit dem SQL-Namen "amWfActivity".

Der Zugriff auf die Liste der Aktivitätenmodelle erfolgt über das Menü Extras/ Workflow/ Aktivitätenmodelle.

---



### Anmerkung:

Wenn die Informationen, die im Detailfenster eines Aktivitätenmodells enthalten sind (Typ der Aktivität usw.) automatisch auf der Ebene des Aktivitätendetails kopiert werden sollen, das sich auf dieses Modell bezieht (Feld **Modell** (SQL-Name: Template) im Aktivitätendetail), muss ein AssetCenter Administrator die geeigneten Standardwerte für die Felder und Verknüpfungen eines Aktivitätendetails definiert haben.

---

## Aktivitäten auslösen

Zum Auslösen einer Aktivität muss die Eingabebedingung erfüllt sein, die im Feld **Eingabebedingung** (SQL-Name: seInCond) der Registerkarte **Allgemein** im Detailfenster der Aktivität erscheint. Diese Bedingung bezieht sich auf die Übergänge, die die Aktivität auslösen.

- Wenn die Aktivität von einem einzelnen Übergang ausgelöst wird, braucht der Übergang nur von AssetCenter oder AssetCenter Server aktiviert zu werden, um die Aktivität auszulösen.
- Wird die Aktivität jedoch von mehreren Übergängen ausgelöst, geschieht folgendes:



- Wenn es sich bei der Eingabebedingung für die Aktivität um den Operatoren **AND** handelt, müssen alle Übergänge aktiviert werden, um die Aktivität auszulösen.
- Wenn es sich bei der Eingabebedingung für die Aktivität um den Operatoren **OR** handelt, muss nur einer der Übergänge aktiviert werden, um die Aktivität auszulösen.

---

 **Anmerkung:**

Wenn die Eingabebedingungen einer Aktivität besonders komplex sind (Kombinationen von **AND** und **OR**), können Sie eine Verkettung der Sonderaktivitäten vom Typ **Test / Skript** durchführen.

---

## Aufgaben

In diesem Abschnitt wird das Erstellen und Ausführen von Workflow-Aufgaben erklärt:

### Aufgaben erstellen

Wenn ein Übergang aktiviert wird, der zum Auslösen einer Aktivität führt, wird von der Workflow-Engine, die den Übergang aktiviert hat, automatisch eine auszuführende Aufgabe erstellt.

Je nach der Option, die Sie im Feld **Aufgabe speichern** (SQL-Name: bLogWorkItem) der Registerkarte **Allgemein** einer Aktivität gewählt haben, wird diese Aufgabe in der Tabelle der Workflow-Aufgaben (SQL-Name: amWfWorkItem) gespeichert oder nicht.

Die Option **Aufgabe speichern** wird automatisch gewählt:

- Wenn es sich bei der Aktivität um den Typ **Frage** oder **Benutzeraktion** handelt
- Wenn es sich bei der Aktivität um den Typ **Automatische Aktion** oder **Test / Skript** handelt, und wenn die Option **Aktionen unmittelbar ausführen** (SQL-Name: bExecImmediately) im Detailfenster der Aktivität nicht markiert wurde

---

 **WARNUNG:**

Wenn eine Aufgabe nicht gespeichert wurde, ist die Erstellung von Workflow-Warnungen für diese Aufgabe nicht möglich: Die Registerkarten **Fristen** und **Warnungen** werden im Detailfenster der Aktivität nur angezeigt, wenn die Option **Aufgabe speichern** nicht bewilligt wird.

---

Je nachdem, ob das Eingreifen des Benutzers erforderlich ist oder nicht, kommt es zu einer entsprechenden Durchführung der Aufgabe.

## Aktivität vom Typ Automatische Aktion oder Test / Skript

Wenn die Aufgabe aus einer Aktivität vom Typ **Automatische Aktion** oder **Test / Skript** hervorgegangen ist, für die die Option **Aktionen unmittelbar ausführen** (SQL-Name: bExecImmediately) markiert wurde, wird die Aufgabe von der Workflow-Engine durchgeführt, die den Übergang aktiviert hat, der zur Erstellung der Aufgabe geführt hat. Dabei kann es sich um den AssetCenter Server oder einen AssetCenter-Agenten handeln.

Ansonsten prüft AssetCenter Server in regelmäßigen Abständen, ob Workflow-Aufgaben vorliegen, und führt sie aus.

Die Häufigkeit, mit der AssetCenter Server den Workflow überwacht, wird in den AssetCenter Server-Optionen festgelegt.

## Aktuelle Aufgabenliste anzeigen

Das Menü Extras/ Laufende Aufgaben ermöglicht die Anzeige der Liste mit den auszuführenden Aufgaben.

Für jede Person, die eine Verbindung zur Datenbank herstellt, erscheint eine spezifische Aufgabenliste:

- Ein AssetCenter-Administrator kann die Liste der aktuellen Aufgaben für sämtliche Workflow-Instanzen einsehen.
- Ein Bearbeiter sieht folgendes:
  - Die Aufgaben, die er ausführen muss.
  - Die Aufgaben, die den Gruppen zugeordnet sind, zu denen er gehört, ohne dass sie einem bestimmten Bearbeiter zugeordnet wären.

Als Administrator können Sie auch über das Detailfenster eines Workflows auf die Liste der aktuellen Aufgaben für eine Aktivität zugreifen. Gehen Sie dazu wie folgt vor:

- 1 Setzen Sie den Cursor auf eine Aktivität. Drücken Sie die rechte Maustaste.
- 2 Wählen Sie in dem daraufhin erscheinenden Kontextmenü die Option **Laufende Aufgaben**.



**Anmerkung:**

Bei der angezeigten Liste handelt es sich nicht um die vollständige Liste der Workflow-Aufgaben mit dem SQL-Namen "amWfWorkItem", sondern nur um die Aufgaben, die ausgeführt werden sollen.

## Benutzeraufgaben ausführen

Ein Bearbeiter greift über das Menü Extras/ Laufende Aufgaben auf die Aufgaben zu, die er ausführen soll.



**WARNUNG:**

Wenn es sich bei dem Benutzer, der die Verbindung zur Datenbank hergestellt hat, um einen AssetCenter-Administrator handelt, werden über das Menü **Extras/ Laufende Aufgaben** alle auszuführenden Aufgaben angezeigt. Wenn es sich bei dem Benutzer, der die Verbindung zur Datenbank hergestellt hat, nicht um einen AssetCenter-Administrator handelt, werden über das Menü **Extras/ Laufende Aufgaben** nur die Aufgaben angezeigt, die dem entsprechenden Benutzer bzw. den Gruppe zugeordnet sind, zu denen der Benutzer gehört.

Klicken Sie auf die Schaltfläche **Detail**, um auf das Detail des Objekts zuzugreifen, auf das sich die Aufgabe bezieht.

Zum Ausführen einer Aufgabe öffnen Sie im Aufgabendetail die Registerkarte **Allgemein** an:

- Wenn es sich bei der Aktivität, die zur Generierung der Aufgabe geführt hat, um den Typ **Frage** handelt, erscheinen auf der Registerkarte **Allgemein** die Frage bzw. die zu befolgenden Anweisungen. Die möglichen Antworten erscheinen in Form von Schaltflächen. Klicken Sie auf die geeignete Schaltfläche. Zusätzlich haben Sie die Möglichkeit, einen Kommentar einzugeben.

- Wenn es sich bei der Aktivität um den Typ **Benutzeraktion** handelt, brauchen Sie nur auf die Schaltfläche **Assistent** zu klicken, um den Assistenten zu starten.

## Benutzeraufgaben zuordnen

Die Informationen zur Zuordnung einer Benutzeraufgabe erscheinen auf der Registerkarte **Zuordnung** des Aufgabendetails.

Sie können die Zuordnung einer Benutzeraufgabe ändern, sofern Sie über die entsprechenden Berechtigungen verfügen:

- Wert des Felds **Zuordnung** (SQL-Name: seAssignment)
- Bearbeiter der Aufgabe

## Workflow-Aufgaben verwalten

Die Informationen bezüglich einer Workflow-Aufgabe erscheinen auf der Registerkarte **Verwaltung** des Aufgabendetails.

Das Anzeigen dieser Informationen ist jedoch nur für einen AssetCenter-Administrator möglich.

## Ereignisse

Die Ereignisse sind Aktivitäten zugeordnet, die wiederum Übergänge auslösen, die zu anderen Aktivitäten führen.

Auf der Ebene einer Aktivität sind drei Typen von Systemereignissen zu unterscheiden. Der Typ des Systemereignisses wird über das Feld **Systemtyp** (SQL-Name: seType) im Ereignisdetail definiert:

- Ereignis vom Typ **System**
- Ereignis vom Typ **Benutzer**
- Ereignis vom Typ **Warnung**

## Ereignisse vom Typ System

Die Ereignisse vom Typ **System** werden bei der Erstellung bzw. Änderung von Aktivitäten automatisch von AssetCenter definiert

Die Ereignisse entsprechen den verschiedenen Arbeitsausgaben, die im Rahmen der Aktivität ausgeführt wurden:

- Antworten auf eine Aktivität vom Typ **Frage**.
- Ergebnisse für eine Aktivität vom Typ **Test / Skript**.
- Ereignis, das im Rahmen einer Aktivität vom Typ **Benutzeraktion** oder **Automatische Aktion** ausgeführt wird.

Beispiel: Wenn eine Aktivität eine Frage stellt, die mit "Ja" und "Nein" beantwortet werden kann, werden zwei Systemereignisse auf der Ebene der Aktivität mit den Namen "Ja" und "Nein" erstellt.

## Ereignisse vom Typ Warnung

Die Ereignisse vom Typ Warnung werden beim Definieren der Warnungen in Zusammenhang mit Aktivitäten erstellt, die Ereignisse auslösen.

Die Definition einer solchen **Warnung** erfolgt auf der Registerkarte Warnungen des Aktivitätendetails. Das Ereignis trägt den Namen der Warnung.

## Ereignisse vom Typ Benutzer

Die Benutzerereignisse sind unabhängig von der Aufgabe, die im Rahmen der Aktivität ausgeführt wird. Die Erstellung der Benutzerereignisse erfolgt manuell mit Hilfe des Grafikeditors über das Kontextmenü und die Option **Ereignis hinzufügen**.



Anmerkung:

Bei den Ereignissen, die der Aktivität **Beginn** zugeordnet sind, handelt es sich um Benutzerereignisse.

---

Bei den Ereignissen vom Typ **Benutzer** sind zwei Arten zu unterscheiden (Feld **Art** (SQL-Name: seMonitoringType) in der oberen Hälfte des Ereignisdetails):

- **Bei Datenbankänderung**
- **In regelmäßigen Abständen**

## Ereignis vom Typ Bei Datenbankänderung

Mit Ereignissen vom Typ **Bei Datenbankänderung** lassen sich Workflow-Instanzen mit bestimmten Datensätzen auslösen.

Ein Ereignis vom Typ **Bei Datenbankänderung** tritt ein:

- Wenn die allgemeinen Aktivierungsbedingungen, die auf der Registerkarte **Allgemein** stehen, erfüllt sind.
- Wenn bestimmte Auslöseparameter auf der Ebene der überwachten Datensätze überprüft worden sind.

Die Beschreibung der Parameter, die ein Ereignis vom Typ **Bei Datenbankänderung** auslösen, befindet sich auf der Registerkarte **Parameter** des Ereignisdetails. Dazu gehören folgende Informationen:

- Die zu überwachenden Datensätze. (Dabei kann es sich um die Datensätzen der im Kontext angegebenen Tabelle oder um verknüpfte Datensätze handeln). Sind diese Datensätze mit der im Kontext angegebenen Tabelle verknüpft, geben Sie die entsprechende Verknüpfung im Feld **Verknüpfung / Kontext** (SQL-Name: LinkToMonitTable) ein.
- Die Aktivierungsbedingungen des Ereignisses für die überwachten Datensätze. Gehen Sie zum Definieren der Aktivierungsbedingungen wie folgt vor:
  - Markieren Sie eines oder mehrere der nachstehenden Kontrollkästchen: **Einfügen** (SQL-Name: bInsert), **Aktualisierung** (SQL-Name: bUpdate), **Löschen** (SQL-Name: bDelete).

Wenn Sie das Kontrollkästchen **Einfügen** markieren, werden die erstellten Datensätze berücksichtigt.

Wenn Sie das Kontrollkästchen **Aktualisierung** markieren, können Sie die Felder angeben, für die die Änderung im Feld **Kontrollierte Felder** (SQL-Name: MonitFields) berücksichtigt werden soll. Wenn Sie die Namen mehrerer Felder angeben wollen, trennen Sie die einzelnen Namen mit einem Komma. Wenn Sie das Feld leer lassen, wird keines der geänderten Felder berücksichtigt.

Wenn Sie das Kontrollkästchen **Löschen** markieren, werden die gelöschten Datensätze berücksichtigt.

---

 **WARNUNG:**

Die Aktivierungsbedingung des Ereignisses kann nicht dem Löschen des Referenzobjekts durch den Kontext entsprechen.

- 
- Schreiben eines Skripts im Bereich **Skript** (SQL-Name: memScript). Wenn Sie ein Skript schreiben und eines oder mehrere der Kontrollkästchen **Einfügen**, **Aktualisierung** und **Löschen** markieren,

kommt es zu einer Einschränkung der Aktivierungsbedingungen durch das Skript.

Beispiel: Wenn ein Ereignis durch die Änderung des Gesamtpreises für eine bereits vorliegende Anforderung ausgelöst werden soll, müssen Sie die Registerkarte **Parameter** wie folgt ausfüllen:

**Abbildung 8.7. Registerkarte Parameter für ein Ereignis vom Typ Bei Datenbankänderung**

The screenshot shows a configuration window with four tabs: 'Allgemein', 'Parameter', 'Historie', and 'Dokumente'. The 'Parameter' tab is active. It contains the following fields and options:

- Zu kontrollierende Datensätze:**
  - Verknüpfung / Kontext: (dropdown menu)
  - Kontrollierte Tabelle: amRequest
- Aktivierungsbedingungen für kontrollierte Datensätze:**
  - Einfügen
  - Aktualisierung
  - Löschen
- Kontrollierte Felder:** mEstimCost
- Skript:** (empty text area)

Sobald ein Ereignis vom Typ **Bei Datenbankänderung** eintritt, wird es von dem AssetCenter-Client-Rechner berücksichtigt, auf dem es zu diesem Ereignis gekommen ist. Die Art der Bearbeitung des Ereignisses ist abhängig von der Option, die Sie im Feld **Bearbeitung** (SQL-Name: seProcessingMode) der Registerkarte **Allgemein** des Ereignisdetails ausgewählt haben.

Weitere Informationen zu diesem Thema finden Sie unter "Bearbeitung der Ereignisse", im Kapitel "Workflow" im vorliegenden Handbuch.

## Ereignis vom Typ In regelmäßigen Abständen

Die Ereignisse vom Typ **In regelmäßigen Abständen** betreffen eine Auswahl von Datensätzen einer bestimmten Tabelle und ermöglichen das regelmäßige Auslösen eines Workflows für jeden einzelnen Datensatz der Auswahl.

Beispiel: Jeden Monat werden die Restwerte der Vermögensgegenstände der Art "PC" aktualisiert.

Ein Ereignis vom Typ **In regelmäßigen Abständen** tritt ein, wenn die auf der Registerkarte **Allgemein** angegebenen Aktivierungsbedingungen bestätigt werden.

In diesem Fall löst AssetCenter Server das Ereignis aus.

Die Häufigkeit, mit der AssetCenter Server die Ereignisse vom Typ **In regelmäßigen Abständen** auslöst, wird von der für die Planung zuständigen Person im Ereignisdetail auf der Registerkarte **Parameter** festgelegt.

Eine Beschreibung der anschließenden Verarbeitung des Ereignisses finden Sie im Abschnitt "Bearbeitung der Ereignisse" im Kapitel "Workflow" des vorliegenden Handbuchs.

## Allgemeine Bedingungen zur Aktivierung von Ereignissen

Unabhängig vom Typ des Ereignisses können Sie auf der Registerkarte **Allgemein** die Bedingungen zur Aktivierung von Ereignissen definieren:

### AQL-Bedingung (SQL-Name: AQLCond)

Das Feld **AQL-Bedingung** legt die Auswahl der Datensätze fest, die im Workflow-Modell berücksichtigt werden sollen.

### Workflow neu starten, wenn bereits ein anderer Workflow läuft (SQL-Name: bReinitialize)

---



Anmerkung:

Die Option **Workflow neu starten, wenn bereits ein anderer Workflow läuft** wird im Detailfenster des Ereignisses angezeigt, das aus der Aktivität "Beginn" hervorgegangen ist.

Über das Kontrollkästchen **Workflow neu starten, wenn bereits ein anderer Workflow läuft** legen Sie den Ablauf des Workflows fest, wenn sich ein Ausgabeereignis der Aktivität **Beginn** auf ein Referenzobjekt der Datenbank bezieht, das bereits Objekt einer Instanz desselben Workflows ist.

Der weitere Verlauf des Workflows ist jedoch nicht nur von dieser Option, sondern auch von der Option **Ein laufender Workflow pro Objekt** (SQL-Name: bUniqueActive) auf der Registerkarte **Allgemein** des Workflows abhängig.

In der nachstehenden Tabelle finden Sie eine Übersicht über die verschiedenen Möglichkeiten:



**Tabelle 8.1. Beschränkungen für Workflow-Instanzen**

		<b>Option Ein laufender Workflow pro Objekt auf der Registerkarte Allgemein des Workflows.</b>	
		<b>Bestätigt</b>	<b>Nicht bestätigt</b>
<b>Option Workflow neu starten, wenn bereits ein anderer Workflow läuft auf der Registerkarte Allgemein des Ausgabereignisses der Aktivität Start.</b>	<b>Bestätigt</b>	Wenn für das Objekt bereits ein aktueller Workflow vorliegt, wird er angehalten und ein neuer Workflow gestartet.	
	<b>Nicht bestätigt</b>	Wenn für das Objekt bereits ein aktueller Workflow vorliegt, wird das Ereignis ignoriert (kein neuer Workflow).	Ein neuer Workflow wird erstellt.

## Bearbeitung der Ereignisse

Sobald die allgemeinen Aktivierungsbedingungen erfüllt sind, ergibt sich die Vorgehensweise bei der Bearbeitung der Ereignisse aus folgenden Elementen:

- "Typ" des Ereignisses (Feld **Art** (SQL-Name: seMonitoringType) in der oberen Hälfte des Ereignisdetails).
- Option, die im Feld **Bearbeitung** (SQL-Name: seProcessingMode) der Registerkarte **Allgemein** des Ereignisdetails gewählt wurde.

In der nachstehenden Tabelle finden Sie die verschiedenen Vorgehensweisen bei der Verarbeitung eines Ereignisses:

Tabelle 8.2. Möglichkeiten zur Verarbeitung von Ereignissen

Ereignis vom Typ In regelmäßigen Abständen	Ereignis speichern und vom Server bearbeiten	Ereignis speichern und unmittelbar bearbeiten	Ereignis unmittelbar bearbeiten, nicht speichern
<p>AssetCenter Server löst das Ereignis aus, wenn die Aktivierungsbedingungen erfüllt sind. Die Definition der Häufigkeit, mit der das Ereignis ausgelöst wird, erfolgt auf der Registerkarte <b>Parameter</b> des Ereignisdetails.</p> <p>Sobald das Ereignis eintritt, speichert AssetCenter Server das Ereignis in der Tabelle mit dem SQL-Namen <b>amWfOccurEvent</b>.</p> <p>Der Übergang wird anschließend von AssetCenter Server aktiviert. Die Häufigkeit, mit der AssetCenter Server die zu aktivierenden Übergänge prüft, wird auf der Ebene der Optionen von AssetCenter Server definiert.</p>	<p>AssetCenter Server löst das Ereignis aus, wenn die Aktivierungsbedingungen erfüllt sind. Die Definition der Häufigkeit, mit der das Ereignis ausgelöst wird, erfolgt auf der Registerkarte <b>Parameter</b> des Ereignisdetails.</p> <p>Sobald das Ereignis eintritt, speichert AssetCenter Server das Ereignis in der Tabelle mit dem SQL-Namen <b>amWfOccurEvent</b>.</p> <p>Der Übergang wird unmittelbar von AssetCenter Server aktiviert.</p>	<p>AssetCenter Server löst das Ereignis aus, wenn die Aktivierungsbedingungen erfüllt sind. Die Definition der Häufigkeit, mit der das Ereignis ausgelöst wird, erfolgt auf der Registerkarte <b>Parameter</b> des Ereignisdetails.</p> <p>Wenn das Ereignis eintritt, wird es nicht in der Tabelle mit dem SQL-Namen <b>amWfOccurEvent</b> gespeichert. Der Übergang wird jedoch unmittelbar von AssetCenter Server ausgeführt.</p>	

	<b>Ereignis speichern und vom Server bearbeiten</b>	<b>Ereignis speichern und unmittelbar bearbeiten</b>	<b>Ereignis unmittelbar bearbeiten, nicht speichern</b>
<b>Ereignis vom Typ Bei Datenbankänderung oder Systemereignis, das von AssetCenter ausgelöst wird (Ergebnis einer Aktivität vom Typ Frage oder Benutzeraktion, Ergebnis einer Aktivität vom Typ Automatische Aktion oder Test / Skript, die von AssetCenter ausgeführt wird.</b>	<p>Sobald ein Ereignis eintritt, wird es vom AssetCenter-Client in der Tabelle mit dem SQL-Namen <b>amWfOccurEvent</b> gespeichert.</p> <p>Der Übergang wird anschließend von AssetCenter Server aktiviert. Die Häufigkeit, mit der AssetCenter Server die zu aktivierenden Übergänge prüft, wird auf der Ebene der Optionen von AssetCenter Server definiert.</p>	<p>Sobald ein Ereignis eintritt, wird es vom AssetCenter-Client in der Tabelle mit dem SQL-Namen <b>amWfOccurEvent</b> gespeichert.</p> <p>Der Übergang wird unmittelbar vom AssetCenter-Client aktiviert.</p>	<p>Wenn ein Ereignis eintritt, wird es nicht in der Tabelle mit dem SQL-Namen <b>amWfOccurEvent</b> gespeichert. Der Übergang wird jedoch unmittelbar vom AssetCenter-Client aktiviert.</p>
<b>Von AssetCenter Server ausgelöstes Systemereignis (Ergebnis einer Aktivität vom Typ Test / Skript oder Automatische Aktion ausgeführt von AssetCenter Server) oder Ereignis mit einer Warnung</b>	<p>Sobald ein Ereignis eintritt, wird es von AssetCenter Server in der Tabelle mit dem SQL-Namen <b>amWfOccurEvent</b> gespeichert.</p> <p>Der Übergang wird anschließend von AssetCenter Server aktiviert. Die Häufigkeit, mit der AssetCenter Server die zu aktivierenden Übergänge prüft, wird auf der Ebene der Optionen von AssetCenter Server definiert.</p>	<p>Sobald ein Ereignis eintritt, wird es von AssetCenter Server in der Tabelle mit dem SQL-Namen <b>amWfOccurEvent</b> gespeichert.</p> <p>Der Übergang wird unmittelbar von AssetCenter Server aktiviert.</p>	<p>Wenn ein Ereignis eintritt, wird es nicht in der Tabelle mit dem SQL-Namen <b>amWfOccurEvent</b> gespeichert. Der Übergang wird jedoch unmittelbar von AssetCenter Server aktiviert.</p>

Tabelle 8.3. Möglichkeiten zur Verarbeitung von Ereignissen

	<b>Ereignis speichern und unmittelbar bearbeiten</b>	<b>Ereignis unmittelbar bearbeiten, nicht speichern</b>
<b>Ereignis vom Typ In regelmäßigen Abständen</b>	<p>AssetCenter Server löst das Ereignis aus, wenn die Aktivierungsbedingungen erfüllt sind. Die Definition der Häufigkeit, mit der das Ereignis ausgelöst wird, erfolgt auf der Registerkarte <b>Parameter</b> des Ereignisdetails.</p> <p>Sobald das Ereignis eintritt, speichert AssetCenter Server das Ereignis in der Tabelle mit dem SQL-Namen <b>amWfOccurEvent</b>.</p> <p>Der Übergang wird unmittelbar von AssetCenter Server aktiviert.</p>	<p>AssetCenter Server löst das Ereignis aus, wenn die Aktivierungsbedingungen erfüllt sind. Die Definition der Häufigkeit, mit der das Ereignis ausgelöst wird, erfolgt auf der Registerkarte <b>Parameter</b> des Ereignisdetails.</p> <p>Wenn das Ereignis eintritt, wird es nicht in der Tabelle mit dem SQL-Namen <b>amWfOccurEvent</b> gespeichert. Der Übergang wird jedoch unmittelbar von AssetCenter Server ausgeführt.</p>
<b>Ereignis vom Typ Bei Datenbankänderung oder Systemereignis, das von AssetCenter ausgelöst wird (Ergebnis einer Aktivität vom Typ Frage oder Benutzeraktion, Ergebnis einer Aktivität vom Typ Automatische Aktion oder Test / Skript, die von AssetCenter ausgeführt wird.</b>	<p>Sobald ein Ereignis eintritt, wird es vom AssetCenter-Client in der Tabelle mit dem SQL-Namen <b>amWfOccurEvent</b> gespeichert.</p> <p>Der Übergang wird unmittelbar vom AssetCenter-Client aktiviert.</p>	<p>Wenn ein Ereignis eintritt, wird es nicht in der Tabelle mit dem SQL-Namen <b>amWfOccurEvent</b> gespeichert. Der Übergang wird jedoch unmittelbar vom AssetCenter-Client aktiviert.</p>
<b>Von AssetCenter Server ausgelöstes Systemereignis (Ergebnis einer Aktivität vom Typ Test / Skript oder Automatische Aktion ausgeführt von AssetCenter Server) oder Ereignis mit einer Warnung</b>	<p>Sobald ein Ereignis eintritt, wird es von AssetCenter Server in der Tabelle mit dem SQL-Namen <b>amWfOccurEvent</b> gespeichert.</p> <p>Der Übergang wird unmittelbar von AssetCenter Server aktiviert.</p>	<p>Wenn das Ereignis eintritt, wird es nicht in der Tabelle mit dem SQL-Namen <b>amWfOccurEvent</b> gespeichert. Der Übergang wird jedoch unmittelbar von AssetCenter Server ausgeführt.</p>

Mit Hilfe dieser verschiedenen Verarbeitungsmethoden ist es möglich, den Ablauf einer Workflow-Instanz mit hoher Genauigkeit zu programmieren.

Dabei können Sie Auswahlen auf folgenden Ebenen treffen:

- Ereignistypen
- Ereignisverarbeitung
- Aktivitäten

Durch die Auswahlen, die Sie auf der Ebene der Ereignistypen, Verarbeitung der Ereignisse und der Aktivitäten treffen, können Sie synchrone, asynchrone und gemischte Workflow-Modelle ausführen.

## Anwendung - Ausführung eines synchronen Workflows

Zum Ausführen eines synchronen Workflows müssen Sie nachstehende Elemente definieren:

- Ereignisse vom Typ **Bei Datenbankänderung**, für die die Option **Ereignis speichern und unmittelbar bearbeiten** gewählt wurde (Feld **Bearbeitung** (SQL-Name: seProcessingMode) der Registerkarte **Allgemein** eines Ereignisdetails)
- Aktivitäten vom Typ **Automatische Aktion** oder **Test / Skript**, für die die Option **Aktionen unmittelbar ausführen** (SQL-Name: bExecImmediately) aktiviert wurde, und die von den zuvor definierten Ereignissen ausgelöst werden

Beispiel:

Im nachstehenden Workflow-Modell wird festgelegt, dass beim Standortwechsel des übergeordneten Vermögensgegenstands die untergeordneten Vermögensgegenstände automatisch an den gleichen Ort gebracht werden:

**Abbildung 8.8. Beispiel eines synchronen Workflows**



Wenn der Standort eines Vermögensgegenstands geändert wird und Sie auf die Schaltfläche **Ändern** klicken, geschieht Folgendes:

- 1 Ein Ereignis vom Typ Bei Datenbankänderung tritt ein.
- 2 Der Standort des Vermögensgegenstands wird geändert.
- 3 Der Workflow wird gestartet.
- 4 Der Übergang des Workflows wird aktiviert.
- 5 Der Standort der untergeordneten Vermögensgegenstände wird geändert.
- 6 Das gesamte Ereignis wird bestätigt.

Sollte es bei einem dieser Schritte zu einem Fehler kommen, werden weder der Standort des Vermögensgegenstands noch die Standorte der untergeordneten Vermögensgegenstände geändert.

Wenn die Schritte erfolgreich durchgeführt werden konnten, werden alle Standorte geändert.

Wenn Sie jedoch dieselbe Prozedur über einen asynchronen Workflow ausführen, wie er im Folgenden abgebildet ist, und es im Verlauf des Workflows zu einem Fehler kommt, kann der Standort des Vermögensgegenstands geändert werden, ohne dass sich diese Änderung auf die untergeordneten Vermögensgegenstände auswirkt.

**Abbildung 8.9. Beispiel eines asynchronen Workflows**



# Abschließendes Ereignis

## Definition

Ein abschließendes Ereignis beendet eine Workflow-Instanz, auch wenn noch weitere Aufgaben im Rahmen der Instanz ausgeführt werden müssen.

Beispiel:

**Abbildung 8.10. Workflow-Modell mit einem abschließenden Ereignis**



Wenn eine Workflow-Instanz dem oben abgebildeten Schema gemäß abläuft, geschieht Folgendes:

- Das Ausgabeereignis der Aktivität 1 tritt ein und löst die Aktivität 2 aus. Dabei wird die auszuführende Aufgabe erstellt.
- Das abschließende Ausgabeereignis der Aktivität 3 tritt ein.

Daraufhin wird die Workflow-Instanz beendet, auch wenn die Aufgabe, die sich aus der Aktivität 2 ergibt, noch nicht durchgeführt wurde.

## Abschließendes Ereignis definieren

Wenn Sie über den Grafikeditor auf der Registerkarte **Aktivitäten** des Modelldetails ein Workflow-Modell erstellen, kennzeichnen Sie ein Ereignis wie folgt als abschließend:

- 1 Wählen Sie das Ereignis. Drücken Sie die rechte Maustaste, um das Kontextmenü anzuzeigen.
- 2 Wählen Sie die Option **Abschlussereignis**.

## Workflow-Übergänge

Die Übergänge verknüpfen die Ausgabeereignisse einer Aktivität mit den anderen Aktivitäten.

Ein Ereignis kann mit mehreren Übergängen verknüpft sein.

Sofern erforderlich, können Sie die Aktivierungsbedingungen des Übergangs im Feld **AQL-Bedingung** (SQL-Name: AQLCond) des Übergangsdetails festlegen.

## Warnungen und Fristen innerhalb eines Workflow-Modells

Für jede Workflow-Aktivität legen Sie folgendes fest:

- Eine Frist für die Ausführung der Aktivität.
- Warnungen, die sich entweder auf diese Fristen oder auf die Daten beziehen, die in der Datenbank gespeichert sind. Über diese Warnungen werden die AssetCenter Aktionen gestartet.

---

### **WARNUNG:**

Wenn Sie die Option **Aufgabe speichern** (SQL-Name: bLogWorkItem) auf der Registerkarte **Allgemein** des Aktivitätendetails nicht markieren, können Sie weder Fristen noch Warnungen definieren.

---

## Fristen

Die Definition der Frist, innerhalb derer eine Workflow-Aktivität ausgeführt worden sein muss, erfolgt auf der Registerkarte **Fristen** des Aktivitätendetails.

---

### **Anmerkung:**

Die Registerkarte **Fristen** wird im Detailfenster der Aktivität nur angezeigt, wenn die Option **Aufgabe speichern** (SQL-Name: bLogWorkItem) auf der Registerkarte **Allgemein** des Aktivitätendetails aktiviert wurde.

---

Diese Frist wird in bezug auf den Zeitpunkt definiert, zu dem die Aktivität ausgeführt wurde.

Die Frist ist dem Kalender mit den Werktagen zugeordnet.



Sie können eine Dauer eingeben und einen der drei vorgegebenen Einträge wählen:

- **Ende des nächsten Arbeitstags**
- **Ende der Arbeitswoche**
- **Ende des Arbeitsmonats**

---

 **WARNUNG:**

Wenn Sie eine Dauer angeben, geht AssetCenter davon aus, dass es sich um einen gearbeiteten Zeitraum handelt und konvertiert diesen Zeitraum in gearbeitete Stunden. Beispiel: Wenn Sie eine Frist von "2 Tagen" eingeben, versteht AssetCenter 48 gearbeitete Stunden.

---

## Warnungen

Sie haben die Möglichkeit, jeder Aktivität auf der Registerkarte **Warnungen** des Aktivitätendetails verschiedene Warnungen zuzuordnen.

---

 **Anmerkung:**

Die Registerkarte **Warnungen** wird im Aktivitätendetail nur angezeigt, wenn die Option **Aufgabe speichern** (SQL-Name: bLogWorkItem) auf der Registerkarte **Allgemein** des Aktivitätendetails markiert wurde.

---

## Fälligkeiten

Wählen Sie zum Definieren der Fälligkeiten, die zum Auslösen von Warnungen führen, eine der nachstehenden Vorgehensweisen:

- Über eine Frist, die nach einem in der Datenbank gespeicherten Datum liegt (Art **Zeitspanne seit**)
- Über eine Frist, die vor einem in der Datenbank gespeicherten Datum liegt (Art **Zeitspanne vor**)
- Über den Prozentsatz einer Frist, vor deren Ablauf die Aktivität ausgeführt worden sein muss (Feld **Frist** (SQL-Name: tsResolDelay) der Registerkarte **Fristen**)

**Anmerkung:**

Bei den Fristen, mit denen die Workflow-Fälligkeiten definiert sind, handelt es sich um gearbeitete Fristen.

Unmittelbar nach der Erstellung einer Aufgabe werden die der Aufgabe zugeordneten Workflow-Warnungen generiert.

Die Workflow-Fälligkeiten werden von AssetCenter Server überwacht. Die Häufigkeit, mit der die Überwachung durchgeführt wird, wird in den Optionen von AssetCenter Server festgelegt.

## Was die Warnungen auslösen

Warnungen lösen folgendes aus:

- AssetCenter-Aktionen.
- Ereignisse. Über Warnungen ausgelöste Ereignisse entsprechen dem Typ **Warnung**. Sie tragen den Namen der Warnungen, über die sie definiert wurden.

## Workflow-Ausführungsgruppen

Workflow-Ausführungsgruppen ermöglichen das Klassifizieren der von Ihnen definierten Workflow-Modelle. Die Ausführungsgruppe, zu der ein Workflow-Modell gehört, erscheint im Feld **Gruppe** (SQL-Name: GroupName) der Registerkarte **Allgemein** des Workflow-Details.

AssetCenter Server prüft die Erstellung von neuen Workflow-Ausführungsgruppen.

Erkennt AssetCenter Server eine neue Ausführungsgruppe "G" für Workflow-Modelle, erstellt das Programm ein neues Überwachungsmodul vom Typ **Workflow-Regeln für Ausführungsgruppe 'G' ausführen**.

Dieser Mechanismus bringt nachstehende Vorteile mit sich:

- Er ermöglicht die Definition von Überwachungsdauern für einzelne Workflow-Ausführungsgruppen.
- Die einzelnen Workflow-Ausführungsgruppen können von verschiedenen AssetCenter Server-Programmen überwacht werden.

Sobald eine Workflow-Ausführungsgruppe erkannt wird, prüft AssetCenter Server die Workflow-Regeln, und führt die für die entsprechende

Workflow-Ausführungsgruppe geltenden Regeln aus (Überwachung der Warnungen, Verarbeitung der Ereignisse vom Typ **In regelmäßigen Abständen**, Aktivierung der Übergänge, Ausführung der Aufgaben usw.).

## Workflow-Protokoll

Wenn es sich bei der AssetCenter-Tabelle um den Kontext des Ausgangsobjekts eines Workflows handelt, erscheint im Detailfenster der Datensätze dieser Tabelle die Registerkarte **Workflow**.

Auf der Registerkarte **Workflow** des Detailfensters eines Datensatzes aus dieser Tabelle erscheint der Status der Workflow-Instanzen, für die dieser Datensatz das Ausgangsobjekt darstellt.

Jede Workflow-Instanz wird in einer Unterregisterkarte beschrieben, in der der Ablauf der Instanz genau festgelegt ist:

- In der linken Hälfte der Unterregisterkarte werden die bereits eingetretenen Ereignisse aufgeführt.
- In der rechten Hälfte der Unterregisterkarte erscheint das Workflow-Modell. Die Aktivitäten, die ausgeführt werden müssen, blinken. Die jeweils nächsten Schritte sind grau unterlegt.

## Abgeschlossene Workflow-Instanzen löschen

### Gründe für das Löschen von abgeschlossenen Workflow-Instanzen

Bei der Ausführung von Workflow-Modellen werden Workflow-Instanzen erstellt (Tabelle **Workflow-Instanzen** (amWfInstance)).

Diese Instanzen werden nach Abschluss der Ausführung nicht automatisch gelöscht (Feld **Status** (seStatus)).

Wenn Sie oft mit Workflow-Modellen arbeiten, entstehen also sehr viele Workflow-Instanzen.

Dadurch wird die Datenbank unnötig groß, so dass die Leistungsfähigkeit von AssetCenter eventuell beeinträchtigt wird.

Es ist daher ratsam, vollständig ausgeführte Workflow-Instanzen regelmäßig zu löschen.

## Das Löschen von abgeschlossenen Workflow-Instanzen automatisieren.

So automatisieren Sie das Löschen abgeschlossener Workflow-Instanzen in Ihrer Betriebsdatenbank:

- 1 Fügen Sie der Tabelle **Workflow-Modelle** (amWfScheme) ein Feld hinzu, über das ein Zeitraum festgelegt wird, nach dessen Ablauf veraltete Instanzen gelöscht werden.
- 2 Erstellen Sie eine Aktion zum Löschen veralteter Workflow-Instanzen.
- 3 Erstellen Sie ein Workflow-Modell, um die Ausführung dieser Aktion zu automatisieren.
- 4 Konfigurieren Sie AssetCenter Server für die automatische Ausführung des Workflow-Modells.

### Der Tabelle Workflow-Modelle ein Feld hinzufügen

Fügen Sie der Tabelle **Workflow-Modelle** (amWfScheme) das folgende Feld hinzu:

Parameter	Wert
SQL-Name	AutoCleaningDelay
Bezeichnung	Zeitspanne vor Löschen der Instanzen
Beschreibung	Zeitraum, nach dessen Ablauf abgeschlossene Workflow-Instanzen gelöscht werden müssen
Typ	Dauer
Index für das Feld erstellen	Wählen Sie diese Option nicht.
Beschreibung (Registerkarte <b>Hilfe</b> )	<LI>Legt fest, nach welcher Zeitspanne abgeschlossene Workflow-Instanzen gelöscht werden können.
Beispiel	<LI>-1: Die Workflow-Instanzen werden nie gelöscht.<LI>Wert positiv oder Null: Die Workflow-Instanzen können nach Ablauf der Zeitspanne gelöscht werden.

Informationen zum Hinzufügen von Feldern zu einer vorhandenen Tabelle finden Sie im Handbuch **Verwaltung**, Kapitel **Anpassung der Datenbank anpassen**, Abschnitt **Neue Objekte erstellen/ Felder, Verknüpfungen oder Indizes erstellen**.

## Das Feld Zeitspanne vor Löschen der Instanzen ausfüllen

Geben Sie für jedes verwendete Workflow-Modell einen Wert in das Feld **Zeitspanne vor Löschen der Instanzen** (AutoCleaningDelay) ein.

## Eine Verbindung mit der Betriebsdatenbank herstellen

- 1 Starten Sie AssetCenter.
- 2 Stellen Sie die Verbindung mit Ihrer Betriebsdatenbank her.

## Eine Aktion zum Löschen veralteter Instanzen erstellen

- 1 Zeigen Sie die Liste der Aktionen (Menü **Extras/ Aktionen/ Bearbeiten**) an.
- 2 Klicken Sie auf **Neu**.
- 3 Füllen Sie folgende Felder aus:

Bezeichnung	SQL-Name	Wert
Name	Name	Abgeschlossene Workflow-Instanzen löschen
Kontext	ContextTable	Workflow-Modelle (amWfScheme)
Typ	seActionType	Skript
SQL-Name	SQLName	DeleteFinishedWfInstances
Aktionsskript	Skript	Siehe nachstehendes (*).

(\*) Aktionsskript:

```
Const NumberOfInstanceToDelete = 50

Dim lRc As Long
Dim i As Long

i = 0

If [AutoCleaningDelay] >= 0 Then
    Dim hqWfInstance As Long
    hqWfInstance = AmQueryCreate()
    lRc = AmQueryExec(hqWfInstance, "SELECT lwfInstanceId FROM amWfInstance WHERE lwfSchId = "& [lwfSchId] & " And seStatus = 1 AND ADDS ECONDS(dtCompleted, " & [AutoCleaningDelay] & ") < GetDate()" )

    Do While (lRc = 0 And i < NumberOfInstanceToDelete)
        Dim hrWfInstance As Long
        hrWfInstance = AmGetRecordHandle(hqWfInstance)
        lRc = AmDeleteRecord(hrWfInstance)
        lRc = AmReleaseHandle(hrWfInstance)
        lRc = AmQueryNext(hqWfInstance)
    End Do
End If
```

```

    i = i + 1
  Loop
End If

```

- 4 Klicken Sie auf **Erstellen**.

## Workflow-Modell zur Ausführungsautomatisierung der Aktion erstellen

- 1 Zeigen Sie die Liste der Workflow-Modelle (Menü **Extras/ Workflow/ Workflow-Modelle**) an.
- 2 Klicken Sie auf **Neu**.
- 3 Füllen Sie folgende Felder aus:

Bezeichnung	SQL-Name	Wert
Name	Name	Löschen abgeschlossener Workflow-Instanzen
Referenz	Ref	ADM_CLEAN_WF_INSTC
Ausgangsobjekt-Kontext	StartContextTable	Workflow-Modelle (amWfScheme)
Ausführungsgruppe	GroupName	Geben Sie einen beliebigen Namen ein, um die Ausführung des Workflow-Modells in AssetCenter Server zu automatisieren (beispielsweise <b>ADMIN</b> ).
Ein laufender Workflow pro Objekt	bUniqueActive	Wählen Sie diese Option.
Keine Instanzen in der Datenbank speichern	bTransient	Wählen Sie diese Option nicht.

- 4 Klicken Sie auf **Erstellen**.
- 5 Wählen Sie die Registerkarte **Aktivitäten**.
- 6 Klicken Sie mit der rechten Maustaste auf die Aktivität **Start**, und wählen Sie das Menü **Ereignis hinzufügen**.
- 7 Füllen Sie folgende Felder aus:

Bezeichnung	SQL-Name	Wert
Name	Name	Timer
Typ	seMonitoringType	In regelmäßigen Abständen
Workflow neu starten, wenn bereits ein anderer Workflow läuft	In regelmäßigen Abständen	Wählen Sie diese Option nicht.

Bezeichnung	SQL-Name	Wert
AQL-Bedingung	AQLCond	AutoCleaningDelay >= 0

- 8 Wählen Sie die Registerkarte **Parameter**, und geben Sie die gewünschten Werte ein.
- 9 Klicken Sie auf **Hinzufügen**.
- 10 Klicken Sie mit der rechten Maustaste, und wählen Sie das Menü **Aktivität hinzufügen**.
- 11 Füllen Sie folgende Felder aus:

Bezeichnung	SQL-Name	Wert
Name	Name	Clean W/F instances
Typ	seType	Automatische Aktion
Aufgabe speichern	bLogWorkItem	Wählen Sie diese Option.
Kontext	ContextTable	Workflow-Modelle (amWfScheme)
Eingabebedingung	seInCond	OR

- 12 Klicken Sie auf **Hinzufügen**.
- 13 Zeigen Sie die Registerkarte **Parameter** an.
- 14 Füllen Sie folgende Felder aus:

Bezeichnung	SQL-Name	Wert
Aktionen unmittelbar ausführen	bExecImmediately	Wählen Sie diese Option.
Aktionen	Aktionen	Abgeschlossene Workflow-Instanzen löschen

- 15 Klicken Sie auf **Schließen**.
- 16 Doppelklicken Sie auf die Aktivität **Clean W/F instances**.
- 17 Klicken Sie mit der rechten Maustaste auf das Ereignis **Ausgeführt**, und wählen Sie das Menü **Ereignisdetail**.
- 18 Füllen Sie folgende Felder aus:

Bezeichnung	SQL-Name	Wert
Bearbeitung	seProcessingMode	Ereignis speichern und unmittelbar bearbeiten

- 19 Klicken Sie auf **Schließen**.
- 20 Stellen Sie mit der Maus die Verbindung zwischen den Aktivitäten **Start** und **Clean W/F instances** her.

- 21 Klicken Sie mit der rechten Maustaste auf das Ereignis **Ausgeführt**, und wählen Sie das Menü **Abschlussereignis**.
- 22 Klicken Sie auf **Ändern**.

### **AssetCenter Server für die Ausführungsautomatisierung von Workflow-Modellen konfigurieren**

- 1 Starten Sie AssetCenter Server.
- 2 Konfigurieren Sie das Modul, das die Ausführung des Workflows **Abgeschlossene Workflow-Instanzen löschen** auslösen soll (Menü **Extras/Module konfigurieren**).  
Dabei handelt es sich um das Modul mit dem Namen **Workflow-Regeln für Ausführungsgruppe 'X' ausführen**, wobei **X** der Wert des im Workflow-Modell definierten Felds **Ausführungsgruppe** (GroupName) ist.
- 3 Lassen Sie AssetCenter Server aktiv, wenn der Workflow automatisch ausgeführt werden soll.



# 9 | Export von Daten und Verwaltung von SQL-Ansichten

## KAPITEL

In diesem Kapitel wird der Export von Daten und die Verwaltung der SQL-Ansichten der Datenbank mit AssetCenter erklärt.

## Definitionen eines Skripts und einer Exportabfrage

### Exportskript

Mit Hilfe der Exportskripte können Sie unter Verwendung von AssetCenter Export oder **amexp.exe** Daten exportieren oder SQL-Ansichten erstellen bzw. löschen. Zusätzlich besteht die Möglichkeit, ein von Ihnen definiertes Exportskript erneut zu verwenden.

Ein Exportskript enthält die nachstehenden Elemente:

- Exportabfragen, mit denen Sie die Informationen definieren, die aus der Datenbank extrahiert werden sollen.
- Exportoptionen, mit denen Sie das Ausgabeformat festlegen.
- Aktionen, die im Fall der Erstellung bzw. dem Löschen von SQL-Ansichten ausgeführt werden sollen.

Ein Exportskript kann auf zwei Arten verwendet werden:

- Im "Exportmodus" zum Exportieren von Daten
- Im "Ansichtenmodus" zum Erstellen oder Löschen von SQL-Ansichten der Datenbank

## Exportabfragen

Die Definition der Exportabfragen erfolgt mit Hilfe von AssetCenter Export.

Eine Exportabfrage wird mit Hilfe der nachstehenden Elemente definiert:


- Einem Namen
- Gegebenenfalls einer Exportdatei (im "Exportmodus")
- Einem Kommentar, der nicht exportiert wird
- Einer Ausgangstabelle
- Einer Liste mit den Spalten, die extrahiert werden sollen (Felder, Verknüpfungen, Merkmale und berechnete Felder in der Ausgangstabelle) sowie den zugeordneten Sortierkriterien
- Einem Filter mit der Klausel WHERE, der die Extraktionsbedingungen festlegt
- Einem Filter mit der Klausel HAVING, der die Extraktionsbedingungen festlegt
- Einer Klartextanzeige der Abfrage, die den Registerkarten **Filter (WHERE-Klausel)** und der "Klausel HAVING" entspricht
- Einer Registerkarte zur Vorschau

## Daten der AssetCenter-Datenbank exportieren

Beim Exportieren von Daten der AssetCenter-Datenbank in Textdateien haben Sie die Wahl zwischen zwei Möglichkeiten:

- Mit Hilfe eines Exportskripts.
- Über das Kontextmenü **Liste exportieren**. Dieses Menü wird angezeigt, wenn wenigstens eine Liste oder eine Registerkartenliste angezeigt ist. Es ermöglicht das Exportieren der aktiven Liste.

## Daten über ein Exportskript exportieren

- 1 Starten Sie AssetCenter Export. Das Programm gehört zu den Programmen des Menüs **Start** bzw. zur AssetCenter-Programmgruppe.
- 2 Definieren Sie ein Exportskript im Modus "Daten exportieren":
  - 1 Schreiben Sie auf der Registerkarte **Abfragen** die Abfrage, mit denen Sie die Daten definieren, die exportiert werden sollen.
  - 2 Geben Sie auf der Registerkarte **Optionen** das Format der Textdateien an, in die die Daten exportiert werden sollen.
  - 3 Speichern Sie das Exportskript über das Menü **Datei/ Skript speichern** oder das Menü **Datei/ Skript speichern unter**.
- 3 Führen Sie das Exportskript aus wie folgt:
  - Entweder direkt in AssetCenter Export über das Symbol <Skript ausführen> 
  - Durch Starten des Programms **amexpl.exe** unter DOS.



### Anmerkung:

Um die Kohärenz der in AssetCenter definierten Zugriffsbeschränkungen aufrechtzuerhalten, ist das Starten von AssetCenter Export Export bzw. das Ausführen des Programms **amexpl.exe** dem Administrator (Login "Admin" bzw. einem Benutzer mit den entsprechenden administrativen Rechten) vorbehalten.

## Daten über das Kontextmenü Liste exportieren exportieren

Das Kontextmenü **Liste exportieren** steht jedem AssetCenter-Benutzer zur Verfügung. Der Benutzer kann damit alle Daten exportieren, die er auch auf den Bildschirm rufen darf.

- 1 Zeigen Sie die Liste an, die Sie exportieren möchten (Hauptliste oder Registerkartenliste). Wenn mehrere Listen gleichzeitig auf dem Bildschirm erscheinen, müssen Sie auf die richtige Position des Cursors achten.
- 2 Wählen Sie das Kontextmenü **Liste exportieren**.
- 3 Geben Sie die erforderlichen Daten im Fenster ein, und klicken Sie dann auf die Schaltfläche **Exportieren**.

---

 **Anmerkung:**

Weitere Informationen zum Menü **Datei/ Export** exportieren finden Sie im Handbuch **Einführung in AssetCenter**, Kapitel "Erste Schritte mit AssetCenter", Abschnitt "Datensatzlisten", Unterabschnitt "Listen exportieren".

---

## SQL-Ansichten der AssetCenter-Datenbank verwalten

AssetCenter Export ermöglicht das Erstellen, erneute Erstellen und Löschen von SQL-Ansichten der AssetCenter-Datenbank. Diese Ansichten können anschließend von externen Programmen bearbeitet werden, die anstelle von Textdateien Ansichten verwenden.

---

 **Anmerkung:**

Die SQL-Ansichten, die mit den Exportskripten erstellt, geändert bzw. gelöscht werden, sind nicht mit den Ansichten im Sinne von AssetCenter zu verwechseln. Die SQL-Ansichten entsprechen dem SQL-Befehl "CREATE VIEW".

---

Gehen Sie zum Erstellen, erneuten Erstellen und Löschen von SQL-Ansichten der AssetCenter-Datenbank wie folgt vor:

- 1 Starten Sie AssetCenter Export.
- 2 Definieren Sie ein Exportskript im Modus "SQL-Ansichten erstellen/entfernen":
  - 1 Schreiben Sie in die Registerkarte Abfragen die **Abfragen**, mit denen die zu extrahierenden Daten definiert werden.
  - 2 Legen Sie auf der Registerkarte **Ansichten** die Aktionen fest, die Sie ausführen möchten: Erstellen, Ändern, bzw. Löschen von Ansichten, direktes Ausführen der generierten SQL-Ansicht bzw. Speichern in einer Datei.
  - 3 Speichern Sie das Exportskript.
- 3 Führen Sie das Exportskript aus wie folgt:
  - Entweder direkt in AssetCenter Export oder
  - Durch Starten des Programms **amexpl.exe** unter DOS.

## Empfehlungen

Wir empfehlen Ihnen, die "Identifizierungsschlüssel" der Tabelle nicht als Abstimmungsschlüssel zu verwenden, wenn Sie die zuvor exportierten Daten erneut importieren wollen. Die entsprechenden Identifizierungsschlüssel sind nicht fest definiert und können durchaus geändert werden. Aus diesem Grund ist die Verwendung von Schlüsseln mit "unveränderlichen" Werten vorzuziehen. Dazu zählt beispielsweise der interne Code der Vermögensgegenstände.

## Exportskripte definieren

Zum Exportieren von Daten bzw. Verwalten der SQL-Ansichten ihrer Datenbank müssen Sie die Exportskripte und die Exportabfragen definieren, aus denen sie sich zusammensetzen. Die Definition erfolgt über AssetCenter Export.

Ein Exportskript enthält die nachstehenden Elemente:

- Exportabfragen, mit denen Sie die Informationen definieren, die aus der Datenbank extrahiert werden sollen.
- Exportoptionen, mit denen Sie das Ausgabeformat festlegen.

Aktionen, die im Fall der Erstellung bzw. dem Löschen von SQL-Ansichten ausgeführt werden sollen.

Ein Exportskript kann auf zwei Arten verwendet werden:

- Im "Exportmodus" zum Exportieren von Daten
- Im "Ansichtenmodus" zum Erstellen oder Löschen von SQL-Ansichten der Datenbank

In diesem Abschnitt wird die Erstellung von Exportskripten beschrieben:

- Vorgehensweise
- Exportabfragen definieren
- Ausgabeformat eines Exportskripts definieren
- SQL-Ansichten bearbeiten

## Vorgehensweise

Gehen Sie zum Erstellen und Ändern eines Exportskripts wie folgt vor:

- 1 Starten Sie AssetCenter Export.

- 2 Öffnen Sie die geeignete Datenbank. Bedenken Sie, dass Sie die Verbindung nur als Administrator herstellen können, also als Benutzer mit dem Login "Admin" oder als Benutzer mit den entsprechenden administrativen Rechten.
- 3 Erstellen Sie über das Menü **Datei/ Neues Skript** ein neues Skript oder öffnen Sie über das Menü **Datei/ Skript öffnen** ein Skript, das Sie ändern möchten.
- 4 Legen Sie oben auf dem AssetCenter Export-Bildschirm fest, ob Sie Daten exportieren ("Daten exportieren") oder die SQL-Ansichten der Datenbank verwalten möchten ("SQL-Ansichten erstellen/entfernen").
- 5 Schreiben Sie die Abfragen für das Exportskript in die Registerkarte **Abfragen**.
- 6 Wenn Sie Daten exportieren, müssen Sie das Ausgabeformat für die exportierten Daten auf der Registerkarte **Optionen** festlegen.
- 7 Wenn Sie SQL-Ansichten verwalten, definieren Sie die Vorgänge, die Sie auf der Registerkarte **Ansichten** ausführen möchten.
- 8 Speichern Sie das Skript über das Menü **Datei/ Skript speichern** oder **Datei/ Skript speichern unter**.

## Exportabfragen definieren

Die Definition der Abfragen des Exportskripts erfolgt auf der Registerkarte **Abfragen** von AssetCenter Export.

- Klicken Sie auf die Schaltfläche **Neu**, um eine Exportabfrage hinzuzufügen.
- Klicken Sie auf die Schaltfläche **Löschen**, um die ausgewählten Exportabfragen zu löschen.

## Abfragen für ein Exportskript erstellen

- 1 Klicken Sie auf der Registerkarte **Abfragen** auf die Schaltfläche **Neu**.
- 2 Definieren Sie den Namen der Abfrage. Dieser Name wird in dem Ausführungsbericht der Registerkarte **Meldungen** im Detailfenster des Exportskripts verwendet.
- 3 Sie können einen Kommentar eingeben, der nicht exportiert wird.
- 4 Die Definition der Daten, die extrahiert werden sollen, erfolgt im Feld **Abfrage**.

- 5 Wenn Sie die Daten nur exportieren und keine Ansichten erstellen, ändern oder löschen wollen, geben Sie im Feld **Datei** den Pfad und den Namen der Ausgabedatei ein, in die die von der Abfrage ausgewählten Daten exportiert werden sollen. Ein Exportskript, das mehrere Exportabfragen enthält, generiert damit mehrere Textdateien.




Anmerkung:

Das Feld **Datei** wird nicht angezeigt, wenn Sie die Option **SQL-Ansichten erstellen/entfernen** ausgewählt haben.

## Zu extrahierende Daten

Zum Festlegen der Daten, die extrahiert werden sollen, geben Sie die erforderlichen Informationen im Feld **Abfrage** des Detailfensters mit der Abfrage des Exportskripts ein. Die Abfrage bezieht sich auf eine Tabelle der AssetCenter-Datenbank.

Sie können die Abfrage entweder direkt eingeben oder auf die Schaltfläche  klicken, um auf das Fenster zuzugreifen, in dem die Anweisungen zum Definieren einer Abfrage erscheinen:

## Zu exportierende Spalten und Sortiervorgang

Auf der Registerkarte **Spalten und Sortieren** definieren Sie die Liste der Felder, Verknüpfungen, Merkmale und berechneten Felder, die exportiert werden sollen, sowie die entsprechenden Sortierkriterien.

Wählen Sie in der Liste auf der linken Seite nacheinander die Felder, Verknüpfungen, Merkmale und berechneten Felder, die dem Export dienen, und klicken Sie auf den Pfeil, um sie in die Liste auf der rechten Seite einzufügen.

Gehen Sie für jede Zeile in der Liste auf der rechten Seite wie folgt vor:

- Markieren Sie das Kontrollkästchen **Sichtbar**, damit die Spalte exportiert wird. Wenn das Kästchen **Sichtbar** nicht markiert ist, kommt es auch nicht zum Export der Spalte. In diesem Fall kann die Spalte jedoch noch zum Sortieren der exportierten Daten verwendet werden.
- Markieren Sie das Kontrollkästchen **Gruppieren**, um die Gruppierungen für das Feld der entsprechende Spalte auszuführen. Genauso gut könnten Sie auch die Klausel "GROUP BY <Name des Felds>" zur SQL-Abfrage hinzufügen.

Beispiel:

```
SELECT Brand, Count (lModelId) FROM amModel GROUP BY Brand ORDER BY Brand
```

 **WARNUNG:**

Durch das Markieren des Kontrollkästchens **Gruppieren** wird die Klausel "GROUP BY" am Ende der Abfrage hinzugefügt. Damit die Abfrage jedoch ihre volle Gültigkeit erhält, müssen Sie die geeigneten Aggregatfunktionen zur Klausel SELECT hinzufügen.

Definieren Sie die Sortierreihenfolge der exportierten Daten wie folgt:

- 1 Wählen Sie einen Index im Feld **Nach Index**, um einen Sortiervorgang anhand des Indizes zu definieren.
- 2 Ansonsten markieren Sie die Kontrollkästchen **Sortieren** in der gewünschten Sortierreihenfolge.

 **Anmerkung:**

Wenn es sich bei der Engine Ihrer Datenbank nicht um SQL Anywhere handelt, können Sie das Kontrollkästchen **Indizes setzen** markieren, um die Verwendung der in der Abfrage angegebenen Indizes zu erzwingen. Weitere Informationen zu diesem Thema finden Sie im Handbuch "Referenzhandbuch: Verwaltung und erweiterte Funktionen von AssetCenter", Kapitel "Schreiben von AQL-Abfragen", Abschnitt "Sortiervorgänge und Indizes".

Markieren Sie das Kontrollkästchen **Nur eindeutige Datensätze**, wenn identische Zeilen ein einziges Mal exportiert werden sollen. Das entspricht dem Hinzufügen der Klausel "DISTINCT" zur SQL-Abfrage.

Beispiel mit nicht markiertem Kästchen **Nur eindeutige Datensätze**:

```
SELECT Brand FROM amModel
```

Beispiel mit markiertem Kästchen **Nur eindeutige Datensätze**:

```
SELECT DISTINCT Brand FROM amModel
```

## Filter

Beim Auswählen der zu extrahierenden Daten können Sie zwei verschiedene Arten von Filtern definieren:

- Auf der Registerkarte **Filter (WHERE-Klausel)** definieren Sie eine AQL-Abfrage, die die Klausel WHERE verwendet.




- Auf der Registerkarte **HAVING-Klausel** definieren Sie eine AQL-Abfrage, die die Klausel HAVING verwendet

## Abfrage anzeigen

Die AQL-Abfrage, die Sie mit Hilfe der Registerkarten **Spalten und Sortieren**, **Filter (WHERE-Klausel)** und **HAVING-Klausel** definieren, erscheint auf der Registerkarte **Abfrage**.

## Vorschau des Abfrageergebnisses

Das Testen der Abfrage sowie ihre Anzeige in der Programmiersprache SQL erfolgt auf der Registerkarte **Vorschau**.

Mit einem einfachen Mausklick auf das Symbol  erhalten Sie eine Vorschau des Abfrageergebnisses in Form einer Datensatzliste. Beachten Sie, dass AssetCenter die Anzahl der Ihrer Abfrage entsprechenden Datensätze unten rechts im Fenster anzeigt.

## Ausgabeformat eines Exportskripts definieren

Wenn Sie die Option **Daten exportieren** wählen, wählen Sie das Format der Ausgabertextdateien auf der Registerkarte **Optionen**. Dieses Format gilt für sämtliche Exportabfragen.

---

 **Anmerkung:**

Wenn Sie sich für das Löschen, Erstellen bzw. erneute Erstellen von Ansichten entscheiden, wird die Registerkarte **Optionen** nicht angezeigt.

---

## Spaltentitel

Wählen Sie einen Wert, wenn in der ersten Zeile der Exportdatei eine der nachstehenden Informationen erscheinen soll:

- Alias der Spalten, die in der Exportabfrage angegeben sind.
- "SQL-Name" der Felder und Verknüpfungen, die den Spalten entsprechen.
- "Beschreibung" der Felder und Verknüpfungen, die den Spalten entsprechen.

## Spaltentrennzeichen

Dieses Zeichen wird zwischen den Informationen jeder Spalte eingefügt.

## Textidentifikation

Mit diesem Zeichen werden sämtliche Strings vom Typ Text abgegrenzt. Wird das Zeichen ' zur Abgrenzung verwendet, erscheint das Zeichen ' nach dem Export als ". Dasselbe gilt für das Zeichen ".

## Zeichensatz

Diese Option lässt Sie zwischen den Zeichensätzen ANSI (Windows) und OEM (DOS) wählen.

## Dezimaltrennzeichen

Dieses Zeichen wird zur Abgrenzung von Dezimalstellen bei exportierten Zahlen verwendet.

## Datumtrennzeichen

Mit diesem Trennzeichen grenzen Sie den Tag, den Monat und das Jahr der exportierten Datumsangaben ab.

## Datumformat

Mit diesen Optionen definieren Sie die Reihenfolge der Anzeige von Tag (JJ), Monat (MM) und Jahr (AA).

## Jahresformat

Je nach der hier gewählten Option werden die Jahresangaben mit 2 oder 4 Ziffern exportiert.

## Stundentrennzeichen

Dieses Trennzeichen wird zur Abgrenzung von Stunden, Minuten und Sekunden verwendet.

## Sekunden anzeigen

Markieren Sie diese Option, wenn für die exportierten Stunden die Sekunden angezeigt werden sollen.

## SQL-Ansichten bearbeiten

Wenn Sie bestimmte, den Exportabfragen entsprechende SQL-Ansichten löschen oder (neu) erstellen möchten, müssen Sie die Aktionen, die dabei ausgeführt werden sollen, auf der Registerkarte **Ansichten** definieren.


 **Anmerkung:**

Wenn Sie die Option **Daten exportieren** wählen, wird die Registerkarte **Ansichten** nicht angezeigt.

Wählen Sie eine der Aktionen, die Sie ausführen möchten, im Rahmen "Aktionen":

- Ansichten erstellen/ neu erstellen
- Ansichten entfernen

Legen Sie im Abschnitt "SQL-Skript zur Ansichtenbearbeitung" die Vorgänge fest, die Sie über die Registerkarte **Abfragen** ausführen möchten, und geben Sie die Option an, die Sie im Rahmen **Aktionen** ausgeführt haben:

- Wählen Sie die Option **SQL direkt ausführen**, wenn Sie die SQL-Ansichten direkt bei der Ausführung des Skripts erstellen oder löschen möchten.
- Wählen Sie die Option **SQL in Datei speichern**, um ein SQL-Skript zum Erstellen (Befehl "CREATE VIEW") oder zum Löschen von Ansichten (Befehl "DROP VIEW") zu generieren. Gehen Sie dann wie folgt vor:
  - 1 Geben Sie den Namen und den Pfad der Datei ein, in der das generierte SQL-Skript gespeichert werden soll, und klicken Sie auf die Schaltfläche .
  - 2 Wählen Sie das Trennzeichen für die SQL-Befehle unter ";" (Oracle-Datenbanken) und "GO" (für alle anderen Datenbanken).

## Exportskripte ausführen


Zum Exportieren von Daten und Verwalten von SQL-Ansichten müssen Sie Exportskripte ausführen.

In diesem Abschnitt werden die beiden Vorgehensweisen zum Ausführen eines Exportskripts beschrieben:

- Exportskripte über AssetCenter Export ausführen
- Exportskripte unter DOS ausführen

## Exportskripte über AssetCenter Export ausführen





Zum Ausführen eines Exportskripts über AssetCenter Export gehen Sie wie folgt vor:

- 1 Starten Sie AssetCenter Export.
- 2 Definieren Sie Ihr Exportskript und speichern Sie es.
- 3 Führen Sie das Skript aus:
  - Verwenden Sie das Menü **Aktionen/ Skript ausführen**, oder
  - Drücken Sie auf die Funktionstaste F8, oder
  - Klicken Sie auf das Symbol .

Die Informationen zum Verlauf des Exports erscheinen auf der Registerkarte **Meldungen**.

Wenn der Export erfolgreich durchgeführt wurde, erscheint als letzte Meldung "Die Ausführung des Skripts XXX wurde erfolgreich abgeschlossen". Sollte es zu einem Fehler kommen, erscheint die Meldung "Fehler beim Ausführen des Skripts XXX".

Jeder Meldung wird ein Symbol vorangestellt:

-  Allgemeine Information
-  Fehler
-  Erfolgreicher Export
-  Warnung

## Exportskripte unter DOS ausführen

### Prinzip

Vor dem Ausführen des Online-DOS-Programms, müssen Sie mit Hilfe von AssetCenter Export ein Exportskript erstellen.

Anschließend können Sie einen Exportbefehl manuell eingeben oder, beispielsweise über eine Batch-Datei, mit Hilfe des Programms **amexpl.exe** automatisch ausführen. Dieses Programm befindet sich im Unterverzeichnis **bin** des AssetCenter-Installationsverzeichnis.

## Syntax

```
amexpl [-verbose] [-?|h|H] -script:<skript>
-cnxf:<Verbindung> [-login:<Login>]
[-password:<Kennwort>]
```

-verbose: Anzeigen der Meldungen im Verlauf des Exports

-, -h oder -H: Anzeigen der Hilfe über das Programm

-script: Pfad und Name des auszuführenden Exportskripts

-cnxf: Name der Verbindung zur AssetCenter-Datenbank, wie er im Menü **Datei/ Verbindungen bearbeiten**) erscheint

-login: Login-Name eines Administrators der Datenbank ("Admin" oder Benutzer mit den entsprechenden administrativen Rechten)

-password: Dem Login zugeordnetes Kennwort

Mit dem Zeichen <> abgegrenzte Strings dürfen keine Leerstellen enthalten.

Beispiel:

```
amexpl32 -verbose -script:Vermibm.scx -cnxf:AllgDatenbank -login:Gerald
-password:Kennwort
```



# 10 | Skripte

## KAPITEL

In diesem Kapitel wird das Arbeiten mit Skripten erklärt.

## Definition eines Skripts

### Allgemeine Informationen

Unter einem Skript ist ganz allgemein ein Programm zu verstehen, das in einer bestimmten Programmiersprache geschrieben wurde. In AssetCenter sind drei Arten von Skripten zu unterscheiden:

- Prozedurenskripte. Dazu gehören:
  - Berechnungsskripte: Dabei handelt es sich um BASIC-Programme, die u. a. zum Berechnen von Feldwerten und Festlegen von Objekteigenschaften der AssetCenter-Datenbank verwendet werden.
  - BASIC-Skripte: Diese Skripte ermöglichen das Ausführen von Aufgaben, insbesondere im Rahmen von Aktionen.



BASIC-Programme können Funktionen verwenden. Diese Art von Skripten wird im Folgenden beschrieben.

---

- Beschreibungsskripte: Es handelt sich dabei um Skripte zum Import und Export von Objekten der AssetCenter-Datenbank, für die eine vollwertige Beschreibungssprache verwendet wird, die sich von BASIC unterscheidet. Ausführliche Informationen zu diesen Skripten finden Sie im Handbuch "Referenzhandbuch: Verwaltung und erweiterte Funktionen" in den Kapiteln "Import von Daten" und "Export von Daten und Verwaltung von SQL-Ansichten".
- "Gemischte" Skripte (Prozedur und Beschreibung). Diese Art von Skripten kommt in den AssetCenter-Assistenten zur Anwendung.

## Informationen zu BASIC

Das in AssetCenter verwendete BASIC besteht aus mehreren Komponenten, die mit "Visual Basic for Applications™" von Cypress kompatibel sind. Wir empfehlen Ihnen, zusätzliche Informationen zur Sprache, Struktur und Syntax im Handbuch der entsprechenden BASIC-Variante nachzulesen.

So werden beispielsweise nur bestimmte Funktionen von "Visual Basic for Applications™" unterstützt. Dazu gehören:

- Funktionen zum Zugriff auf nicht unterstützte Dateien.
- Die Funktionen für Datum und Uhrzeit werden nur begrenzt unterstützt. Dies gilt insbesondere unter UNIX.
- Die Steuerelemente von "Visual Basic for Applications™" sind nicht verfügbar.



Um die Programmierhinweise zu einer BASIC-Funktion oder einem -Schlüsselwort aufzurufen, setzen Sie den Cursor auf das entsprechende Wort und drücken auf die Taste F1. Die Direkthilfe wird angezeigt.

---



## Schreibweise für den Zugriff auf Daten

Die Syntax des von AssetCenter verwendeten BASIC ähnelt der Standardsyntax. Eine Ausnahme bildet die Schreibweise für den Zugriff auf die Daten über einen aktuellen Datensatz. In diesem Fall gilt:

```
[Verknüpfung.Verknüpfung.Feld]
```

Beispiel (von der Tabelle der Produkte aus):

```
[Category.FullName]
```



Anmerkung:

Unter Verwendung der nachstehenden Syntax können Sie den Identifizierungsschlüssel einer Verknüpfung ermitteln:

```
[Verknüpfung.Verknüpfung]
```

Wenn Sie einen Verweis auf eine Verknüpfung herstellen möchten, können Sie sowohl den SQL-Namen einer Verknüpfung als auch den Schlüssel der Verknüpfung verwenden.

Beispiel:

```
RetVal=[Contact.Location] oder RetVal=[Contact.lLocaId]
```

Beide Beispiele führen zu demselben Ergebnis, nämlich der Kennung der Verknüpfung.

## Anwendungsbereiche von Skripten

AssetCenter bietet Ihnen die Möglichkeit der Verwendung von BASIC-Skripten in den nachstehenden Bereichen:

- Zur Konfiguration des Standardwerts für das Feld (Kontextmenü **Objekt konfigurieren**).
- Für den Standardwert eines der Tabelle zugeordneten Merkmals.
- In berechneten Feldern vom Typ "BASIC".
- Zum Konfigurieren von Feldern (Kontextmenü, Option **Objekt konfigurieren** oder AssetCenter Database Administrator):
  - **Standardwert**
  - **Obligatorische Eingabe**
  - **Historie anlegen**

- **Nur Lesen**
- Für die Parameter eines Merkmals, das einer Tabelle zugeordnet ist:
  - **Standardwert** (SQL-Name: DefValScript)
  - **Verfügbar** (SQL-Name: seAvailable)
  - **Anzeige erzwingen** (SQL-Name: seForceDisplay)
  - **Obligatorisch** (SQL-Name: seMandatory)
  - **Historie anlegen** (SQL-Name: seKeepHistory)
- Für Aktionen vom Typ **Skript**:
  - **Aktionsskript** (SQL-Name: Script) einer Aktion vom Typ **Skript**
- In Assistenten:
  - Skripte für den Anfang und das Ende eines Assistenten.
  - Skripte zur Definition der Werte für die Eigenschaften eines Knotens.
- In berechneten Feldern vom Typ "BASIC".
- Im Workflow:
  - Auf der Ebene von Workflow-Aktivitäten vom Typ **Test / Skript**.
  - Auf der Ebene von Workflow-Ereignissen vom Typ **Bei Datenbankänderung**.
  - Auf der Ebene der Bearbeiter vom Typ **Berechnet** der Tabelle amWfOrfRole.

## Einführung in die Funktionen

Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

- Definition einer Funktion
- Vordefinierte und programmierbare Funktionen
- Typ der Funktionen und Funktionsparameter

## Definition einer Funktion

Eine Funktion ist ein Programm, das bestimmte Operationen ausführt, und dem Benutzer einen Wert zurückgibt. Dieser Wert wird "Rückgabewert" genannt.

Eine Funktion weist die folgende Struktur auf:

```

Funktion <Name der Funktion> (<Parameter> As <Typ des Parameters>[, ..
., <Parameter> As <Typ des Parameters>]) As <Typ der Funktion>

<Von der Funktion ausgeführtes Programm (Skript). Dieses Programm muss
den Rückgabewert definieren.>

End Function

```

Diese Struktur gilt sowohl für vordefinierte als auch für programmierbare Funktionen.

## Vordefinierte und programmierbare Funktionen

Die vordefinierten Funktionen und die programmierbaren Funktionen stellen die beiden großen, unter AssetCenter verfügbaren Funktionsfamilien dar.

### Vordefinierte Funktionen

Vordefinierte Funktionen sind mit Programmen vergleichbar, die für den Benutzer geschrieben wurden. Diese Programme führen die verschiedensten Aufgaben durch (Berechnungen, Konvertierung der vom Benutzer gelieferten Informationen) und geben ein Ergebnis zurück. Der Benutzer muss die Funktion nur über ihren Namen aufrufen und ggf. die zur Rückgabe eines Ergebnisses erforderlichen Informationen eingeben. Die vom Benutzer gelieferten Informationen werden Parameter genannt.

Ein Beispiel: Die Funktion **AmConvertCurrency()** führt die Umrechnung eines in der Währung A ausgedrückten Betrags in die Währung B durch, und verwendet dabei einen Wechselkurs, der zu einem bestimmten Datum festgelegt wurde. In unserem Beispiel bedeutet das:

- Der Name der Funktion lautet **AmConvertCurrency**
- Die Parameter, die der Benutzer eingeben muss, entsprechen:
  - Währung A
  - Währung B
  - Umzurechnender Betrag
  - Das Datum, an dem die Umrechnung erfolgen soll, und mit dem der Wechselkurs festgelegt wird

Diese Funktion führt die Umrechnung durch und gibt den Wert zurück, der dem Ergebnis der Berechnung entspricht.

## Programmierbare Funktion

Die programmierbaren Funktionen sind Programme, die der Benutzer selbst schreiben muss. Dabei gibt der Benutzer ausdrücklich den Wert an, der in der Variablen **RetVal**, auch "Rückgabewert" genannt, von der programmierbaren Funktion in der nachstehenden Form zurückgegeben werden muss:

```
RetVal=<Ausdruck>
```



Anmerkung:

AssetCenter lehnt es ab, das Skript einer programmierbaren Funktion zu kompilieren, wenn der Rückgabewert nicht ausdrücklich angegeben ist.


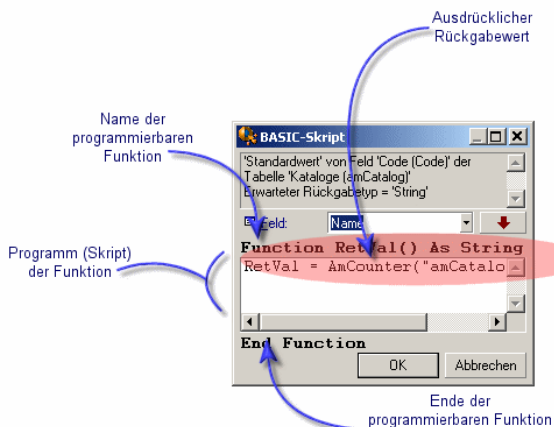
Klicken Sie auf die Schaltfläche  eines Felds, für das ein Skript erstellt werden kann, um über den daraufhin erscheinenden Skripteditor auf die programmierbaren Funktionen zuzugreifen. Der Skripteditor unterstützt den Benutzer beim Erstellen des Programms für eine bestimmte Funktion. Die nachstehende Abbildung zeigt die Eingabemaske zum Schreiben von programmierbaren Funktionen:

Abbildung 10.1. Skript - Editor



In der oberen Hälfte des Skripteditors erscheint eine Beschreibung der programmierbaren Funktion. Dieser Beschreibung entnehmen Sie das von der

Funktion betroffene Objekt, beispielsweise den Standardwert des Felds **Strichcode** (SQL-Name: BarCode) der Tabelle der Vermögen und Lose sowie den Typ des erwarteten Rückgabewerts (unter Verwendung desselben Beispiels wie zuvor: "String").

## Typ der Funktionen und Funktionsparameter

### Typ der Funktionen

Der Typ einer vordefinierten Funktion entspricht dem Typ eines von der Funktion zurückgegebenen Werts. Merken Sie sich diese Tatsache gut, da sie die Ursache von Fehlern bei der Kompilierung und Ausführung von BASIC-Skripten darstellen kann.

Ein Beispiel: Sie können keine Funktion verwenden, die einen Wert eines bestimmten Typs in die Definition des Standardwerts eines Feld mit einem anderen Typ zurückgibt. Versuchen Sie beispielsweise, dieses Skript für einen Standardwert einem beliebigen Feld vom Typ "Date" bzw. "Date and time" zuzuordnen:

```
RetVal=AmLoginName ()
```

Die Funktion **AmLoginName()** gibt den Namen des Benutzers zurück, der in Form einer Zeichenfolge vom Typ "String" mit der Datenbank verbunden ist. Das Format dieses Rückgabewerts ist daher nicht kompatibel mit dem Format eines Felds vom Typ "Date". AssetCenter zeigt folgerichtig eine Fehlermeldung an.

### Typ der Parameter

Die in den vordefinierten Funktionen verwendeten Parameter verfügen ebenfalls über einen Typ, der zur einwandfreien Ausführung der Funktion unbedingt berücksichtigt werden muss. Wenn es in bezug auf den Typ des Parameters zu einem Fehler kommt, zeigt AssetCenter bei der Ausführung der Funktion eine Fehlermeldung an.

### Liste der Typen

Der nachstehenden Tabelle entnehmen Sie die Liste der möglichen Typen für eine Funktion oder einen Parameter:

**Tabelle 10.1. Funktionen/Parameter - Typen**

Typ	Bedeutung
Integer	Ganze Zahl von -32 768 bis +32 767.
Long	Ganze Zahl von -2 147 483 647 bis +2 147 483 646.
Double	Gleitkommazahl mit 8 Byte.
String	Text, in dem sämtliche Zeichen verwendet werden können.
Date	Datum bzw. Datum/Uhrzeit.
Variant	Allgemeiner Typ, der jeden beliebigen Typ darstellen kann.

## Typ des Rückgabewerts für eine programmierbare Funktion festlegen

Bevor Sie ein Skript bearbeiten können, müssen Sie die Funktion und den dieser Funktion zugeordneten Rückgabewert festlegen. Diese beiden Funktionen erscheinen in sämtlichen Fenstern mit dem Titel "BASIC-Skript" fett gedruckt, und zwar in folgendem Format:

```
Funktion <Name der Funktion>() As <Typ der Funktion>
```

Die am häufigsten verwendeten Typen für programmierbare Funktionen lauten "Boolean", "Integer" und "String":

- Funktionen vom Typ "Boolean" (Boolesch) können nur die Werte "TRUE" oder "FALSE" annehmen. Jeder andere Wert führt bei der Kompilierung eines BASIC-Skripts zu einem Fehler.
- Wenn es sich um eine Funktion vom Typ "Integer" (Ganze Zahl) handelt, kann sie nur den Wert einer ganzen Zahl annehmen, z. B. 0, 1, 8, 12.
- Wenn es sich um eine Funktion vom Typ "String" (Zeichenfolge) handelt, kann sie nur den Wert einer in Anführungszeichen gesetzten Zeichenfolge, wie z. B.: "Gebäude21" annehmen.



### Anmerkung:

Wenn Sie den Funktionstyp nicht beachten, kann es bei der Kompilierung des BASIC-Programms zu Fehlern kommen. Notieren Sie sich grundsätzlich den Typ der Funktion, an der Sie gerade arbeiten.

Mit Hilfe des Namens und des Typs der Funktion können Sie den Rückgabewert feststellen, den Sie über ein Skript im folgenden Format angeben müssen:

```
RetVal=<Ausdruck unter Berücksichtigung des Typs der Funktion>
```

## Klassifizierung der BASIC-Funktionen

Das BASIC, das zum Schreiben von Skripten verwendet wird, greift auf Funktionen unterschiedlicher Klassen zurück:

- Herkömmliche BASIC-Funktionen, die dem Standard "Visual Basic pour Applications TM" entsprechen.
- Allgemeine AssetCenter spezifische Funktionen, die in der gesamten Software verwendbar sind.
- Spezifische Funktionen, die in bestimmten Teilen der Software verwendbar sind.

## Erste Schritte zum Schreiben eines Skripts

In diesem Kapitel beschreiben wir die Funktionsweise von Skripten anhand eines Anwendungsbeispiels:

- Beispielszenario
- Schritt 1 - Erstellen des Merkmals Lernprogramm
- Schritt 2 - Öffnen des Bearbeitungsfensters
- Schritt 3 - Analysieren und Schreiben des Algorithmus
- Schritt 4 - Schreiben des BASIC-Programms
- Schritt 5 - Testen des BASIC-Programms

### Beispielszenario

#### Ziel

Das Merkmal "Lernprogramm" soll nur für das Modell "Rechner/ Systemplatine" und die ihm untergeordneten Elemente verfügbar sein.

#### Vorgehensweise


Ordnen Sie dem Feld **Verfügbar** (SQL-Name: seAvailable) der Parameter des Merkmals "Lernprogramm" ein BASIC-Skript zu.

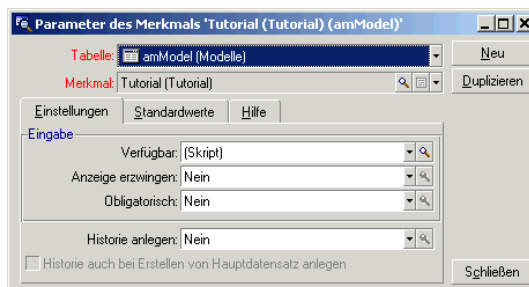
## Schritt 1 - Erstellen des Merkmals Lernprogramm

Öffnen Sie den Bildschirm zur Erstellung eines Merkmals über das Menü **Verwaltung/Merkmale**, und klicken Sie dann auf **Neu**, um ein neues Merkmal zu erstellen. Die Informationen für das Merkmal entnehmen Sie bitte der nachstehenden Tabelle:


Name des Feld	Wert für das Feld
<b>Bez.</b> (SQL-Name: TextLabel)	"Lernprogramm"
<b>SQL-Name</b> (SQL-Name: SQLName)	"Lernprogramm" (In diesem Beispiel nicht verwendet)
<b>Eingabeart</b> (SQL-Name: seDataType)	<b>Numerisch</b>

Klicken Sie auf **Erstellen**, um das Merkmal zu erstellen.

Setzen Sie den Cursor auf die Registerkarte **Parameter**, und klicken Sie auf , um die Parameter des Merkmals "Lernprogramm" zu bearbeiten. Geben Sie die erforderlichen Informationen auf der Registerkarte **Einstellungen** ein, wie in der nachstehenden Abbildung zu sehen ist.



## Schritt 2 - Öffnen des Bearbeitungsfensters

Wählen Sie auf der Registerkarte **Einstellungen** im Detailbereich mit den Parametern des Merkmals "Lernprogramm" den Wert "(Skript)" für das Feld **Verfügbar** (SQL-Name: seAvailable), und klicken Sie auf die Schaltfläche . AssetCenter öffnet daraufhin das Fenster zur Bearbeitung eines Skripts.



## Schritt 3 - Analysieren und Schreiben des Algorithmus


Der Algorithmus muss die nachstehenden Aufgaben übernehmen:


- Änderung des Werts im Feld **Verfügbar** (SQL-Name: seAvailable) in **Ja**, wenn das Modell oder eines der ihm untergeordneten Elemente "/Rechner/Systemplatine" entspricht.
- In allen anderen Fällen Beibehaltung des Werts **Nein** im Feld **Verfügbar**.

Unser Algorithmus schreibt sich wie folgt:

```
Wenn der vollständige Modellname mit "/Rechner/Systemplatine" beginnt,
dann ist
Das Merkmal verfügbar
Ansonsten ist
Das Merkmal nicht verfügbar
```

Damit bestimmt der Wert im Feld **Vollständiger Name** (SQL-Name: FullName) aus der Tabelle der Modelle den Wert im Feld **Verfügbar** des Merkmals. Nur das letzte Feld kommt in unserem Algorithmus zur Anwendung.

Klicken Sie neben dem Feld **Verfügbar** auf die Schaltfläche , um mit der Bearbeitung des BASIC-Skripts zu beginnen. Sie finden das Feld **Vollständiger Name** (SQL-Name: FullName) in der Tabelle der Modelle über die Dropdown-Liste im Bearbeitungsfenster.

Sobald Sie das Feld ausgewählt haben, klicken Sie auf die Schaltfläche , um es in das Bearbeitungsfenster zu übertragen.

## Schritt 4 - Schreiben des BASIC-Programms

Jetzt brauchen Sie nur noch das Programm in dem für diesen Zweck vorgesehenen Fenster zu schreiben. Es entspricht dem in Schritt 3 geschriebenen Algorithmus, nur dass es hier in BASIC übersetzt wird.

```
If Left([FullName], Len("/Rechner/Systemplatine/"))="/Rechner/Systempl
atine/" Then
    RetVal=1
Else
    RetVal=0
End If
```



 **Anmerkung:**

Die Groß- und Kleinschreibung hat keinerlei Auswirkung auf die Bearbeitung und Kompilierung des Skripts.

Klicken Sie auf **OK**, um das Skript zu bestätigen.

## Schritt 5 - Testen des BASIC-Programms

Mit diesem Kontrollschritt vergewissern Sie sich der einwandfreien Funktionsweise des Skripts.

- 1 Wählen Sie das Menü **Ausrüstung/ Modelle**, um die Tabelle der Modelle zu öffnen, und erstellen Sie durch Klicken auf die Schaltfläche **Neu** ein neues Modell.
- 2 Füllen Sie nur die obligatorischen Felder aus:
  - 1 **Name**
  - 2 **UntermodeLL von** (SQL-Name: Parent) mit "Computer/Systemplatine"
  - 3 **Art** (SQL-Name: Nature)
  - 4 **Strichcode** (SQL-Name: BarCode)
- 3 Erstellen Sie das neue Modell durch Klicken auf **Erstellen**.
- 4 Wählen Sie jetzt die Registerkarte **Merkmale**, und fügen Sie durch Klicken auf die Schaltfläche  ein Merkmal hinzu. Auf dem Auswahlbildschirm erscheint der Merkmalname, für den Sie das Skript bearbeitet haben.
- 5 Ändern Sie jetzt den Wert des Felds **UntermodeLL von** in "/Rechner/", und bestätigen Sie die Änderung durch Klicken auf **Ändern**.
- 6 Wählen Sie die Registerkarte **Merkmale** im Modelldetail, und klicken Sie auf die Schaltfläche , um ein Merkmal hinzuzufügen. Der Name des Merkmals, für das Sie das Skript bearbeitet haben, erscheint nicht mehr im Auswahlfenster.

Das Skript erfüllt seine Funktion.

## Skriptbibliotheken

Um den Zugriff auf die Skripte zu zentralisieren, ermöglicht AssetCenter das Speichern von Skriptbibliotheken.

Der Zugriff auf Skriptbibliotheken erfolgt über das Menü **Verwaltung/ Skriptbibliotheken**.

Die gespeicherten Bibliotheken werden über den API-Befehl **amEvalScript** aufgerufen.

Weitere Informationen zum API **amEvalScript** finden Sie im Handbuch "Programmierreferenz", Abschnitt "Alphabetische Referenz".

## Konzepte

In AssetCenter werden Funktionen durch Skripte beschrieben.

Die Erstellung einer Skriptbibliothek kann mit der Definition einer Gruppe von Funktionen verglichen werden.

## Skriptbibliotheken erstellen

So erstellen Sie eine Skriptbibliothek:

- 1 Öffnen Sie die Liste der Skriptbibliotheken.
- 2 Geben Sie im Feld **Name** den Namen der Bibliothek an.
- 3 Geben Sie das Skript in das Feld **Skripte** ein.
- 4 Bestätigen Sie den Vorgang, indem Sie auf **Erstellen** klicken.

Dazu ein Beispiel. Um eine Bibliothek mit dem Namen "biblio" zu erstellen, geben Sie das folgende Skript ein:

```
function FullName(strName As String, strFirstName As String) As String
    FullName = strFirstName & ", " & strName
end function
```

Die Funktion gibt einen String zurück, der aus dem Namen und dem Vornamen einer Person besteht.



**WARNUNG:**

Jede erstellte Funktion muss einen eigenen Namen für alle erstellten Bibliotheken und Skripte aufweisen.

---

## In Skriptbibliotheken gespeicherte Skripte aufrufen

Um ein Skript aus einer Bibliothek abzurufen, müssen Sie mehrere Parameter festlegen: den Namen der Bibliothek, die im Skript definierte Funktion und die entsprechenden Funktionsparameter.

Sie erstellen beispielsweise eine Aktion vom Typ Skript, mit dem Namen "callEvalScript", die die zuvor erstellte Bibliothek verwendet:

- 1 Geben Sie im Feld **Kontext** (SQL-Name: ContextTable) die Tabelle der Personen (SQL-Name: amEmplDept) an.
- 2 Geben Sie auf der Registerkarte **Skript** das folgende Skript ein:

```
Dim strFullName As String
strFullName = amEvalScript("biblio", "FullName", "", [Name], [First
Name])
amMsgBox (strFullName)
```

Das Skript ruft die Funktion "FullName" der Bibliothek "biblio" auf und gibt den Namen und Vornamen der Person in einem Dialogfeld aus.

- 3 Bestätigen Sie den Vorgang, indem Sie auf **Erstellen** klicken.



#### Anmerkung:

Der normalerweise mit dem API **amEvalScript** eingesetzte Kontextparameter, wird nicht verwendet, wenn eine Skriptbibliothek aufgerufen wird.

## Hinweise und Tipps

Im den folgenden Abschnitt finden Sie einige Hinweise und Tipps, die Sie zum schnellen und effizienten Schreiben Ihrer Berechnungsskripte unbedingt beachten sollten.

### Hinweise zum Arbeiten mit programmierbaren Funktionen

Im Folgenden finden Sie einige Hinweise, die Sie beim Schreiben Ihrer Skripte beachten sollten:

- Sowohl programmierbare Funktionen, als auch Funktionen, mit denen der Standardwert eines Felds oder einer Verknüpfung definiert wird, dienen der Bestimmung des Rückgabewerts der Funktion. Dementsprechend raten wir Ihnen dringend davon ab, andere Operationen innerhalb einer programmierbaren Funktion auszuführen. Sie laufen dabei Gefahr, die Leistungsfähigkeit ihrer Datenbank zu beeinträchtigen, oder die Datenbank sogar zu beschädigen.
- In AssetCenter kommen die programmierbaren Funktionen regelmäßig zur Anwendung. Versuchen Sie, Ihre Skripte so weit wie möglich zu optimieren, um die Leistungsfähigkeit von AssetCenter aufrechtzuerhalten.

## Format der Konstanten vom Typ Datum/Uhrzeit in den Skripten

Für die Darstellung der Daten in den Skripten gilt der internationale Standard, und zwar unabhängig von den Anzeigeeoptionen, die vom Benutzer eingestellt wurden:

**yyyy/mm/dd hh:mm:ss**

Beispiel:

```
RetVal="2001/07/12 13:05:00"
```



**Anmerkung:**

Der Bindestrich ("-") kann ebenfalls als Trennzeichen für Datumsangaben verwendet werden.

## Datum in BASIC

In BASIC kann ein Datum entweder im internationalen Format oder in Form einer Zahl vom Typ "Double-precision number" ausgedrückt werden. In diesem Fall stellt die Zahl links vom Komma die Anzahl der Tage dar, die seit dem 30/12/1899 um Mitternacht vergangen sind, während die Zahl rechts vom Komma der Anzahl der verstrichenen Stunden des aktuellen Tages entspricht.

## Datum unter UNIX

Ein Datum werden in BASIC und unter UNIX unterschiedlich geschrieben:

Unter UNIX werden die Daten in Form einer Zahl vom Typ "Integer (32 Bit)" ausgedrückt, mit der die Anzahl der Sekunden dargestellt wird, die seit dem 01/01/1870 um Mitternacht vergangen sind. Den Zeitzone (GMT-Zeit) wird dabei keine Rechnung getragen.

## Format der Konstanten vom Typ Dauer in den Skripten

In den Skripten wird die Dauer in Sekunden gespeichert und ausgedrückt. Ein Beispiel: Wenn Sie für den Standardwert eines Felds vom Typ "Duration" den Wert 3 Tage festlegen wollen, müssen Sie das nachstehende Skript verwenden:

```
RetVal=259200
```

Auch die Funktionen, die eine Dauer berechnen, wie z. B. die Funktion **AmWorkTimeSpanBetween()**, geben ein Ergebnis in Sekunden aus.

 **Anmerkung:**

Bei Konvertierungen geht AssetCenter davon aus, dass ein Jahr 12 Monaten und ein Monat 30 Tagen entspricht (Also gilt: 1 Jahr = 360 Tage).

## Lesen- und Schreibzugriff auf den Wert einer Systemaufzählung

AssetCenter verwaltet die Systemaufzählungen durch Zuordnung einer ganzen Zahl zu jedem möglichen Wert der Aufzählung.

Nehmen wir beispielsweise die Systemaufzählung für das Feld **Situation** (SQL-Name: seAssignment) im Bereich **Standardzuordnung** auf der Registerkarte **Allgemein** Vermögensdetail.

In der nachstehenden Tabelle finden Sie die Werte der Aufzählung:

Werte der Aufzählung	Dem Wert zugeordnete ganze Zahl
In Betrieb	0
Auf Lager	1
Abgang	2
Ausstehende Lieferung	3

Zum Definieren des Standardwerts einer Aufzählung brauchen Sie dann nur noch folgendes zu tun:

- 1 Identifizieren Sie die ganze Zahl, die dem gewünschten Wert entspricht.
- 2 Bearbeiten Sie die nachstehende Zeichenfolge:

```
RetVal=<Dem gewünschten Wert zugeordnete ganze Zahl>
```

Wenn der Standardwert, wie in unserem Beispiel, zum Abrufen des Werts **Ausstehende Lieferung** in das Feld **Situation** dienen soll, müssen Sie die Zeichenfolge wie folgt bearbeiten:

```
RetVal=3
```

 **Anmerkung:**

Eine Systemaufzählung ist nicht mit einer vom Benutzer definierten geschlossenen Aufzählung zu verwechseln.

 Anmerkung:

Eine vollständige Liste der Systemaufzählungswerte finden Sie in der Datei **database.txt** im Unterverzeichnis **doc\infos** des AssetCenter-Installationsordners. Die Spalten "Data display and entry type" und "Additional information on data display and entry type" beschreiben den Typ der Aufzählung bzw. die von der Aufzählung abgerufenen Werte.

## Die virtuelle Verknüpfung "CurrentUser"

### Definition

Der Ausdruck "CurrentUser" kann als Verknüpfung betrachtet werden, die von allen Tabellen ausgeht und auf die Tabelle der Abteilungen und Personen zeigt, die dem aktuellen Benutzer entsprechen.

- In der Form "CurrentUser" zeigt die Verknüpfung auf den Datensatz mit dem aktuellen Benutzer und gibt die Beschreibungsfolge für die Tabelle der Abteilungen und Personen zurück.
- In der Form "CurrentUser.<Champ>" gibt die Verknüpfung den Wert des Felds für den aktuellen Benutzer zurück.

 Anmerkung:

Diese virtuelle Verknüpfung wird nicht in der Liste der Felder und Verknüpfungen geführt, und ist damit auch nicht direkt über den Skripteditor verfügbar. Sie müssen diesen Ausdruck manuell eingeben.

### Äquivalenzen

Die Funktionen **AmLoginName()** und **AmLoginId()**, die respektive den Namen und die Feld-Id des aktuellen Benutzers enthalten, können als von "CurrentUser" abgeleitete Funktionen betrachtet werden. Dabei gelten nachstehende Äquivalenzen:

- AmLoginName()=[CurrentUser.Name]
- AmLoginId()=[CurrentUser.lPersId]

## Beschränkungen

**CurrentUser** kann nur verwendet werden, wenn ein Kontext (d. h. eine Tabelle) definiert ist.

Ist kein Kontext vorhanden, müssen Sie eine andere Funktion einsetzen.

Beispiel:

Sie möchten eine nicht kontextbezogene Aktion erstellen, die eine Datei ausführt, deren Zugriffspfad von dem mit der AssetCenter-Datenbank verbundenen Benutzer abhängt.

Läge ein Kontext vor, hätten Sie eine Aktion des Typs **Programm** erstellen können, deren Feld **Ordner** (Folder) beispielsweise den folgenden Wert aufweisen würde: `c:\scripts\[CurrentUser.Name]\`.

Wenn eine Aktion vom Typ **Programm** keinen Kontext aufweist, wird `[CurrentUser.Name]` als fester Text interpretiert.

In diesem Fall müssen Sie daher eine andere Lösung finden. Sie könnten beispielsweise eine nicht kontextbezogene Aktion vom Typ **Skript** erstellen:

```
RetVal = amActionExec("program.exe", "c:\scripts\" + amLoginName())
```

## Kommentare zu einem BASIC-Skript eingeben

Es ist mitunter sinnvoll, BASIC-Skripte zu kommentieren, um die vom Skript ausgeführten Operationen im Klartext zu beschreiben oder um einem Benutzer das Verstehen und Ändern des Skripts zu erleichtern. AssetCenter kennzeichnet Kommentare mit einem Apostroph ('). Alle Zeichen, die auf das Apostroph folgen und auf derselben Zeile stehen, werden vom Compiler als Kommentare erkannt und ignoriert. Dabei sind zwei Situationen zu unterscheiden:

- Der Kommentar füllt die gesamte Zeile des BASIC-Skripts aus (siehe nachstehendes Beispiel). In diesem Fall sind keine weiteren Maßnahmen nötig.

```
' Wir testen hier den Wert des Felds BarCode der Tabelle der Vermögen und Lose
' Wenn dieser Wert PC1 entspricht, gilt für den Rückgabewert TRUE
If [BarCode]="PC1" Then
RetVal=True
End If
```

- Der Kommentar wird am Ende der Zeile eingefügt, die vom BASIC-Compiler interpretiert wird. In diesem Fall muss das Skript durch



einen Doppelpunkt (:) vom Kommentar getrennt werden. Dem Kommentar bleibt weiterhin ein Apostroph vorangestellt.

```
If [BarCode]="PC1" Then : ' Wenn BarCode gleich PC1
RetVal=TRUE : ' Für den Rückgabewert gilt TRUE
End If : ' Ende des Tests
```

## Fehlermeldungen auslösen

Unter Verwendung der Funktion Err.Raise können Sie eine Fehlermeldung auslösen. Die Syntax dieser Funktion lautet:

```
Err.Raise (<Nummer des Fehlers>, <Fehlermeldung>)
```

 Anmerkung:

Wenn die Erstellung bzw. Änderung eines Datensatzes durch den im Feld Gültigkeit der entsprechenden Tabelle erscheinenden Wert ungültig wird, sollte mit Hilfe der Funktion Err.Raise eine Fehlermeldung ausgelöst werden, um den Benutzer davon in Kenntnis zu setzen. Wenn Sie keine Fehlermeldung auslösen, ist es für den Benutzer nicht unbedingt offensichtlich, warum er den Datensatz weder ändern noch erstellen kann.

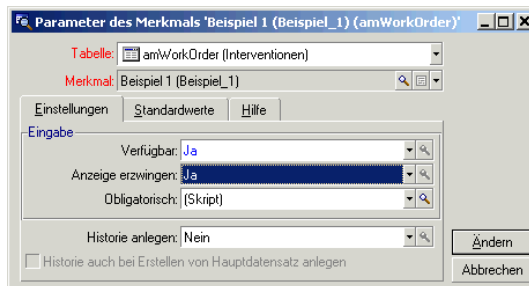
## Erstes Beispiel für ein Skript

In diesem Abschnitt wird ein Problem dargestellt, das mithilfe eines BASIC-Skripts gelöst werden kann. Versuchen Sie, die richtige Lösung zu finden, bevor Sie sie sich ansehen.

## Darstellung des Problems

Ein Merkmal, das wir in unserem Beispiel "Beispiel1" nennen, und das der Tabelle der Interventionen zugeordnet ist, muss unbedingt vor dem Abschluß der Intervention eingegeben werden. Für alle nicht abgeschlossenen Interventionen bleibt die Eingabe dieses Merkmal optional. Im nun folgenden Beispiel gehen wir davon aus, daß das Merkmal erstellt wurde und über einen beliebigen Eingabetyp verfügt, dass das Merkmal der Tabelle der Interventionen

zugeordnet und verfügbar ist und standardmäßig auf dem Bildschirm erscheint, wie in der folgenden Abbildung zu sehen ist:



## Schritt 1 - Analysieren und Schreiben des Algorithmus

Der Algorithmus muss nachstehende Aufgaben erfüllen:

- Änderung des Werts im Feld **Obligatorisch** (SQL-Name: seMandatory) in **Ja**, wenn die Intervention abgeschlossen ist.
- In allen anderen Fällen Beibehaltung des Werts **Nein** im Feld **Obligatorisch**.


Unser Algorithmus schreibt sich wie folgt:

```
Wenn die Intervention abgeschlossen ist, dann ist
Die Eingabe des Merkmals obligatorisch
Ansonsten
Die Eingabe des Merkmals ist nicht obligatorisch
```

Eine Intervention gilt als abgeschlossen, wenn ihr Feld **Status** (SQL-Name: seStatus) den Wert **Abgeschlossen** aufweist.

Damit bestimmt der Wert im Feld **Status** (SQL-Name: seStatus) aus der Tabelle der Interventionen den Wert im Feld **Obligatorisch** des Merkmals. In unserem Algorithmus kommt nur dieses Feld zur Anwendung.

Über die Dropdown-Liste im Bearbeitungsfenster können Sie das Feld **Status** (SQL-Name: seStatus) der Tabelle der Interventionen auswählen.

Sobald Sie das Feld ausgewählt haben, klicken Sie auf die Schaltfläche , um es in das Bearbeitungsfenster zu übertragen.

Die Eingaben in diesem Feld erfolgen über eine Systemaufzählung mit den nachstehenden Werten:

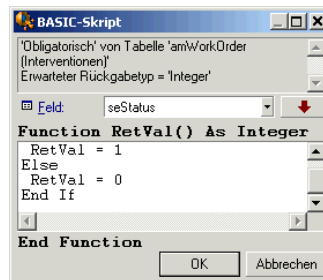
Wert der Aufzählung	Diesem Wert zugeordnete ganze Zahl
Benachrichtigt	0
Geplant	1
In Arbeit	2
Abgeschlossen	3

Der Wert, der uns in dieser Aufzählung interessiert, lautet:

- **Abgeschlossen** und entspricht der "3"

## Schritt 2 - Schreiben des BASIC-Skripts

Jetzt muss der im Schritt 1 definierte Algorithmus nur noch in dem zu diesem Zweck vorgesehenen Fenster übersetzt werden.



Klicken Sie auf **OK**, um das Skript zu bestätigen.

## Schritt 3 - Testen des BASIC-Programms

Mit diesem Kontrollschritt vergewissern Sie sich der einwandfreien Funktionsweise des Skripts.

- 1 Wählen Sie das Menü **Ausrüstung/ Interventionen**, um die Tabelle der Interventionen zu öffnen, und wählen Sie eine Intervention mit dem Status **Abgeschlossen** aus. Wenn keine abgeschlossene Intervention vorhanden ist, können Sie sie auch erstellen.
- 2 Wählen Sie die Registerkarte **Merkmale**. AssetCenter fügt der jeweiligen Intervention ein Merkmal hinzu. Die Definition des Merkmals ist obligatorisch.

- 3 Wählen Sie jetzt eine Intervention, für die der Status nicht den Wert **Abgeschlossen** aufweist, und rufen Sie die Registerkarte **Merkmale** der Intervention auf. Das Merkmal "Beispiel1" wird angezeigt. Die Definition ist jedoch nicht obligatorisch.

Das Skript erfüllt seine Funktion.

## Zweites Beispiel für ein Skript

In diesem Abschnitt wird ein Problem dargestellt, das mithilfe eines BASIC-Skripts gelöst werden kann. Versuchen Sie, die richtige Lösung zu finden, bevor Sie sie sich ansehen.

### Darstellung des Problems

Im Feld **Feld1** (SQL-Name: Field1) des Personendetails sollen standardmäßig der Name und der Vorname der Person erscheinen. Wenn kein Vorname vorhanden ist, soll nur der Nachname angezeigt werden.

### Schritt 1 - Analysieren und Schreiben des Algorithmus

Der Algorithmus muss nachstehende Aufgaben erfüllen:


- Standardmäßiges Anzeigen von Namen und Vornamen der Person im Feld **Feld1** (SQL-Name: Field1) eines Personendetails, sofern diese Angaben existieren.
- Standardmäßiges Anzeigen des Nachnamens der Person im Feld **Feld1** des Personendetails, wenn kein Vorname vorhanden ist.

Unser Algorithmus schreibt sich wie folgt:

```
Wenn der Vorname einer Person nicht existiert, dann ist
Der Standardwert im Feld "Feld1" der Name der Person
Ansonsten
Der Standardwert im Feld "Feld1" lautet "Name, Vorname"
```

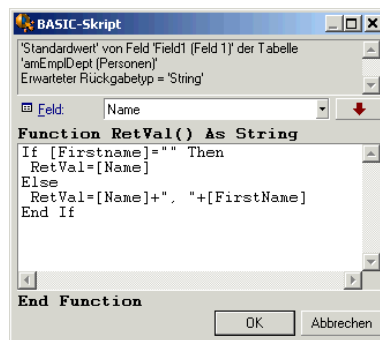
Damit bestimmt der Wert in den Feldern **Name** (SQL-Name: Name) und **Vorname** (SQL-Name: FirstName) aus der Tabelle der Personen den Standardwert im Feld **Feld1** im Personendetail. In unserem Algorithmus kommen nur diese beiden Felder zur Anwendung.

Jetzt brauchen Sie nur noch im Personendetail auf das Feld **Feld1** rechtszuklicken und in dem daraufhin erscheinenden Kontextmenü die Option **Objekt konfigurieren** zu wählen.

Klicken Sie auf die Schaltfläche  neben dem Feld **Standard**, um das BASIC-Skript zu bearbeiten.

## Schritt 2 - Schreiben des BASIC-Skripts

Jetzt muss der im Schritt 1 definierte Algorithmus nur noch in das zu diesem Zweck vorgesehene Fenster übersetzt werden.



Klicken Sie auf **OK**, um das Skript zu bestätigen.

## Schritt 3 - Testen des BASIC-Programms

Mit diesem Kontrollschritt vergewissern Sie sich der einwandfreien Funktionsweise des Skripts.

- 1 Wählen Sie das Menü **Ausrüstung/ Abteilungen und Personen**, um die Tabelle der Personen zu öffnen, und erstellen Sie eine neue Person.
- 2 Geben Sie die erforderlichen Informationen in den Feldern **Name** (SQL-Name: Name) und **Vorname** (SQL-Name: FirstName) ein, und bestätigen Sie die Erstellung durch Klicken auf **Erstellen**. AssetCenter zeigt im Feld **Feld1** (SQL-Name: Field1) den Namen und den Vornamen der Person an.

Das Skript erfüllt seine Funktion.



# 11 | Kalender

## KAPITEL

Der Zugriff auf die Kalender erfolgt über das Menü Extras/ Kalender.

## Allgemeine Übersicht und Funktionsweise eines Kalenders

### Allgemeine Übersicht

Das Detail eines Kalenders setzt sich aus folgenden Elemente zusammen:

- Informationen zur eindeutigen Identifizierung des Kalenders:
  - Der **Name** (SQL-Name: Name) des Kalenders
  - Die **Zeitzone** (SQL-Name: TimeZone), der der Kalender zugeordnet ist
- Beschreibung der allgemeingültigen Arbeitszeiten in einer Woche auf der Registerkarte **Arbeitszeiten**.
- Liste mit den Ausnahmen auf der Registerkarte **Ausnahmen**, und zwar im Vergleich zu den oben genannten Arbeitszeiten.

- Tabelle mit einer Übersicht über die gearbeiteten Stunden innerhalb eines Zeitraums, in der die Informationen aus den Registerkarten **Arbeitszeiten** und **Ausnahmen** berücksichtigt werden.

## Allgemeine Funktionsweise und Verwendung von Kalendern

Die Kalender sind nachstehenden Elementen zugeordnet:

- Den Workflow-Aktivitäten

Mit Hilfe der Kalender legen Sie den Zeitpunkt fest, zu dem die Warnungen, die in den Workflow-Aktivitäten definiert wurden, ausgelöst werden sollen. Der in den Warnungen festgelegte Wert des Felds **Frist** (SQL-Name: tsDelay) wird in gearbeiteten Stunden angegeben.



**WARNUNG:**

Wenn Sie einen Kalender in der Datenbank ändern, treten diese Änderungen für die mit dem Kalender verknüpften AssetCenter-Felder in Kraft, sobald Sie AssetCenter beenden und neu starten.

---

## Auswirkung der Kalender auf bestimmte Funktionalitäten

Die Kalender wirken sich auf bestimmte AssetCenter Funktionalitäten aus. Die Änderung eines Kalenders hat direkt oder indirekt die Änderung bestimmter Datensätze der Datenbank zur Folge. Kalender kommen in folgenden Teilen der Software zur Anwendung:

- In der Ausführungsfrist einer Workflow-Aufgabe.
- In Warnungen, die einer Workflow-Aktivität zugeordnet sind.

## Vorgehensweise beim Erstellen von Kalendern

Gehen Sie zum Erstellen eines Kalenders wie folgt vor:

- 1 Geben Sie dem Kalender einen eindeutigen **Name** (SQL-Namen: Name).
- 2 Ordnen Sie dem Kalender gegebenenfalls eine **Zeitzone** zu, und geben Sie dazu die erforderlichen Informationen im Feld **Zeitzone** (SQL-Namen: TimeZone) ein.



- 3 Definieren Sie die wöchentlichen Arbeitsstunden auf der Registerkarte **Arbeitszeiten** des Kalenderdetails.
- 4 Definieren Sie die Ausnahmen auf der Registerkarte **Ausnahmen** des Kalenderdetails.
- 5 Kontrollieren Sie in regelmäßigen Abständen die einwandfreie Funktionsweise des Kalenders auf der Registerkarte **Vorschau**.

## Beschreibung der Erstellung eines Kalenders

Die Erstellung eines Kalenders erfolgt schrittweise:

- 1 Allgemeine Informationen zur Eingabe des Kalenders
- 2 Eingaben auf der Registerkarte Arbeitszeiten
- 3 Eingaben auf der Registerkarte Ausnahmen eines Kalenders
- 4 Funktionsweise des Kalenders prüfen

## Allgemeine Informationen zur Eingabe des Kalenders

Bevor Sie mit der Eingabe der gearbeiteten Perioden und der entsprechenden Ausnahmen fortfahren, müssen Sie zunächst einmal einen Kalender identifizieren. Geben Sie dazu die erforderlichen Informationen im Feld **Name** (SQL-Name: Name) des Detailfensters ein.

Weiterhin haben Sie über das Feld **Zeitzone** (SQL-Name: TimeZone) die Möglichkeit, einen Kalender einer bestimmten Zeitzone zuzuordnen.

## Eingaben auf der Registerkarte Arbeitszeiten

Die Registerkarte **Arbeitszeiten** im Detailfenster eines Kalenders definiert die dem Kalender zugeordneten Arbeitszeiten einer Woche. Die in dieser Registerkarte eingegebenen Zeitbereiche beschreiben einen Standard, das heißt, dass die gearbeiteten Stunden allgemein in diesem Kalender verwendet werden. Urlaubstage sowie halbe oder ganz freie Tage stellen Ausnahmen dar und werden auf der Registerkarte **Ausnahmen** eingegeben.

## Abbildung 11.1. Kalender - Registerkarte Arbeitszeiten

Tag	Arbeitszeiten
Montag	8:00-12:00;14:00-18:00
Dienstag	8:00-12:00;14:00-18:00
Mittwoch	8:00-12:00;14:00-18:00
Donnerstag	8:00-12:00;14:00-18:00
Freitag	8:00-12:00;14:00-18:00
Samstag	
Sonntag	

Für jeden Tag der Woche lassen sich ein oder mehrere Zeitbereiche definieren, die gearbeitete Perioden darstellen. Sie haben die Wahl zwischen zwei Eingabearten:

- Eine grafische Eingabe mit Hilfe eines gradweisen Reglers, der neben den einzelnen Wochentagen erscheint.
  - 1 Klicken Sie mit der linken Maustaste auf den Regler, und zwar dort, wo der Zeitbereich beginnen soll.
  - 2 Ziehen Sie den Cursor bei niedergedrückter linker Maustaste auf dem Regler bis zu der Stelle, an der der Zeitbereich enden soll. AssetCenter füllt automatisch das Textfeld rechts neben dem gradweisen Regler mit dem entsprechenden Wert aus.
  - 3 Wiederholen Sie diese Prozedur für jeden neuen Eintrag.
- Eine "manuelle" Eingabe über das Textfeld. Bei dieser Art der Eingabe muss eine bestimmte Syntax eingehalten werden:

```
<Anfangsuhrzeit der gearbeiteten Periode>-<Enduhrzeit der gearbeiteten Periode>;<Anfangsuhrzeit der gearbeiteten Periode>-<Enduhrzeit der gearbeiteten Periode>;...
```

Die Stunden erscheinen in nachstehendem Format:

```
<hh:mm [ {AM|PM} ] >
```

Wenn der optionale Parameter [AM|PM] nicht definiert ist, geht AssetCenter standardmäßig davon aus, dass die Uhrzeit im 24-Stunden-Format eingegeben wird.

AssetCenter füllt automatisch den gradweisen Regler links neben dem Textfeld aus.

---

 **Anmerkung:**

Bei der grafischen Eingabe können Sie nur halbe Stunden eingeben. Bei der manuellen Eingabe ist eine minutengenaue Einstellung möglich.

---

## Eingaben auf der Registerkarte **Ausnahmen eines Kalenders**

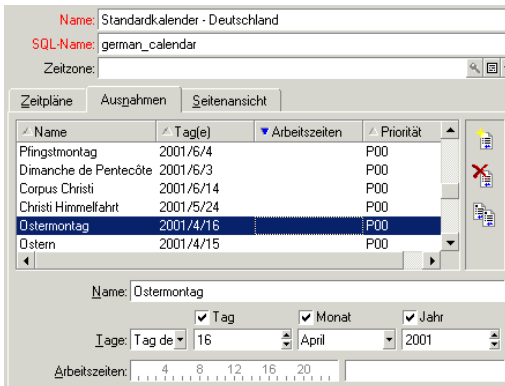
Auf der Registerkarte **Ausnahmen** im Detailfenster eines Kalenders werden die Ausnahmen in bezug auf die gearbeiteten Stunden festgelegt, wie sie auf der Registerkarte **Arbeitszeiten** definiert sind.

### Vorgehensweise




Gehen Sie zum Erstellen einer Ausnahme wie folgt vor:

- 1 Geben Sie der Ausnahme einen "Namen".
- 2 Legen Sie den Anwendungsbereich der Ausnahme mit einer entsprechenden Eingabe im Feld **Tage** fest. Je nach der von Ihnen getroffenen Wahl können Sie den Anwendungsbereich der Ausnahme in bezug auf einen Tag, einen Monat oder ein Jahr definieren.
- 3 Definieren Sie gegebenenfalls mit Hilfe des Felds **Arbeitszeiten** eine gearbeitete Periode innerhalb der Ausnahme. Über dieses Feld können Sie die Ausnahmen präziser festlegen, beispielsweise: "Am letzten Freitag eines jeden Monats arbeitet das Team von 8:30 Uhr bis 10:30 Uhr und von 17:30 Uhr bis 18:30 Uhr".

## Beschreibung der Registerkarte Ausnahmen



Diese Registerkarte ist in zwei Bereiche aufgliedert.

- Im oberen Bereich erscheint eine Liste mit Ausnahmen, über die Sie die Änderungen einer Ausnahme unter Verwendung der Schaltflächen in der Symbolleiste erstellen, duplizieren, löschen, ändern oder abbrechen können:
  -  : Durch Klicken auf diese Schaltfläche erstellen Sie eine neue Ausnahme.
  -  : Durch Klicken auf diese Schaltfläche wird eine Ausnahme gelöscht.
  -  : Durch Klicken auf diese Schaltfläche duplizieren Sie eine Ausnahme.

### Anmerkung:

Die Spalte **Priorität** in diesem Abschnitt des Bildschirms ermöglicht das Sortieren der Ausnahmen unter Berücksichtigung ihrer Priorität: Im Zweifelsfall wird damit die Notwendigkeit einer vorrangigen Bearbeitung für eine Ausnahme festgelegt. AssetCenter ordnet einer Ausnahme automatisch einen entsprechenden Code zu (von "P00" bis "P15"). Je niedriger dieser Code, desto höher die Priorität. Eine Ausnahme mit dem Code "P06" hat dementsprechend Vorrang vor einer Ausnahme mit dem Code "P10".

- Im unteren Bereich erscheinen Detailinformationen.  
Der Wert im Feld **Tag** definiert den Kontext, in dem die Ausnahmen zur Anwendung kommen:

Wert des Felds Tage	Kontext, in dem die Ausnahme zur Anwendung kommt
"Täglich"	Die Ausnahme gilt für alle Tage des Jahres.
"Tag des Jahres"	Die Ausnahme gilt für einen bestimmten Tag oder eine Auswahl von Tagen, die mit Hilfe der Kontrollkästchen <b>Tag</b> , <b>Monat</b> und <b>Jahr</b> festgelegt werden.
"Am ersten"	Die Ausnahme gilt für einen Tag in der Woche, der mit Hilfe des Kontrollkästchens <b>Tag</b> festgelegt wird, und für Monate und Jahre, die mit Hilfe der Kontrollkästchen <b>Monat</b> und <b>Jahr</b> festgelegt werden. <b>Beispiel</b> "Am ersten" Freitag eines jeden Monats.
"Am zweiten"	Die Ausnahme gilt für einen Tag in der Woche, der mit Hilfe des Kontrollkästchens <b>Tag</b> , festgelegt wird, und für Monate und Jahre, die mit Hilfe der Kontrollkästchen <b>Monat</b> und <b>Jahr</b> festgelegt werden. <b>Beispiel</b> "Am zweiten" Montag des Monats September.
"Am vorletzten"	Die Ausnahme gilt für einen Tag in der Woche, der mit Hilfe des Kontrollkästchens <b>Tag</b> , und für Monate und Jahre, die mit Hilfe der Kontrollkästchen <b>Monat</b> und <b>Jahr</b> festgelegt werden. <b>Beispiel</b> "Am vorletzten" Mittwoch des Monats September.
"Am letzten"	Die Ausnahme gilt für einen Tag in der Woche, der mit Hilfe des Kontrollkästchens <b>Tag</b> , und für Monate und Jahre, die mit Hilfe der Kontrollkästchen <b>Monat</b> und <b>Jahr</b> festgelegt werden. <b>Beispiel</b> "Am letzten" Dienstag eines jeden Monats im Jahr 1998.

## Beispiel für die Eingabe einer Ausnahme

Die Mitarbeiter des Unternehmens Taltek haben Anspruch auf Urlaub:

- Der erste Freitag eines jeden Monats ist ein Ruhetag.
- Im August wird nur morgens von 8:30 Uhr bis 12:30 Uhr gearbeitet.

### Regel Nr.1: Der erste Freitag eines jeden Monats ist ein Ruhetag.

- 1 Klicken Sie auf die Schaltfläche **Neu**, um mit der Erstellung einer Ausnahme zu beginnen.
- 2 Die Ausnahme gilt für den ersten Freitag eines jeden Monats, und das für sämtliche Jahre. Die Markierung der Kontrollkästchen **Monat** und **Jahr**

bleibt aufgehoben, was bedeutet, dass die Ausnahme ihre Gültigkeit unabhängig von Monat und Jahr besitzt. Das Kontrollkästchen **Tag** ist markiert, da sich die Ausnahme nur auf den Freitag bezieht.

- 3 Sie beenden die Definition der Ausnahme mit der Zuordnung, in dem Sie dem Feld **Tage** den Wert "Am ersten" zuordnen.

## Regel Nr. 2: Im August wird nur morgens von 8:30 Uhr bis 12:30 Uhr gearbeitet.

- 1 Klicken Sie auf die Schaltfläche **Neu**, um mit der Erstellung einer Ausnahme zu beginnen.
- 2 Diese Ausnahme gilt lediglich für einen Monat, nämlich den Monat August. Die Markierung der Kontrollkästchen **Tag** und **Jahr** ist dementsprechend aufgehoben, das Kontrollkästchen **Monat** ist markiert und verweist auf den Wert "August".
- 3 Die Ausnahme gilt nur für eine Auswahl von Tagen (alle Tage im Monat August). Sie brauchen dem Feld **Tage** also nur noch den Wert "Tag des Jahres" zuzuordnen.
- 4 In dieser Periode wird von 8:30 Uhr bis 12:30 Uhr gearbeitet. Zum Abschließen der Eingabe dieser Ausnahme wählen Sie im Feld **Arbeitszeiten** den Zeitbereich 8:30 Uhr bis 12:30 Uhr.

## Funktionsweise des Kalenders prüfen

Über die Registerkarte **Vorschau** können sämtliche, in den Registerkarten **Arbeitszeiten** und **Ausnahmen** für einen bestimmten Zeitraum definierten Regeln überprüft werden. Verwenden Sie dazu die Felder **Anfangsdatum** und **Enddatum** im Rahmen **Test**.

Abbildung 11.2. Kalender - Registerkarte Vorschau

The screenshot displays a software interface for viewing a calendar. At the top, there are tabs for 'Zeitpläne', 'Ausnahmen', and 'Seitenansicht'. Below these, the 'Test' entry is shown with the following details:

- Anfangsdatum:** 05/03/2002 16:16:23
- Enddatum:** 05/03/2002 16:16:23
- Verstrichene Zeit:** 0s

Below this, a section titled 'Dem Anfangsdatum zugeordneter Kalender' shows a weekly calendar view for the week of March 4-10, 2002. The days are listed with corresponding work blocks:

Tag	Arbeitszeiten
Montag 4 März 2002	9:30-13:00; 14:30-19:00
Dienstag 5 März 2002	9:30-13:00; 14:30-19:00
Mittwoch 6 März 2002	
Donnerstag 7 März 2002	9:30-13:00; 14:30-19:00
Freitag 8 März 2002	9:30-13:00; 14:30-19:00
Samstag 9 März 2002	
Sonntag 10 März 2002	

- Im Rahmen **Dem Anfangsdatum zugeordneter Kalender** erscheint standardmäßig eine Vorschau der gearbeiteten Tage einer Woche, einschließlich des ausgewählten "Anfangsdatum".
- Das Feld **Verstrichene Zeit** zeigt die Gesamtanzahl der gearbeiteten Stunden des ausgewählten Zeitraums.

 **Anmerkung:**

Sie können im Feld **Verstrichene Zeit** eine Dauer eingeben, um die erneute Berechnung des **Enddatum** unter Berücksichtigung des **Anfangsdatum** zu erzwingen.





# 12 | Zeitzonen

## KAPITEL

In diesem Kapitel wird die Verwaltung von Zeitzonen erläutert.

---

 Anmerkung:

Diese Funktion ist nur im Rahmen bestimmter AssetCenter-Lizenzverträge verfügbar.

---

## Vorteile der Verwaltung von Zeitzonen

Es kommt immer häufiger vor, dass sich Server und Clients an geographisch weit auseinanderliegenden Standorten befinden. AssetCenter verwaltet Zeitzonen und Zeitverschiebungen in bezug auf die mittlere Greenwich-Zeit (GMT). AssetCenter beachtet dabei nachstehende Regeln:

- Alle Felder vom Typ "Date and time" werden auf dem Client unter Berücksichtigung der für diesen Rechner geltenden Zeitzone angezeigt.
- Alle Felder vom Typ "Date and time" werden auf dem Server unter Berücksichtigung der für die Datumsangaben definierten Zeitzone gespeichert.

- Alle Berechnungen, bei denen Datumsangaben und Uhrzeiten herangezogen werden, tragen den eventuell vorhandenen Zeitverschiebungen Rechnung.

## Beispiel

Nehmen wir beispielsweise einen Server, der sich in New York befindet und über Daten verfügt, die für die Zeitzone in Paris indexiert sind. Die Clientrechner stehen in Paris bzw. London. Wir definieren die Zeitzone für die einzelnen Rechner und beziehen uns dabei auf den Meridian von Greenwich:

- Zeitzone der Server-Station = GMT-5
- Zeitzone des Clients in Paris = GMT+1
- Zeitzone des Clients in London = GMT
- Zeitzone der Daten = GMT+1

Alle Felder vom Typ "Date and time" sind in der Datenbank des Servers im Format GMT+1 gespeichert und müssen auf dem Client in Paris im Format GMT+1 und auf dem Client in London im Format GMT erscheinen. Ein Beispiel: Sie nehmen einen Anruf auf dem Client in London an. Wenn für das Datum, bis zu dem die Support-Unterlage abgeschlossen sein muss, der 15.05.00 um 17:30 Uhr gilt, erscheint dieses Datum auf den anderen Rechnern wie folgt:

- Auf dem Server: 15.05.00 um 12:30 Uhr
- Auf dem Client in Paris: 15.05.00 um 18:30 Uhr
- Auf dem Client in London: 15.05.00 um 17:30 Uhr

## Zeitzone einrichten

Um eine optimale Verwaltung der Zeitzone durch AssetCenter zu gewährleisten, müssen Sie vorgehen wie folgt:



- 1 Definieren Sie die Zeitzone bei der Erstellung der Datenbank unter AssetCenter Database Administrator mit Hilfe der Option **Zeitzone verwenden**.
- 2 Erstellen Sie die Zeitzone, beispielsweise durch Importieren von Informationen bezüglich der Zeitzone.
- 3 Definieren Sie die für Ihren Rechner gültige Zeitzone über das Menü **Extras/ Zeitzone ändern**.
- 4 Definieren Sie die Kalender in Übereinstimmung mit den geltenden Zeitzone.

## Zeitzone erstellen

Im Gegensatz zu Windows berücksichtigt AssetCenter Änderungen der Regeln beim Wechsel zwischen Sommer- auf Winterzeit im Laufe der Jahre. Auf diese Weise lassen sich die Ortszeiten mit großer Präzision anzeigen. Bei Verwendung der Informationen, die Ihnen von AssetCenter in Bezug auf die Zeitzone geboten werden, bestehen folgende Möglichkeiten:

- Anzeigen von lokalen Uhrzeiten unter Berücksichtigung der Wechsel von der Sommerzeit zur Winterzeit, und umgekehrt
- Anzeigen der Uhrzeiten an anderen Standorten

Im Lieferumfang von AssetCenter ist eine Datei mit der Beschreibung der hauptsächlich verwendeten Zeitzone enthalten, um Ihnen die manuelle Eingabe dieser Zone zu ersparen. Gehen Sie zum Importieren dieser Datei wie folgt vor:

- 1 Wählen Sie das Menü **Datei/ Import**. AssetCenter öffnet den Bildschirm zur Auswahl des Importtyps.
- 2 Wählen Sie "Skript ausführen", indem Sie auf die Schaltfläche  klicken. AssetCenter öffnet den Bildschirm zur Aktualisierung der Datenbank. Klicken Sie auf die Schaltfläche , um das auszuführende Skript zu wählen. Dabei sollte es sich um das Skript **tz.scr** im Unterordner **datasys** des AssetCenter-Installationsordners handeln.
- 3 Klicken Sie auf die Schaltfläche **Importieren**. AssetCenter führt den Import gemäß den im Skript enthaltenen Anweisungen durch.

## Zeitzone verwalten

In diesem Abschnitt gehen wir näher auf das Feld **Übergang (SZ-WZ)** (SQL-Name: memDaylightInfo) ein.

### Format des Felds Übergang (SZ-WZ)

Die Struktur des Felds **Übergang (SZ-WZ)** (SQL-Name: memDaylightInfo) stellt sich dar wie folgt:

```
<Year>=<DaylightInfo>|<Year>
=<DaylightInfo>|<Year>=<DaylightInfo>|...
```

Im nun folgenden Abschnitt gelten nachstehende Konventionen:

- Die Gruppe `<Year>=<DaylightInfo>` wird im folgenden Parameter genannt.
- `<Year>` und `<DaylightInfo>` werden im folgenden Argumente genannt.

Die nachstehende Tabelle gibt einen schnellen Überblick über die Merkmale der Informationen zum Wechsel zwischen der Sommerzeit und der Winterzeit, und zwar unter Berücksichtigung der Werte für die Argumente `<Year>` und `<DaylightInfo>`.

	<b>Das Argument <code>&lt;DaylightInfo&gt;</code> ist leer</b>	<b>Das Argument <code>&lt;DaylightInfo&gt;</code> weist einen Wert auf</b>
<b>Das Argument <code>&lt;Year&gt;</code> ist leer ("<code>&lt;Year&gt;="</code> erscheint nicht)</b>	Für diese Zeitzone liegt kein Wechsel zwischen Sommer- und Winterzeit vor.	Die Informationen zum Wechsel zwischen Sommer- und Winterzeit innerhalb der Zeitzone sind für sämtliche Jahre gültig, mit Ausnahme der Jahre, die von den Parametern mit dem Argument <code>&lt;Year&gt;</code> definiert sind.
<b>Das Argument <code>&lt;Year&gt;</code> verfügt über einen Wert</b>	Nicht anwendbar	Die Informationen zum Wechsel zwischen Sommer- und Winterzeit innerhalb der Zeitzone sind für jedes Jahr ab dem Jahr gültig, das über das Argument <code>&lt;Year&gt;</code> definiert wird, und das bis zum nächsten Argument <code>&lt;Year&gt;</code>

## Werte für das Argument `<Year>`

Mit dem Argument `<Year>` wird das Jahr definiert, ab dem die über das Argument `<DaylightInfo>` festgelegten Informationen zum Wechsel zwischen Sommer- und Winterzeit gültig sind. Dieses Argument kann jeden beliebigen Wert für ein Jahr mit vier Ziffern annehmen. (Bsp: 1990, 1997, 1998, 2012).

## Werte für das Argument `<DaylightInfo>`

Das komplette Format des Arguments `<DaylightInfo>` lautet wie folgt:

```
<StdShift>, <DltShift>, <SDay>
, <SMonth>, <SDayPos>, <SHour>
, <DDay>, <DMonth>, <DDayPos>, <DHour>
```

Dieses Argument wird in mehrere Nebenargumente untergliedert, die in der nachstehenden Tabelle zusammengefasst sind:

Nebenargument	Beschreibung	Mögliche Werte
<StdShift>	<p>Mit diesem Argument wird die Zeitverschiebung von der Winterzeit im Vergleich zu der in der betreffenden Zeitzone gültigen Zonenzeit in Minuten angegeben.</p> <p>Beispiel für Paris (in der Zeitzone GMT+1Std.): Wenn &lt;StdShift&gt; dem Wert 30 (Minuten) entspricht, gilt für die Winterzeit GMT+1Std.30Min. und nicht GMT+1Std.</p>	<p>Standardmäßig gilt für dieses Nebenargument der Wert Null. Das Argument kann jedoch jeden beliebigen Wert annehmen. Der Benutzer ist für die Überprüfung der Kohärenz des Werts dieses Nebenarguments zuständig.</p>
<DltShift>	<p>Mit diesem Argument wird die Zeitverschiebung von der Sommerzeit im Vergleich zu der in der betreffenden Zeitzone gültigen Zonenzeit in Minuten angegeben.</p>	<p>Standardmäßig gilt für dieses Nebenargument der Wert 60, was der Zeitverschiebung von einer Stunde zwischen der Sommerzeit und der Zonenzeit "GMT" + entspricht. Das Argument kann jedoch jeden beliebigen Wert annehmen. Der Benutzer ist für die Überprüfung der Kohärenz des Werts dieses Nebenarguments zuständig.</p>
<SDay>	<p>Tag des Wechsels von der Sommerzeit zur Winterzeit.</p>	<p>"Monday"            "Tuesday"            "Wednesday"            "Thursday"            "Friday"            "Saturday"            "Sunday"</p> <p>Keine möglichen Werte. Sie müssen dem Argument &lt;DDayPos&gt; einen Wert zwischen 1 und 31 geben, um den Tag festzulegen, an dem der Wechsel von der Sommerzeit zur Winterzeit stattfinden soll.</p>

Nebenargument	Beschreibung	Mögliche Werte
<SMonth>	Monat des Wechsels von der Sommerzeit zur Winterzeit.	"January" "February" "March" "April" ... "November" "December"
<SDay-Pos>	Position des Tags innerhalb eines Monats, an dem der Wechsel von der Sommerzeit zur Winterzeit stattfindet.	"First" "Second" "Third" "Fourth" "Last" "Penultimate (vorletzter)" Ein Wert zwischen 1 und 31, wenn <DDay> keinen Wert aufweist.
<SHour>	Uhrzeit des Wechsels von der Sommerzeit zur Winterzeit, ausgedrückt in der Sommerzeit.	Ein beliebiger Wert im Format 24 Stunden (HH:MM:SS).
<DDay>	Tag des Wechsels von der Winterzeit zur Sommerzeit.	"Monday" "Tuesday" "Wednesday" "Thursday" "Friday" "Saturday" "Sunday" Keine möglichen Werte. Sie müssen dem Argument <SDayPos> einen Wert zwischen 1 und 31 geben, um den Tag festzulegen, an dem der Wechsel von der Winterzeit zur Sommerzeit stattfinden soll.

Nebenargument	Beschreibung	Mögliche Werte
<DMonth>	Monat des Wechsels von der Winterzeit zur Sommerzeit.	"January" "February" "March" "April" ... "November" "December"
<DDay-Pos>	Position des Tags innerhalb eines Monats, an dem der Wechsel von der Winterzeit zur Sommerzeit stattfindet	"First" "Second" "Third" "Fourth" "Last" "Penultimate (vorletzter)" Ein Wert zwischen 1 und 31, wenn <SDay> keinen Wert aufweist.
<DHour>	Uhrzeit des Wechsels von der Winterzeit zur Sommerzeit, ausgedrückt in der Winterzeit.	Ein beliebiger Wert im 24-Stunden-Format (HH:MM:SS).

## Beispiel zur Verwaltung der Sommerzeit

In diesem Beispiel nehmen wir die Informationen zur Sommerzeit der Zeitzone "(GMT+01:00) Paris, Madrid, Amsterdam".

```
2000=0,60,Sunday,October,last,03:00:00,Sunday,March,last,02:00:00|
0,60,Sunday,September,last,03:00:00,Sunday,March,last,02:00:00
```

Die verwendeten Parameter lauten:

```
2000=0,60,Sunday,October,last,03:00:00,Sunday,March,last,02:00:00
0,60,Sunday,September,last,03:00:00,Sunday,March,last,02:00:00
```

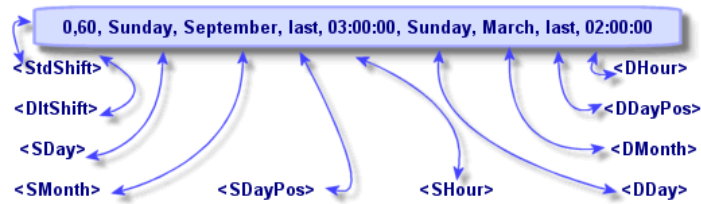
## Erster Parameter



- <Year> = 2000 bedeutet, dass die folgenden Parameter vom Typ <DaylightInfo> ab dem Jahr 2000 gültig sind.
- <StdShift> = 0 bedeutet, dass die Zeitverschiebung zwischen der Winterzeit und der Zonenzeit gleich Null ist. Die Winterzeit entspricht damit die Zonenzeit.
- <DltShift> = 60 bedeutet, dass die Zeitverschiebung zwischen der Sommerzeit und der Zonenzeit 60 Minuten, und damit eine Stunde beträgt. Die Sommerzeit entspricht damit der Zonenzeit plus eine Stunde.
- <SDay> = Sunday besagt, dass der Wechsel zur Winterzeit an einem Sonntag stattfindet.
- <SMonth> = October besagt, dass der Wechsel zur Winterzeit im Oktober stattfindet.
- <SDayPos> = Last gibt die Position des Tags im Monat an. In unserem Beispiel erfolgt der Wechsel zur Winterzeit am letzten Sonntag im Monat Oktober.
- <SHour> = 03:00:00 bedeutet, dass der Wechsel zur Winterzeit um 3 Uhr morgens stattfindet.
- <DDay> = Sunday besagt, dass der Wechsel zur Sommerzeit an einem Sonntag stattfindet.
- <DMonth> = March besagt, dass der Wechsel zur Sommerzeit im März stattfindet.
- <DDayPos> = Last gibt die Position des Tags im Monat an. In unserem Beispiel erfolgt der Wechsel zur Sommerzeit am letzten Sonntag im Monat März.
- <DHour> = 02:00:00 bedeutet, dass der Wechsel zur Sommerzeit um 2 Uhr morgens stattfindet.



## Zweiter Parameter



- Die Abwesenheit des Arguments <Year> für diesen Parameter besagt, dass dieser Parameter nur auf die Jahre anwendbar sind, die im vorigen Parameter nicht beschrieben wurden.
- <StdShift> = 0 bedeutet, dass die Zeitverschiebung zwischen der Winterzeit und der Zonenzeit gleich Null ist. Die Winterzeit entspricht damit die Zonenzeit.
- <DltShift> = 60 bedeutet, dass die Zeitverschiebung zwischen der Sommerzeit und der Zonenzeit 60 Minuten, und damit eine Stunde beträgt. Die Sommerzeit entspricht damit der Zonenzeit plus eine Stunde.
- <SDay> = Sunday besagt, dass der Wechsel zur Winterzeit an einem Sonntag stattfindet.
- <SMonth> = September besagt, dass der Wechsel zur Winterzeit im September stattfindet.
- <SDayPos> = Last gibt die Position des Tags im Monat an. In unserem Beispiel erfolgt der Wechsel zur Winterzeit am letzten Sonntag im Monat September.
- <SHour> = 03:00:00 bedeutet, dass der Wechsel zur Winterzeit um 3 Uhr morgens stattfindet.
- <DDay> = Sunday besagt, dass der Wechsel zur Sommerzeit an einem Sonntag stattfindet.
- <DMonth> = March besagt, dass der Wechsel zur Sommerzeit im März stattfindet.
- <DDayPos> = Last gibt die Position des Tags im Monat an. In unserem Beispiel erfolgt der Wechsel zur Sommerzeit am letzten Sonntag im Monat März.
- <DHour> = 02:00:00 bedeutet, dass der Wechsel zur Sommerzeit um 2 Uhr morgens stattfindet.

Daraus folgt:

---



Anmerkung:

Ab dem Jahr 2000 erfolgt der Wechsel zur Winterzeit am letzten Sonntag im Oktober 03:00:00, und die Uhr wird dementsprechend auf 02:00:00 zurückgestellt. Der Wechsel zur Sommerzeit erfolgt am letzten Sonntag im Monat März um 02:00:00, wobei die Uhr auf 03:00:00 vorgestellt wird.

Für die Jahre vor 2000 erfolgt der Wechsel zur Winterzeit am letzten Sonntag im Monat September um 03:00:00, und der Wechsel zur Sommerzeit erfolgt am letzten Sonntag im Monat März 02:00:00.

---

## Zeitzone in AssetCenter Server verwalten

AssetCenter Server erlaubt das Konfigurieren von Zeitzonentests. Dies erfolgt über das Menü **Optionen/ Module konfigurieren**.

### Tests

Auf der Registerkarte **Allgemein** des Konfigurationsbildschirms konfigurieren Sie den Test, den Sie ausführen möchten:

- Zeitzone des Datenbankservers prüfen
- Ortszeit mit Serverzeit vergleichen

Bei beiden Tests wird die Uhrzeit des Datenbankservers mit der Uhrzeit der Arbeitsstation verglichen, auf der AssetCenter Server installiert ist. Dabei ist zu beachten, dass die Abweichung zwischen den beiden Uhrzeiten wird folgt ausgedrückt wird:  $[(n * 30 \text{ Minuten}) + m]$ , wobei m einem Wert zwischen -15 Minuten und + 15 Minuten entspricht.

### In beiden Fällen

Wenn die Abweichung m 5 Minuten überschreitet, bietet AssetCenter Server die Aktualisierung der lokalen Uhrzeit der Arbeitsstation an, auf der der Server installiert ist.

Wenn Sie diese Aktualisierung ablehnen, weil Sie beispielsweise davon überzeugt sind, dass die Uhrzeit des Servers berichtigt werden muss, wird die Herstellung der Verbindung abgelehnt. Die Verbindung läßt sich erneut herstellen, wenn

die Abweichung zwischen den beiden Uhrzeiten 5 Minuten nicht überschreitet. Das kann der Fall sein, wenn die Uhrzeit des Datenbankservers oder die Uhrzeit der Arbeitsstation, auf der AssetCenter Server installiert ist, oder beide Uhrzeiten aktualisiert wurden.

## Besonderheit der Option Zeitzone des Datenbankservers prüfen



Anmerkung:

Dazu muss die Arbeitsstation, auf der AssetCenter Server läuft, über die genaue Uhrzeit sowie die Funktion zum Umschalten von der Sommer- auf die Winterzeit verfügen.

Die Informationen zu der Zeitzone des Servers werden gegebenenfalls in der Tabelle der AssetCenter Optionen aktualisiert, wenn die Zahl ( $n * 30$  Minuten) nicht der Zeitzone des Servers entspricht.

## Besonderheit der Option Ortszeit mit Serverzeit vergleichen

Sie rufen die Zeitzone, die zum erfolgreichen Ausführen von AssetCenter internen Operationen erforderlich sind, auf der Ebene des Servers ab.

## Häufigkeit der Tests

Die Tests werden

- 1 in erster Linie bei der Herstellung der Verbindung zwischen AssetCenter Server und der Datenbank durchgeführt.
- 2 Je nach dem Programm, das Sie auf dem Bildschirm zur Konfiguration der AssetCenter Server-Module einrichten (**Extras/ Module konfigurieren**), können diese Tests auch häufiger durchgeführt werden.

## Auswirkungen der Zeitzonen auf verschiedene Vorgänge

Die Verwaltung der Zeitzonen wirkt sich auf eine Reihe von Vorgängen in AssetCenter aus:

- Datenbanken erstellen
- Verbindung zu einer Datenbank und Zeitzonen
- Import und Export

## Datenbanken erstellen

Bei der Erstellung einer Datenbank gibt Ihnen AssetCenter die Möglichkeit, die Optionen zur Verwaltung der Zeitzonen festzulegen. Wählen Sie in AssetCenter Database Administrator das Menü **Aktionen/ Datenbank anlegen**. Der Bereich **Systemdaten erstellen** enthält die Optionen zur Definition der Zeitzonen.

Über das Kontrollkästchen **Zeitzone verwenden** legen Sie die Berücksichtigung von Zeitzonen bei der Erstellung von Datenbanken fest.

- Wenn dieses Kontrollkästchen markiert ist, kommen die Zeitzonen bei der Erstellung der Datenbank zur Anwendung.
- Wenn die Markierung dieses Kontrollkästchens aufgehoben ist, werden die Zeitzonen bei der Erstellung der Datenbank ignoriert.

Über die beiden Felder **Serverzeitzone** und **Datenzeitzone** legen Sie zum einen die Zeitzone fest, die zum Zeitpunkt der Erstellung der Datenbank gültig ist,

und zum anderen bestimmen Sie die Zeitzone, die bei der Speicherung von Daten berücksichtigt werden soll.



#### Anmerkung:

Diese Option ist nur bei der Erstellung einer Datenbank aktiv und ermöglicht die Definition der Zeitzone für den Server und die Daten. Wenn Sie diese Werte ändern, haben die Angaben, die für die Felder "Date and time" in der Datenbank gespeichert wurden, keinen Sinn mehr.

## Verbindung zu einer Datenbank und Zeitzonen

Bei der Herstellung einer Verbindung sucht AssetCenter in der Datei **am.ini** nach dem Eintrag "LocalTimeZone", der die Zeitzone des Clientrechners definiert.

Ablageort der Datei: ► Handbuch **AssetCenter - Installation und Aktualisierung**, Kapitel **.ini- und .cfg-Dateien**.

Sollten dort keine Informationen vorliegen, verwendet AssetCenter die für das System festgelegte Zeitzone.

Anschließend versucht AssetCenter die Zeitzone in der Tabelle der Zeitzonen der Datenbank zu finden, die entweder dem Eintrag "LocalTimeZone" in der Datei **am.ini** oder der unter Windows definierten Zeitzone entspricht.

Die nachstehenden Tabelle zeigt die verschiedenen Möglichkeiten:

Vorhandensein des Eintrags "Local-TimeZone" in der Datei am.ini	Vorhandensein einer entsprechenden Zeitzone in der Zeitzonen-tabelle	In der Datei am.ini für den Eintrag "LocalTimeZone" gespeicherte Informationen
Ja	Ja	Zeitzone der Datenbank
	Nein	Unverändert
Nein	Ja	Zeitzone der Datenbank
	Nein	Zeitzone des Systems

## Uhrzeit des Clients anpassen

Bei der Herstellung einer Verbindung zu einer entfernten Datenbank kontrolliert AssetCenter die Gültigkeit der Uhrzeit des Clients im Vergleich zur Server-Station.

Dabei wird die Uhrzeit des Clients mit der Uhrzeit des Datenbankservers verglichen. AssetCenter berechnet die Zeitzone der Uhr des Clients und ermittelt die Zeitverschiebung zwischen den beiden Rechnern. Die Zeitverschiebung wird wie folgt berechnet:

$$\text{Zeitverschiebung} = \text{Modulo}(\text{Unterschied zwischen den Uhrzeiten der beiden Rechner, ausgedrückt in Minuten}) / 30$$

 **Anmerkung:**

Unter Modulo ist der ganze Rest bei einer Teilung zu verstehen.

Ein Beispiel:

- Station A befindet sich in GMT mit der Uhrzeit 18:02
- Station B befindet sich in GMT+1 mit der Uhrzeit 18:19 (also 17:19 Uhr für die Station A, woraus sich ein Unterschied von 43 Minuten im Vergleich zur Station A ergibt).

$$\text{Zeitverschiebung} = \text{Modulo}(43/30) = 13 \text{ Minuten}$$

Wenn die Zeitverschiebung den nicht veränderbaren Wert von fünf Minuten überschreitet, schlägt AssetCenter die Anpassung der Uhrzeit auf dem Client vor.

Sollte der Benutzer diese Anpassung ablehnen, kommt die Verbindung nicht zustande.

AssetCenter führt diese Kontrolle regelmäßig und bei jeder Änderung der Uhrzeit auf dem Clientrechner durch. Standardmäßig erfolgt die Kontrolle alle 60 Minuten. Dieser Wert kann jedoch über die Option **g\_lTimeZoneCheckInMns** in der Datei **am.ini** im Abschnitt [option] geändert werden.

Ablageort der Datei: ► Handbuch **AssetCenter - Installation und Aktualisierung**, Kapitel **.ini- und .cfg-Dateien**.

```
[option]
g_lTimeZoneCheckInMns = 30
```

Im obigen Beispiel wird die Kontrolle einer Zeitverschiebung alle 30 Minuten durchgeführt.

Die Definition der Häufigkeit der Kontrolle kann auch über die Option **Ortszeit mit Serverzeit vergleichen** in den Optionen von AssetCenter erfolgen.

---

 **Anmerkung:**

Diese Kontrollfunktion ist nur aktiv, wenn die Zeitzonen bei der Erstellung der Datenbank berücksichtigt wurden.

---

## Import und Export

Bei diesen beiden Funktionalitäten wird bei der Umrechnung davon ausgegangen, dass sich alle Felder vom Typ "Date and time" auf die Zeitzone des Rechners beziehen, der den Import bzw. den Export durchführt.





# 13 | Berechnete Felder

## KAPITEL

Der Zugriff auf den Bildschirm zur Erstellung der berechneten Felder erfolgt über das Menü **Verwaltung/ Berechnete Felder**.

## Definition eines berechneten Felds

Bei einem berechneten Feld handelt es sich um ein Feld, dessen Wert unter Berücksichtigung des Werts aus anderen Feldern sowie unter Berücksichtigung von Variablen unter Verwendung einer benutzerspezifischen Formel ermittelt wurde. Dabei sind drei Typen von berechneten Feldern zu unterscheiden:

- AQL
- BASIC
- Berechneter String

Für jeden dieser Typen ist eine andere Sprache zum Schreiben der Berechnungsformel für das Feld erforderlich. Zusätzlich bestimmen die Typen die Möglichkeiten und Einschränkungen der Verwendung des Felds. Ein Beispiel: Zusammen mit Filtern können nur die berechneten Felder vom Typ "AQL" verwendet werden.



Berechnete Felder sind virtuelle Felder, die nur über einen Lesezugriff verfügbar sind. Virtuuell bedeutet in diesem Zusammenhang, dass nur die Formel zur Berechnung des Feldwerts einmal in der AssetCenter-Datenbank gespeichert wird. Sie können eine beliebige Anzahl von berechneten Feldern definieren und ihnen entsprechende Nutzungsberechtigungen zuordnen.

## Vorteile der berechneten Felder

Berechnete Felder lassen Sie zusätzliche Informationen definieren und synthetische Informationen für sämtliche Datensätze einer Tabelle der AssetCenter-Datenbank ermitteln. In diesem Sinne ähneln berechnete Felder "herkömmlichen" Feldern der Datenbank, bis auf folgende Ausnahmen:

- Im Gegensatz zu den "herkömmlichen" Feldern wird der Wert der berechneten Felder nicht in der AssetCenter-Datenbank gespeichert.
- Der Wert der berechneten Felder wird nicht vom Benutzer eingegeben, sondern mit Hilfe einer Formel ermittelt.
- Sie können ein berechnetes Feld nicht einem einzigen Feld in einer bestimmten Tabelle zuordnen. Wie die "herkömmlichen" Felder der Datenbank werden die berechneten Felder allen Datensätzen einer Datenbank zugeordnet. Die Felder weisen einen Wert für jeden Datensatz der entsprechenden Tabelle auf, auch wenn dieser Wert Null entspricht.
- Die berechneten Felder erscheinen nicht im Detailbildschirm eines Datensatzes, und können nur in Listenform angezeigt werden.
- Die berechneten Felder können nur dann zur Ermittlung des Standardwerts eines Standardfelds verwendet werden, wenn es sich um ein Feld vom Typ **Berechneter String** oder **BASIC-Skript** handelt.

## Berechnete Felder erstellen

Bevor Sie mit der Erstellung eines berechneten Felds beginnen, sollten Sie sich mit den Besonderheiten der drei verschiedenen Typen vertraut machen.

In diesem Abschnitt gehen wir auf folgende Punkte näher ein:

- Einführung

- Vorgehensweise

## Einführung

Jeder der drei Typen von berechneten Feldern verfügt über unterschiedliche Eigenschaften, die sich auf seine Verwendung auswirken.

In der nachstehenden Tabelle finden Sie wichtigsten Unterschiede zwischen den drei Typen:

**Tabelle 13.1. Typen von berechneten Feldern**

Typ des Felds	Eigenschaften der Felder dieses Typs			Die Berechnung des Felds erfolgt über den	Merkmale der von der Berechnungsformel verwendeten Sprache	
	Anzeigbar	Sortierbar	In Filtern verwendbar		Vorteile	Nachteile
AQL	Ja	Ja	Ja	Datenbank-Server	Leistungsfähig, Bearbeitungsassistent	Vereinfachte Sprache (z. B.: keine einfachen Tests vom Typ IF, THEN, ELSE). Die Felder dieses Typs können nicht in den Standardwerten verwendet werden.

Typ des Felds	Eigenschaften der Felder dieses Typs			Die Berechnung des Felds erfolgt über den	Merkmale der von der Berechnungsformel verwendeten Sprache	
	Ja	Ja	Nein		Benutzerfreundlich	Nicht sehr leistungsfähig (einfache Verkettung von Strings und Werten von Feldern und Verknüpfungen).
Berechnete Strings	Ja	Ja	Nein	Client	Benutzerfreundlich	Nicht sehr leistungsfähig (einfache Verkettung von Strings und Werten von Feldern und Verknüpfungen).
BASIC	Ja	Nein	Nein	Client	Vielseitig Flexibel	Die Felder dieses Typs sind nur anzeigbar.

 **Anmerkung:**

Aus dieser Tabelle ist zu ersehen, dass die berechneten Felder vom Typ "AQL" vielseitigere Verwendungsmöglichkeiten haben, als die beiden anderen Feldtypen.

Jede dieser drei Eigenschaften kann mit einer AQL-Abfrage abgestimmt werden:

Eigenschaft	Entsprechende AQL-Parameter
Anzeigbar	SELECT Klausel
Sortierbar	SELECT  ORDER BY  GROUP BY Klauseln

Eigenschaft	Entsprechende AQL-Parameter
In Filtern verwendbar	SELECT
	ORDER BY
	GROUP BY
	WHERE
	HAVING Klauseln

Weitere Informationen zu den AQL-Abfragen finden Sie im Handbuch "Referenzhandbuch: Verwaltung und erweiterte Funktionen von AssetCenter" Kapitel "Schreiben von AQL-Abfragen".

## Berechnungen auf dem Server bzw. Client

Im Fall eines Felds vom Typ "AQL" werden die Berechnungen vom Datenbankserver durchgeführt, der die Ergebnisse an den Client zurückgibt. Die Berechnungen wirken sich dementsprechend nicht auf die Verarbeitungsgeschwindigkeit des Clients aus, und die Bandbreite ist weniger belastet. Die AQL-Abfragen, die an die Datenbank-Engine weitergegeben werden, sind jedoch komplexer.

## Vorgehensweise

In diesem Kapitel wird die Vorgehensweise zur Erstellung eines berechneten Felds im Detail beschrieben.

## Analysieren Sie Ihre Anforderungen

Die Identifizierung des von Ihnen benötigten Feldtyps erfolgt mit Hilfe von zwei Konzepten:

- Das Konzept der "Auflage" spiegelt die Eigenschaften des Feldtyps wieder: anzeigbar, sortierbar, in Filtern oder in Standardwerten verwendbar.
- Das Konzept des "Aufwands" stellt das Verhältnis der Komplexität der Formel und ihren Verwendungsmöglichkeiten dar. Sie können die berechneten Felder anhand dieses "Aufwands" klassifizieren:
  - Berechneter String
  - AQL
  - BASIC

### Anmerkung:

Wir empfehlen Ihnen, sooft wie möglich den Typ des berechneten Felds mit dem geringsten "Aufwand" zu verwenden.

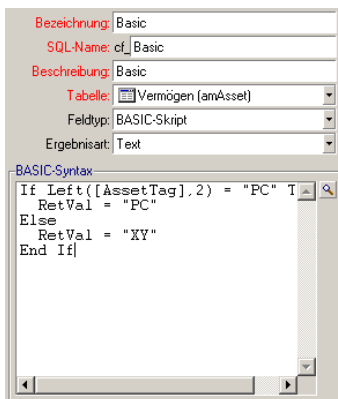
Ein Beispiel:

- Wenn das Feld ausschließlich Informationszwecken dienen soll, ist ein berechnetes Feld vom Typ BASIC völlig ausreichend.
- Wenn Sie die Datensätze einer Tabelle gegebenenfalls unter Berücksichtigung des Feldwerts sortieren möchten, müssen Sie sich für ein berechnetes Feld vom Typ "AQL" oder "Berechneter String" entscheiden.
- Wenn Sie die Datensätze einer Tabelle gegebenenfalls unter Berücksichtigung des Feldwerts filtern möchten, muss das berechnete Feld dem Typ "AQL" entsprechen.

Sobald Sie Ihre Anforderungen analysiert haben, können Sie zur eigentlichen Erstellung des berechneten Felds übergehen.

## Öffnen Sie den Bildschirm zur Erstellung eines berechneten Felds

Wählen Sie das Menü **Verwaltung/ Berechnete Felder**. AssetCenter zeigt den Bildschirm zur Erstellung der berechneten Felder an.



Bezeichnung: Basic

SQL-Name: cf\_Basic

Beschreibung: Basic

Tabelle:  Vermögen (amAsset)

Feldtyp: BASIC-Skript

Ergebnisart: Text

BASIC-Syntax:

```
If Left([AssetTag].2) = "PC" Then
  RetVal = "PC"
Else
  RetVal = "XY"
End If
```

## Identifizieren Sie das berechnete Feld

Geben Sie zunächst die erforderlichen Daten in der oberen Hälfte des Bildschirms ein, um das berechnete Feld eindeutig zu identifizieren:

- Das Feld **Bez.** (SQL-Name: TextLabel) enthält die Bezeichnung des berechneten Felds. Diese Bezeichnung wird in den Listen für die Spaltenüberschriften verwendet.
- Das Feld **SQL-Name** (SQL-Name: SQLName) enthält den SQL-Namen des berechneten Felds. Diesem Namen ist das Präfix "cf\_" vorangestellt ist und er wird beispielsweise in den BASIC-Skripten, den Abfragen und den Filtern als Referenz für das Feld verwendet.

 **Anmerkung:**

Sie dürfen den SQL-Namen eines berechneten Felds nach seiner Erstellung unter keinen Umständen ändern. Die Verweise auf das Feld, die den alten SQL-Namen verwenden, verlieren sonst ihre Gültigkeit.

- Das Feld **Beschreibung** (SQL-Name: Description) enthält eine Kurzbeschreibung des Felds. Diese Beschreibung kommt in den Listen zur Anwendung, in denen diese Felder vorgestellt werden. Das ist beispielsweise der Fall bei Filtern bzw. in den Bildschirmen zur Konfiguration einer Liste.


## Definieren Sie den Kontext der Verwendung eines Feld

Über die Felder **Tabelle** (SQL-Name: TableName) und **Feldtyp** (SQL-Name: seType) definieren Sie den Kontext, in dem das berechnete Feld verwendet werden soll:

- Über das Feld **Tabelle** ordnen Sie das berechnete Felds einer bestimmten Tabelle zu. Das Feld ist dann nur für die jeweilige Tabelle verfügbar.
- Über das Feld **Feldtyp** legen Sie den Typ des berechneten Felds fest. Je nach dem von Ihnen definierten Typ weist das Feld unterschiedliche Eigenschaften auf (anzeigbar, sortierbar, in Filtern verwendbar).
- Über das Feld **Ergebnisart** (SQL-Name: seDataType) geben Sie den Typ des Ergebnisses des berechneten Felds an. Dieser Typ kommt bei der Formatierung und der Anzeige zur Anwendung. Wenn es sich bei dem Ergebnis eines berechneten Fels um ein Datum handelt, wird es wie alle anderen Felder vom Typ "Datum" in der Datenbank angezeigt.

## Geben Sie die Formel zur Berechnung des Felds ein

Jetzt brauchen Sie nur noch die Berechnungsformel für das Feld zu bearbeiten. Sie können diese Formel entweder direkt im Textfeld in der unteren Hälfte des Bildschirms eingeben, wobei die Bezeichnung dieses Felds je nach Funktion

des Typs des berechneten Feld wechselt, oder Sie können auf die Eingabehilfe zurückgreifen, die für jeden Typ eines berechneten Felds zur Verfügung stehen. Klicken Sie hierzu auf die Schaltfläche  oder drücken Sie auf die Taste "F4".

 **Anmerkung:**

Je nach Typ eines berechneten Felds wird eine andere Sprache verwendet.

Weitere Informationen zu den Sprachen, die zum Schreiben der Berechnungsformeln verwendet werden können, finden Sie in den nachstehenden Handbüchern:


- "Referenzhandbuch: Verwaltung und erweiterte Funktionen von AssetCenter", Kapitel "Arbeiten mit Skripten" für die Sprache BASIC. Die verwendete Funktion lautet **RetVal()**.
- "Referenzhandbuch: Verwaltung und erweiterte Funktionen von AssetCenter" Kapitel "Schreiben von AQL-Abfragen" für die Sprache AQL.
- "Referenzhandbuch: Beschaffungsverwaltung", Kapitel "Struktur der AssetCenter-Datenbank", Abschnitt "Beschreibung der Tabellen", Absatz "Beschreibungsfolgen der AssetCenter-Tabellen" für die berechneten Strings.

## Bestimmen Sie die Standardberechtigungen für das berechnete Feld

Wählen Sie das Menü **Verwaltung/ Berechtigungsverwaltung/ Benutzungsberechtigungen**. AssetCenter zeigt den Bildschirm zur Erstellung der Benutzungsberechtigungen an.

 **Anmerkung:**

Berechnete Felder sind schreibgeschützt.

- 1 Geben Sie im Feld Beschreibung (SQL-Name: Description) eine kurze **Beschreibung** der Standardberechtigung und im Feld **Kommentar** (SQL-Name: Comment) eventuell einen Kommentar ein.
- 2 Öffnen Sie die Baumstruktur der Tabelle, der das berechnete Feld zugeordnet ist. Der mit dem Symbol  (Berechnete Felder) gekennzeichnete Zweig enthält eine komplette Liste der berechneten Felder der betreffenden Tabelle.
- 3 Wählen Sie das Feld, für das Sie eine Berechtigung erstellen möchten. Über das Kontrollkästchen **Schreibgeschützt** im Rahmen Felder, **Verknüpfungen und Merkmale** können Sie die Berechtigungen für die Lesezugriffe definieren. Wenn dieses Kontrollkästchen markiert ist, kann das berechnete Feld nur von den Benutzern mit der entsprechenden Berechtigung



eingesehen werden. Ist die Markierung des Kontrollkästchens jedoch aufgehoben, sind alle Benutzer zum Lesezugriff auf das Feld berechtigt.

## Arbeiten mit berechneten Feldern

Der Typ eines berechneten Felds bestimmt seinen Verwendungszweck. Achten Sie darauf, dass der vorgesehene Verwendungszweck auch tatsächlich mit dem Typ des Felds kompatibel ist. In den Listen, in denen die Felder enthalten sind, (Bearbeitung eines Filters, Konfiguration einer Liste, ...), werden Sie von AssetCenter unterstützt, indem die Software nur die verwendbaren Felder anzeigt.

### Berechnete Felder in der Konfiguration einer Liste verwenden

Mit Hilfe des Kontextmenüs und dem Befehl **Liste konfigurieren** können Sie den Wert eines berechneten Felds für alle Datensätze einer Tabelle anzeigen.

Öffnen Sie in der Liste der Felder und Verknüpfungen einer Tabelle den Zweig **Berechnete Felder**, um auf die verfügbaren berechneten Felder zugreifen zu können. AssetCenter zeigt die berechneten Felder in dem Format an, das über das Menü **Bearbeiten/ Optionen**, Kategorie **Anzeigen** definiert wurde.

Anschließend können Sie die berechneten Felder wie jedes andere Feld zu der Liste hinzufügen.

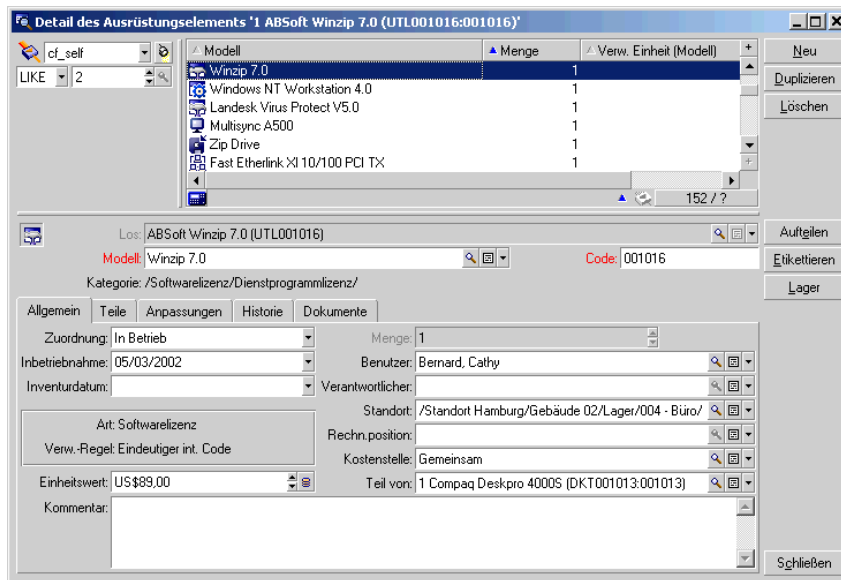
### Datensätze einer Tabelle filtern

AssetCenter kann die Datensätze einer Tabelle unter Berücksichtigung des Werts eines berechneten Felds vom Typ "AQL" filtern. Wählen Sie dazu im Kontextmenü die Option **Feldfilter** und durchsuchen Sie die Baumstruktur bis zur Verzweigung **Berechnete Felder**. AssetCenter zeigt nur die Felder vom Typ "AQL" an.

### Referenz für ein berechnetes Feld erstellen

Der SQL-Name, dem die Zeichen "cf\_" vorangestellt sind, wird zur Erstellung der Referenz eines berechneten Felds verwendet. Die nachstehende Abbildung

veranschaulicht die Verwendung des SQL-Namens eines berechneten Felds in einem Filter:



Berechnete Felder können von verschiedenen AssetCenter-Modulen bzw. -Funktionalitäten verwendet werden:

- AssetCenter Web
- AssetCenter API
- Berichte
- Formulare

Weitere Informationen zu den Modulen bzw. Funktionalitäten finden Sie in den entsprechenden Handbüchern.

# 14 | Assistenten

## KAPITEL

Mit AssetCenter können Sie eigene Assistenten erstellen bzw. vorhandene Assistenten zur Anpassung an Ihre Anforderungen konfigurieren. Ein Assistent wird im Feld **Assistentenskript** (SQL-Name: WizardScript) der Registerkarte **Assistent** im Detailfenster einer Aktion vom Typ **Assistent** gespeichert. Zur Erstellung eines Assistenten geben Sie den entsprechenden Code in diesem Feld ein oder verwenden den Grafikeditor. Voraussetzung dafür ist die Kenntnis der Struktur eines Assistenten und der Skriptsprache, die zur Beschreibung dieser Struktur verwendet wird.

## Schreibweisenkonventionen

Die Struktur des Assistenten wird mit den folgenden Notationen beschrieben:

**Tabelle 14.1. Verwendete Konventionen**

[ ]	Die eckigen Klammern werden verwendet, um den Wert eines Felds der Datenbank (bei kontextbezogenen Assistenten) oder der Spezialfelder "CurrentSelection" und "CurrentTable" als Referenz auszuweisen. Darüber hinaus dienen sie als optionale Parameter.
< >	Der Wert für eine in Klartext beschriebene Eigenschaft wird in diese Klammern gesetzt. Geben Sie diese Klammern nicht erneut ein, sondern ersetzen Sie den darin enthaltenen Text mit den erforderlichen Informationen.
	Der vertikale Strich trennt die möglichen Werte einer Eigenschaft. Zusätzlich wird dieser Strich zum Trennen von Überschriften und Werten in einer mehrspaltigen Liste verwendet.
{ }	Die geschweiften Klammern umfassen die Definition eines Knotens oder eines Skriptblocks mit mehreren Zeilen für eine Eigenschaft. Sie dienen ebenfalls dazu, den Wert der Eigenschaft eines Assistenten als Referenz auszuweisen.
'	In den Beispielen zum Basic-Code bezeichnet das Apostroph eine Zeile mit einem Kommentar, die von AssetCenter nicht interpretiert wird.
; oder //	In den Assistenten kennzeichnen das Semikolon oder zwei Schrägstriche eine Kommentarzeile, die von AssetCenter nicht interpretiert wird.

## Definitionen zu Assistenten

Auf den nächsten Seiten finden Sie die Definition der Begriffe, die zur Beschreibung der Struktur eines Assistenten verwendet werden.

### Twip

Ein "Twip" ist die Maßeinheit, die von den Assistenten standardmäßig für Schriftgrad und -abstand verwendet werden. Diese Maßeinheit gilt unabhängig von der Auflösung des Bildschirms. Nachfolgend finden Sie einige Entsprechungen mit herkömmlichen Maßeinheiten:

- 1440 "Twips" entsprechen einem Zoll.
- 567 "Twips" entsprechen einem Zentimeter.
- Bei einer Auflösung von 96 dpi (Windows-Standard) entsprechen 15 Twips einem Pixel.

## Steuerelement

Ein Steuerelement bezeichnet ein grafisches Element, mit dem Daten bearbeitet werden können. Dabei kann es sich u. a. um ein Kontrollkästchen, ein Feld zur Eingabe von Text, eine Schaltfläche oder eine Aufzählung handeln.

## Knoten

Ein Knoten entspricht einer hierarchischen Ebene in der Baumstruktur des Assistenten. Unter einem Nebenknoten des Knotens "N" ist ein Knoten zu verstehen, der sich auf einer dem Knoten "N" hierarchisch untergeordneten Ebene befindet, und dabei gleichzeitig mit dem Knoten "N" verbunden ist.

---

 Anmerkung:

Für die Knotennamen dürfen nur alphanumerische Zeichen verwendet werden. Die Länge eines Knotennamens ist auf 22 Zeichen beschränkt.

---

## Objekt

Ein Objekt ist ein allgemeiner Name, der eines der nachstehenden Elemente bezeichnet:

- Ein Assistent in seiner Gesamtheit
- Eine Seite eines Assistenten
- Ein Steuerelement (Kontrollkästchen, Textfeld, Schaltfläche, Feld, Aufzählung usw.) einer Seite
- Eine Variable
- Sonstige

## Übergeordnete und untergeordnete Objekte

Wenn eine Objekt "A" ein Objekt "B" enthält, dann gilt folgendes:

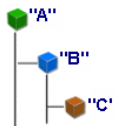
- Das Objekt "A" ist dem Objekt "B" "übergeordnet".
- Das Objekt "B" ist dem Objekt "A" "untergeordnet".

 **WARNUNG:**

Bei diesem Konzept handelt es sich um eine Zusammensetzung und nicht um eine Vererbung.

## Vollständiger Name eines Objekts

Der vollständige Name eines Objekts besteht aus dem Namen sämtlicher übergeordneter Objekte und dem Name des betroffenen Objekts. Die einzelnen Objektnamen sind durch einen Punkt (".") voneinander getrennt. Nehmen wir beispielsweise die nachstehende Struktur:



Der vollständige Name des Objekts "C" lautet wie folgt:

```
<Name des Objekts "A">.<Name des Objekts "B">.<Name des Objekts "C">
```

## Variable

Eine Variable ist ein mit einem Namen versehener Speicherplatz mit Daten, die im Verlauf der Ausführung des Assistenten geändert werden können. Jede Variable verfügt über einen Namen, mit dem sie innerhalb des Assistenten eindeutig gekennzeichnet ist.

Die AssetCenter-Assistenten verwenden zwei Arten von Variablen:

- Die proprietären Variablen, auch "Assistentenvariablen" genannt, sind in einem Knoten vom Typ "LONG" oder "STRING" definiert. Über den Knotentyp wird der Typ der Variablen festgelegt. Bei einer Variablen, die in einem Knoten vom Typ "LONG" definiert ist, handelt es sich um eine ganze lange Zahl, während eine Variable, die in einem Knoten vom Typ "STRING" definiert ist, einer Zeichenfolge entspricht. Diese Variablen werden als globale Variablen bezeichnet. Das bedeutet, dass sie über ihren vollständigen Namen in jedem beliebigen Knoten des Assistenten als

Referenz ausgewiesen werden können. Diese Variablen werden gegebenenfalls von AssetCenter automatisch neu berechnet.

- Die BASIC-Variablen werden in den BASIC-Skripten innerhalb des Assistenten verwendet. Es handelt sich dabei standardmäßig um lokale Variablen, die jedoch mit Hilfe der Eigenschaften "COMMON" und "GLOBAL" in globale Eigenschaften umgewandelt werden können. Diese Variablen werden von AssetCenter nicht automatisch neu berechnet.

## Übergang

Mit einem Übergang wird der Wechsel von einer Seite zur nächsten innerhalb des Assistenten bezeichnet. Dabei lassen sich für eine einzelne Seite des Assistenten durchaus mehrere Übergänge definieren. Für jeden Übergang gelten bestimmte, vom Benutzer definierte Bedingungen, deren Erfüllung zum Auslösen des Übergangs führt.

- Wenn der Benutzer auf die Schaltfläche **Weiter** des Assistenten klickt, wird der erste gültige Übergang ausgeführt. Ein Übergang ist gültig, wenn alle Bedingungen erfüllt sind. Wenn kein gültiger Übergang vorliegt, ist die Schaltfläche **Weiter** deaktiviert.
- Wenn der Assistent obligatorische (MANDATORY) Eigenschaften enthält und diese nicht ausgefüllt sind, ist eine Verwendung der Schaltfläche **Weiter** unmöglich.
- Wenn der Benutzer vor dem Abschließen der einzelnen Schritte auf **Beenden** klickt, werden Standardwerte für die nicht abgeschlossenen Schritte verwendet.

## Strukturmodelle

Ein AssetCenter-Assistent ist das Ergebnis der Ausführung eines Skripts. Definitionsgemäß entspricht ein Assistent einer bestimmten Struktur:

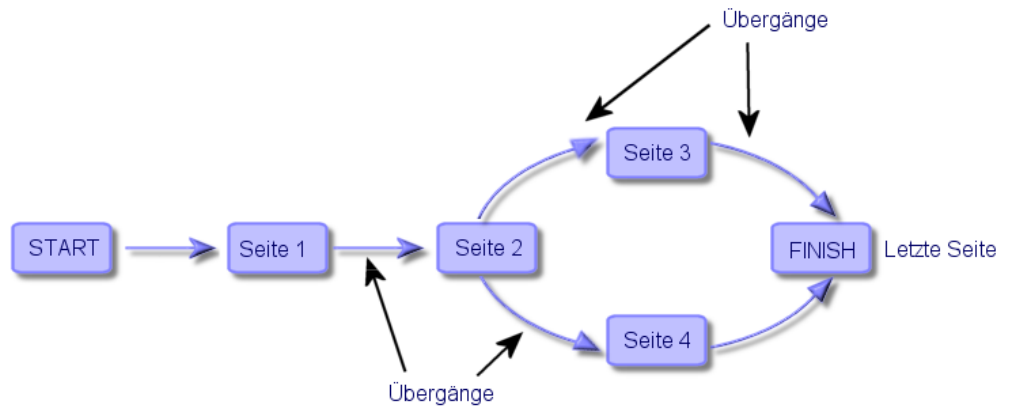
- Ein Assistentenskript, also ein Assistent, besteht aus einzelnen Knoten:
- Jeder Knoten des Assistenten verfügt über einen Namen, einen oder mehrere Nebenknoten und eine Reihe von Eigenschaften. Bei den Knoten sind nachstehende Typen zu unterscheiden:
  - "ROOT" (Knoten Root oder Wurzel). Dieser Knoten ist nur einmal vorhanden und enthält alle anderen Knoten.

- "START". Dieser Knoten ist nur einmal vorhanden und enthält ein Skript, das beim Starten des Assistenten ausgeführt wird.
- "PAGE". Diese Art von Knoten beschreibt eine Seite innerhalb des Assistenten.
- "TRANSITION". Diese Art von Knoten beschreibt den Übergang zwischen zwei Knoten vom Typ "PAGE".
- "FINISH". Dieser Knoten ist nur einmal vorhanden und wird zum Beenden des Assistenten ausgeführt.
- "PARAMS". Dieser Knoten ist nur einmal vorhanden und enthält die Parameter, die an einen anderen Assistenten weitergeleitet werden sollen. Es können durchaus mehrere Assistenten hintereinander ausgeführt werden, wobei gegebenenfalls der Austausch von Parametern möglich ist. Man spricht in diesem Fall von verketteten Assistenten.
- "LONG" oder "STRING". Diese Art von Knoten definiert eine Variable des entsprechenden Typs.
- Der Wert einer Eigenschaft wird entweder mithilfe einer Konstanten oder über ein Basic-Skript festgelegt. Im letzteren Fall entspricht der Wert dem Ergebnisse der Skriptbeurteilung.

Assistenten setzen sich aus mehreren Seiten zusammen, die durch Übergänge miteinander verbunden sind. Die Verzweigung von einer Seite zur nächsten wird durch die eingegebenen Informationen oder die vom Benutzer getroffenen Auswahlen bestimmt. Die nachstehende Abbildung zeigt das Strukturmodell eines Assistenten:



Abbildung 14.1. Assistenten - Strukturmodell



## Modell der Seite eines Assistenten

Die Seite eines Assistenten besteht aus folgenden Elementen:



## Allgemeine Informationen zu Assistenten

Der Code eines Assistenten wird im Feld **Assistentenskript** (SQL-Name: WizardScript) der Tabelle **amAction** in der Registerkarte **Assistent** im Detailfenster einer Aktion eingegeben, die auf einen Assistenten zurückgreift. Dieser Code besteht aus einem strukturierten Text, in dem Textblöcke durch geschweifte Klammern ( { } ) abgegrenzt sind, und definiert die Struktur eines Assistenten. Der Aufbau erfolgt in Form einer Baumstruktur.

Jeder dieser Knoten ("Wurzel", "Seite" usw.) der Baumstruktur des Assistenten enthält eine unbegrenzte Anzahl von Nebenknoten und eine Reihe von Eigenschaften.

## Allgemeine Struktur und Syntax eines Assistentenknotens

Ein Knoten weist die nachstehende Struktur und Syntax auf:

```

; Dies ist ein Kommentar, der sich außerhalb eines Skripts befindet
{
  <Typ des Knotens> <Name des Knotens>
  <Name der Eigenschaft> = <Wert der Eigenschaft>
  ' Dies ist ein Kommentar, der sich innerhalb eines Skripts befindet
  ...
  { <Name der Eigenschaft> =
    ...
  }
  { <Typ des Knotens> <Name des Knotens>
    <Name der Eigenschaft> = <Wert der Eigenschaft>
    ...
    { <Name der Eigenschaft> =
      ...
    }
  }
}

```

Für die Knoten gelten folgende Regeln:

- Die Namen der Knoten sind optional. Wenn Sie keinen Name für den Knoten angegeben haben, wird ihm von AssetCenter automatisch ein Name und eine Nummer zugeordnet.
- Die Namen der Knoten dürfen keine Leerstellen aufweisen.
- Wenn der Name eines Knotens dem Zeichen = entspricht, handelt es sich nicht mehr um einen Knoten, sondern um eine mehrzeilige Eigenschaft. Weitere Informationen zu mehrzeiligen Eigenschaften finden Sie im Abschnitt "Syntax eines mehrzeiligen Skripts" in diesem Kapitel.

- Die Zeilen, die außerhalb eines Skripts mit einem Komma (";") und innerhalb eines Skripts mit einem Apostroph (') beginnen, werden als Kommentarzeilen interpretiert und ignoriert.

 Anmerkung:

Zwischen der geschweiften Klammer ("{") und dem Typ des Knotens muss ein Leerzeichen vorhanden sein. Ist das nicht der Fall, wird die Ausführung des Assistenten von AssetCenter verweigert.

## Eigenschaften eines Knotens

Die Werte der Eigenschaften können mit Hilfe von Konstanten oder Skripten definiert werden. Dabei kann es sich um numerische oder boolesche Konstanten bzw. Konstanten vom Typ Text handeln.

 Anmerkung:

Objekten zugeordnete Eigenschaften können sowohl optional als auch obligatorisch sein. Zusätzlich ist bei den Eigenschaften der logische Typ vom physischen Typ zu unterscheiden. "Logische" Eigenschaften vervollständigen die Definition des Objekts, und "physische" Eigenschaften wirken sich auf den visuellen Aspekt des Objekts aus.

## Beschreibendes Modell

Die Definition einer Eigenschaft erfolgt mittels einer Beschreibung, in der Kreisreferenzen zwischen Variablen erkennbar sind ( $A=\{B\}$ ,  $B=\{A\}$ ):

```
<Name der Eigenschaft>=<Skript>
```

Dieser Definition ist eine Liste mit Abhängigkeiten zugeordnet. Dabei gilt:

```
A={B}+{C}
```

Die Eigenschaft "A" ist abhängig von den Eigenschaften "B" und "C". Die Liste der Abhängigkeiten von "A" setzt sich zusammen wie folgt: "B", "C".

Daraus ergibt sich eine Änderung der Eigenschaft in folgenden Situationen:

- Änderung der Eigenschaften in der Liste der Abhängigkeiten

- Änderung infolge der Aktion eines Benutzers, die zur Änderung der Eigenschaft bzw. einer der abhängigen Eigenschaften führt

## Eine Konstante als Wert für eine Eigenschaft definieren

Mit den nachstehenden Syntaxstrukturen definieren Sie einen konstanten Wert für eine Eigenschaft:

- Eigenschaft vom Typ Text:
  - `<Name der Eigenschaft> = "<Text>"`
- Eigenschaft vom Typ Boolesch:
  - `<Name der Eigenschaft> = TRUE`
  - `<Name der Eigenschaft> = FALSE`
  - `<Name der Eigenschaft> = entspricht <Name der Eigenschaft> = TRUE`
- Eigenschaft vom Typ Numerisch:
  - `<Name der Eigenschaft> = 42`
- `<Name der Eigenschaft> = {<Vollständiger Name einer BASIC-Variablen oder einer Eigenschaft>}`



### Anmerkung:

Der boolesche Wert "TRUE" entspricht einem numerischen Wert ungleich Null. Der boolesche Wert "FALSE" entspricht dem numerischen Wert 0.

## Eigenschaft als Referenz darstellen

Verwenden Sie die folgende Syntax, um eine Eigenschaft oder ein Objekt als Referenz darzustellen, sich also auf den Inhalt dieser Eigenschaft oder dieses Objekts, und insbesondere auf seinen Wert zu beziehen:

```
{<Vollständiger Name der Eigenschaft>}
```

Wenn Sie die Eigenschaft "Prop" einer Seite "Page1" als Referenz darstellen möchten, schreiben Sie:

```
{Page1.Prop}
```

Bei dieser Syntax spielt die Groß- und Kleinschreibung im vollständigen Namen keine Rolle.

## Ein Skript als Wert für eine Eigenschaft definieren

### Konzept eines Skripts

Ein Skript ist ein einzeliliges oder mehrzeiliges BASIC-Programm, das einen Wert in der globalen Variablen "RetVal" zurückgibt. Im Fall eines einzeliligen Skripts ist diese Variable implizit enthalten. In einem mehrzeiligen Skript muss Sie jedoch explizit vorhanden sein.

Wie allgemein in einem BASIC-Skript üblich, müssen Sie auch hier auf den Typ des zurückgegebenen Werts achten. Dabei ist der Wert abhängig vom Typ der mit Hilfe eines Skripts berechneten Eigenschaft.

### Syntax eines einzeliligen Skripts

```
<Name der Eigenschaft>=<Skript>
```

Ein Beispiel:

```
Variable="Der Name lautet: " & {Name}
```

Das zuvor gezeigte einzelilige Skript ist identisch mit dem nun folgenden mehrzeiligen Skript:

```
{ Variable =  
RetVal="Der Name lautet: " & {Name}  
}
```

### Syntax eines mehrzeiligen Skripts

```
{ <Name der Eigenschaft >=  
<Skript>  
}
```

Ein Beispiel:

```
{ LABEL =  
IF {Page1.Title}="Wählen Sie eine Person" THEN  
RetVal="Person"  
ELSE  
RetVal="Abteilung"  
END IF  
}
```

## Auf Eigenschaften anwendbare Methoden

Mit einer Methode rufen sie den Wert ab, der mit einer Eigenschaft oder einem Knoten verknüpft ist, und Sie führen eine Funktion mit dieser Eigenschaft aus. In diesem Sinne kann die Methode als erweiterte Funktion betrachtet werden.

Die Syntax der Methode lautet wie folgt:

```
{Knoten.Knoten.Knoten[.Eigenschaft][.Methode([arg1[, arg2[ ]]])]}
```

mit:

- Knoten: Name des Knoten
- Eigenschaft: Name der Eigenschaft
- Methode: Name der Methode
- arg1, arg2, ...: Konstante bzw. BASIC-Ausdruck (dieser Ausdruck darf keine geschweiften Klammern enthalten)

 Anmerkung:

In unserem Beispiel grenzen die Zeichen "[" und "]" die fakultativen Elemente ab.

Wenn Sie beispielsweise die Anzahl der Zeilen des Steuerelements "LISTBOX" aus der Seite "PAGE1" abrufen möchten, verwenden Sie die diesem Steuertyp zugeordnete Methode "COUNT". Der Befehl lautet wie folgt:

```
{PAGE1.LISTBOX.VALUES.COUNT() }
```

## Eigenschaft vom Typ Tabelle

Die Eigenschaften vom Typ Tabelle sind die Eigenschaften, deren Wert im nachstehenden Format definiert ist:

```
<Spalte|Spalte|Spalte|...>=<Wert der Spalte>, <Spalte|Spalte|Spalte|...>=<Wert der Spalte>, ...
```

Der Wert dieser Eigenschaften kann in Form einer Tabelle angezeigt werden:

		Spalte 1	Spalte 2	Spalte 3
Nummer der Zeile: 1	Wert der Zeile (z. B.: 18)	Zelle (1,1)	Zelle (2,1)	Zelle (3,1)
Nummer der Zeile: 2	Wert der Zeile (z. B.: 29)	Zelle (1,2)	Zelle (2,2)	Zelle (3,2)

		<b>Spalte 1</b>	<b>Spalte 2</b>	<b>Spalte 3</b>
<b>Nummer der Zeile: 3</b>	<b>Wert der Zeile (z. B.: 78)</b>	Zelle (1,3)	Zelle (2,3)	Zelle (3,3)
...	...	...	...	...



Anmerkung:

Die Kennung ist vom Typ 'Text'.

## Beispiel

Wir nehmen die Eigenschaft "VALUES" des Knotens "LISTBOX", deren Wert dem Ergebnis einer Abfrage in der Tabelle der Abteilungen und Personen entspricht. Die betreffende Abfrage gibt für jeden Datensatz der Tabelle die Werte der Felder **Name** (SQL-Name: Name) und **Vorname** (SQL-Name: FirstName) zurück. Wir gehen davon aus, dass die Eigenschaft den folgenden Wert aufweist:

```
VALUES="Colombo|Gerald=32,Lübeck|Alexandre=64,Daquin|William=24"
```

Dieser Wert kann in Form einer Tabelle angezeigt werden:

		<b>Name</b>	<b>Vorname</b>
<b>1</b>	<b>32</b>	Colombo	Gérard
<b>2</b>	<b>64</b>	Lübeck	Alexandre
<b>3</b>	<b>24</b>	Daquin	William

## Die globalen Variablen CurrentTable und CurrentSelection verwenden

Der Inhalt dieser Variablen kann unter Verwendung der nachstehenden Syntax abgerufen werden:

```
[CurrentTable]
[CurrentSelection]
```

Die folgende Tabelle zeigt die Merkmale dieser beiden Variablen:

Name der Variablen	Beschreibung der Variablen	Kommentar
CurrentTable	Enthält den SQL-Namen der Tabelle, die beim Starten des Assistenten aktiv ist. Wenn keine aktive Tabelle vorliegt, erscheint ein leerer String. Variable vom Typ "String".	Diese Variable wird von AssetCenter automatisch abgerufen. Der Benutzer kann den Wert nicht erzwingen.
CurrentSelection	Enthält die Liste der durch ein Komma getrennten internen Datensatzkennungen, die zum Zeitpunkt des Starts des Assistenten ausgewählt waren. Variable vom Typ "String".	Diese Variable wird von AssetCenter automatisch abgerufen. Wenn keine Auswahl stattgefunden hat bzw. keine Tabelle definiert wurde, enthält sie einen leeren string. Der Benutzer kann den Wert nicht erzwingen.

## Verkettung von Assistenten

Nach seiner Ausführung kann ein Assistent die Ausführung eines anderen Assistenten auslösen und die entsprechenden Parameter (Variablen) weitergeben. Dieser Vorgang wird mit dem Begriff der Verkettung von Assistenten umschrieben.

### Ausführung

Wenn ein Assistent A einen Assistenten B auslösen soll, muß der Knoten des Assistenten A über die Eigenschaft CHAIN verfügen. Der Wert dieser Eigenschaft muss dem SQL-Namen der Aktion vom Typ Auszuführender **Assistent** entsprechen, in diesem Fall also "B".

### Parameter

Die Parameter werden über den Knoten PARAMS vom Assistenten A zum Assistenten B weitergeleitet und zu den Parametern des Knotens PARAMS im Assistenten B hinzugefügt. Wenn ein Parameter sowohl im Knoten PARAMS des Assistenten A als auch im Knoten PARAMS des Assistenten B definiert ist, wird nur die im Assistenten A aufgeführte Definition berücksichtigt.



## BASIC-Funktionen

Neben den allgemeinen AssetCenter-Funktionen, mit Ausnahme der Funktion "AmCounter", akzeptieren die Assistenten die folgenden Funktionen:

- AmComputeString()
- AmDecrementLogLevel()
- AmExecTransition()
- AmLog()
- AmMsgBox()
- AmPagePath()
- AmProgress()
- AmRefreshProperties
- AmSetProperty
- AmUpdateDetail
- AmValueOf
- AmWizChain

### WARNUNG:

Wenn Sie im Rahmen des Assistentenskripts auf eine BASIC-Funktion zurückgreifen, müssen Sie der Variablen grundsätzlich den von der Funktion zurückgegebenen Wert zuordnen. Wenn Sie das nicht tun, gibt der BASIC-Compiler einen Fehler aus. Im folgenden Beispiel ist die Kompilierung nicht möglich:

```
AmGetFieldLongValue(hRecord, "lUserId", {lEmplDeptId})
```

Das folgende Skript weist keine Fehler auf:

```
Dim lValue as Long
lValue=AmGetFieldLongValue(hRecord, "lUserId", {lEmplDeptId})
```

## Definition eines Knoten vom Typ Root

Der Knoten vom Typ "Root" beschreibt den Assistenten in seiner Gesamtheit. Dieser Knoten setzt sich aus einem Block von allgemeinen Eigenschaften zusammen, die auf jeden Assistenten anwendbar sind, und aus einer Reihe von

untergeordneten Knoten, mit denen die im Assistenten enthaltenen Objekte dargestellt werden.

## Syntax eines Knotens vom Typ Root

Die Syntax eines Knotens vom Typ "Root" lautet wie folgt:

```
' Block der allgemeinen Eigenschaften des Knotens "Root"
NAME=...
IMAGE=...
...
' Definition der Knoten, die dem Knoten "Root" untergeordnet sind
{ FINISH
  ...
}
{ PAGE
  ...
}
{ TRANSITION
  ...
}
```

## Eigenschaften des Knotens Root

In der nachstehenden Tabelle sind alle logischen und physischen Eigenschaften aufgeführt, die in einem Knoten vom Typ "Root" definiert werden können:

**Tabelle 14.2. Logische Eigenschaften des Knotens "Root"**

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>	<b>Kommentar</b>
NAME="<Name des Assistenten>"	Definiert den Namen des Assistenten. Eigenschaft vom Typ "String".	NAME="Umzugsassistent"	Sie müssen den Wert für diese Eigenschaft definieren. Der Name des Assistenten ist auf 22 Zeichen begrenzt. Diese Eigenschaft kommt bei Operationen zur Serienwandlung des Assistenten zur Anwendung: Die Daten, die sich auf den Assistenten beziehen, werden unter diesem Namen gespeichert. Daher ist es vorzuziehen, dass zwei verschiedene Assistenten auch zwei verschiedene Namen tragen.
TITLE="<Fenstertitel>"	Definiert den Fenstertitel des Assistenten. Eigenschaft vom Typ "String".	TITLE="Umzugsassistent"	Wir empfehlen Ihnen dringend, einen Wert für diese Eigenschaft zu definieren.

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
GLOBAL=<Skript>	<p>Enthält ein BASIC-Skript (oder einen Code). Diese Eigenschaft ist beispielsweise beim Definieren globaler Variablen hilfreich, die auf sämtliche Assistenten angewendet werden können, und beim Definieren globaler Funktionen, die ebenfalls in allen Knoten des Assistenten verwendet werden können.</p> <p>Hierbei handelt es sich um ein deklaratives Skript, das nicht ausgeführt wird. Verwenden Sie den Knoten "START", um ein Skript beim Start auszuführen.</p> <p>Eigenschaft vom Typ "Script".</p>	{GLOBAL=Dim Filter As String}	
COMMON	<p>Enthält die BASIC-Funktionen, die automatisch in allen Assistenten enthalten sind.</p> <p>Diese Eigenschaft entspricht der geschützten Datei <b>gbbase.wiz</b>, die bei ihrer Erstellung in die Datenbank eingefügt wird.</p>		

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
SERIALIZE=<TRUE FALSE>	<p>Ermöglicht die Ausführung einer Serienwandlung des Assistenten (=TRUE) oder nicht (=FALSE). Wenn Sie die Serienwandlung für den Assistenten durchführen, speichert er die zuvor eingegebenen Werte bis zu seiner nächsten Ausführung in der Datei <b>.ini</b>.</p> <p>Über die Eigenschaft NAME wird festgelegt, in welchem die Werte in der Datei <b>.ini</b> gespeichert werden sollen.</p> <p>Eigenschaft vom Typ "Boolean".</p> <p>Ablageort der <b>.ini</b>-Dateien: ► Handbuch <b>AssetCenter - Installation und Aktualisierung</b>, Kapitel <b>.ini- und .cfg-Dateien</b>.</p>	SERIALIZE=TRUE	Standardmäßig gilt für diese Eigenschaft der Wert "FALSE".
MODAL=<TRUE FALSE>	Definiert die Modalität des Assistenten (=TRUE) oder nicht (=FALSE).		

**Tabelle 14.3. Physische Eigenschaften des Knotens "Root"**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
IMAGE="<Pfad der Bitmap-Datei> "IMAGE16="<Pfad der Bitmap-Datei>"	Definiert die grafische Datei vom Typ Bitmap (".bmp"), die vom Assistenten angezeigt werden soll.  Eigenschaft vom Typ "String".	IMAGE="Assistent.bmp"	Wenn für diese Eigenschaft kein Wert definiert wurde, erscheint kein Bild. Der Pfad der Grafikdatei wird relativ zum AssetCenter-Ordner config angegeben.  AssetCenter sucht das Bild zunächst in der Datenbank.  Wenn Sie für die Eigenschaft "IMAGE16" einen Wert eingeben, kommt diese Eigenschaft anstelle der Eigenschaft "IMAGE" bei der Einstellung des Bildschirms auf 16 Farben zur Anwendung.
WIDTH=<Breite>	Definiert die Breite ("<Width>"), die standardmäßig für das Assistentenfenster gelten soll. Die Angaben werden in Twips gemacht.  Eigenschaft des Typs "Long".	WIDTH=6000	

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
HEIGHT=<Höhe>	Definiert die Höhe (" <b>&lt;Height&gt;</b> "), die standardmäßig für das Assistentenfenster gelten soll. Die Angaben werden in Twips gemacht. Eigenschaft des Typs "Long".	HEIGHT=5000	
MINWIDTH=<MinWidth>	Definiert die minimale Breite des Assistentenfensters. In <b>twips</b> ausgedrückter Wert.		
MINHEIGHT=<MinHeight>	Definiert die maximale Breite des Assistentenfensters In <b>twips</b> ausgedrückter Wert.		
CTRLHEIGHT=<CtrlHeight>	Definiert die Höhe eines Steuerelements mit fester Vertikalität (z. B. das Steuerelement TEXT-BOX). In <b>twips</b> ausgedrückter Abstand.		

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
LABELSPACING=<labelSpacing>	Definiert den Abstand zwischen der Bezeichnung eines Steuerelements und dem Steuerelement selbst, wdn sich die Bezeichnung über dem Steuerelement befindet. In <b>twips</b> ausgedrückter Abstand.		
CTRLSPACING=<CtrlSpacing>	Definiert den vertikalen Abstand zwischen zwei Steuerelementen. In <b>twips</b> ausgedrückter Abstand.		
IMGBORDER=<Width>	Definiert den horizontalen Abstand zwischen dem Bild des Assistenten und seinem Steuerelementen. In <b>twips</b> ausgedrückter Wert.		
NAVIGATION=<TRUE FALSE>	Zeigt die Navigationsleiste mit den Schaltflächen <b>Weiter</b> , <b>Abbrechen</b> im Assistentenfenster an (=TRUE) oder nicht (=FALSE).		



Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
CONFIRMCANCEL=<TRUE FALSE>	Zeigt die Meldung zur Bestätigung des Abbruchs an (=TRUE) oder nicht (=FALSE).		
DEFAULTONNEXT=<TRUE FALSE>	Wählt standardmäßig (=TRUE) die Schaltfläche <b>Weiter</b> . Wenn DEFAULTONNEXT=FALSE, dann ist die standardmäßig ausgewählte Schaltfläche <b>Fertig stellen</b> .		

## Dem Knoten Root untergeordnete Knoten

In der nachstehenden Tabelle sind die verschiedenen Typen der untergeordneten Knoten aufgeführt, die Sie für einen Knoten vom Typ Root definieren können. Dabei stellt jeder Knotentyp ein "Objekt" dar.

**Tabelle 14.4. Dem Knoten "Root" untergeordnete Knoten**

Typ des Knotens	Beschreibung
PAGE	Beschreibt eine Seite des Assistenten.
FINISH	Beschreibt den letzten Übergang auf der letzten Seite des Assistenten, mit dem der Assistent normalerweise geschlossen wird. Für diese Knoten vom Typ "Transition" sind die Eigenschaften "FROM" und "TO" nicht vorhanden.

Typ des Knotens	Beschreibung
START	Enthält beispielsweise ein Skript, das beim Starten des Assistenten über die Eigenschaft "DO" ausgeführt werden soll, sowie den Name der Startseite des Assistenten (Eigenschaft "TO").
PARAMS	Ermöglicht das Übertragen der Parameter eines Assistenten auf einen andere, wenn die Eigenschaft CHAIN des Nebenknotens FINISH definiert wurde.
TIMER	Ermöglicht das Zuordnen einer Zeituhr zu einer Seite des Assistenten.

## Definition des Knotens Page

Mit einem Knoten vom Typ "Page" wird die Seite eines Assistenten beschrieben. Diese Knoten bestehen aus einem Block von Eigenschaften, die auf diesen Knoten sowie auf die entsprechenden untergeordneten Knoten anwendbar sind, sowie aus einer Reihe von Nebenknoten, mit denen die auf der Seite enthaltenen Objekte definiert werden.

## Syntax eines Knotens vom Typ Page

Die Syntax eines Knotens vom Typ "Page" lautet wie folgt:

```
' Beschreibung der Seite
{ Seite <Name der Seite>
' Block der Eigenschaften für den Knoten "Page"
IMAGE=...
TITLE=...
' Definition der Knoten, die dem Knoten "Page" untergeordnet sind
{ TRANSITION
...
}
{ <Typ des Steuerelements> <Name des Steuerelements>
...
}
...
}
```

## Eigenschaften eines Knotens vom Typ Page

In der nachstehenden Tabelle sind alle Eigenschaften aufgeführt, die in einem Knoten vom Typ "Page" definiert werden können:

**Tabelle 14.5. Logische Eigenschaften eines Knotens vom Typ "Page"**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
TITLE="<Titel der Seite>"	Definiert den Titel der Seite, der in Fettdruck am oberen Seitenrand erscheint.  Eigenschaft vom Typ "String".	TITLE="Umzug"	Wenn für diese Eigenschaft kein Wert definiert wurde, erbt sie den Wert der Eigenschaft "TITLE" des Knotens vom Typ "Root". Im Gegensatz zu den Bezeichnungen, wird HTML von diesem String nicht unterstützt.
ONENTER=<Skript>	Definiert ein Basic-Skript, das beim Zugriff auf die Seite mit einem Mausklick auf die Schaltfläche <b>Weiter</b> oder <b>Zurück</b> ausgeführt wird.  Eigenschaft vom Typ "Script".	{ONENTER = AmMsgBox ("Hello") }	

**Tabelle 14.6. Physische Eigenschaften eines Knotens vom Typ "Page"**

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>	<b>Kommentar</b>
IMAGE="<Pfad der Bitmap-Datei>"	Definiert die Bitmap-Datei (".bmp"), die auf der Seite des Assistenten angezeigt werden soll.	IMAGE ="C:\Images\Pa-ge1.bmp"	Wenn für diese Eigenschaft kein Wert definiert wurde, erbt sie den Wert der Eigenschaft "IMAGE" des Knotens vom Typ "Root".
IMAGE16="<Pfad der Bitmap-Datei>"	Eigenschaft vom Typ "Script".		Wenn für diese Eigenschaft kein Wert definiert ist, erscheint auch kein Bild. Wenn Sie für die Eigenschaft "IMAGE16" einen Wert eingeben, kommt diese Eigenschaft anstelle der Eigenschaft "IMAGE" bei der Einstellung des Bildschirms auf 16 Farben zur Anwendung.

## Dem Knoten Page untergeordnete Knoten

Für einen Knoten vom Typ "Page" können zwei verschiedenen Typen von untergeordneten Knoten definiert werden:

**Tabelle 14.7. Dem Knoten "Page" untergeordnete Knoten**

<b>Typ des Knotens / "Objekt"</b>	<b>Beschreibung</b>
<Typ des Steuerelements> <Name des Steuerelements>	Definiert ein Steuerelement, das auf der aktuellen Seite erscheint.
TRANSITION <Name des Übergangs>	Beschreibt den Übergang von der aktuellen Seite zur nächsten Seite des Assistenten.

Typ des Knotens / "Objekt"	Beschreibung
TIMER	Ermöglicht das Zuordnen einer Zeituhr zu einer Seite des Assistenten.

## Definition eines Knotens vom Typ Transition

Ein Knoten vom Typ "Transition" beschreibt den Übergang zwischen zwei Seiten eines Assistenten. Dieser Knoten setzt sich ausschließlich aus einem Block von Eigenschaften zusammen.

 Anmerkung:

Die Definition von Übergängen kann innerhalb eines Knotens vom Typ "Page" oder eines Knotens vom Typ "Root" erfolgen. Bei Übergängen innerhalb eines Knotens vom Typ Page ist die Eigenschaft "FROM" nicht erforderlich. Der letzte Übergang vor dem Schließen des Assistenten wird im Knoten "Finish" (auf der Ebene eines Knotens vom Typ "Root") beschrieben und verfügt weder über die Eigenschaft "FROM" noch über die Eigenschaft vom Typ "TO".

## Syntax eines Knotens vom Typ "Transition"

Die Syntax eines Knotens vom "Typ" Transition lautet wie folgt:

```
' Beschreibung des Übergangs
{ TRANSITION <Name des Übergangs>
' Block mit den Eigenschaften des Knotens vom Typ "Transition"
FROM=...
TO=...
CONDITION=...
}
```

## Eigenschaften eines Knotens vom Typ Transition

Die nachstehende Tabelle führt alle allgemeinen Eigenschaften auf, die in einem Knoten vom Typ "Transition" definiert werden können:

**Tabelle 14.8. Logische Eigenschaften eines Knotens vom Typ "Transition"**

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>	<b>Kommentar</b>
FROM="<Name der ursprünglichen Seite>"	Definiert den Namen der ursprüngliche Seite des Übergangs. Eigenschaft vom Typ "String".	FROM="Page2"	Diese Eigenschaft ist obligatorisch, wenn der Übergang innerhalb eines Knotens vom Typ "Root" definiert wurde, und nicht anwendbar, wenn der Übergang innerhalb eines Knotens vom Typ "Page" oder "Finish" oder "Start" definiert wurde.
TO="<Name der Zielseite>"	Definiert den Namen der Zielseite des Übergangs. Eigenschaft vom Typ "String".	TO="Page3"	Diese Eigenschaft ist obligatorisch, wenn der Übergang innerhalb eines Knotens vom Typ "Root" oder "Page" definiert wurde, und nicht anwendbar, wenn der Übergang innerhalb eines Knotens vom Typ "Finish" definiert wurde.
CONDITION=<Skript>	Definiert die Bedingung, die zum Aktivieren des Übergangs erfüllt werden muss. Eigenschaft vom Typ "Skript", die einen booleschen Wert zurückgibt.	CONDITION={Comment}="user"	Eigenschaft, die in einem Knoten vom Typ "Start" nicht verfügbar ist.
DO=<Skript>	Definiert ein Skript, das beim Übergang ausgeführt werden muss. Eigenschaft vom Typ "Script".	{DO= Filter=""}	

## Besonderheiten eines Knotens vom Typ Transition

Ein Knoten vom Typ "Transition" verfügt über keinerlei Nebenknoten.

## Wozu dienen die Übergänge in Knoten vom Typ "Root"?

Mit Hilfe von Übergängen in Knoten vom Typ "Page" können Sie in jedem beliebigen Assistenten wiederverwendbare Seiten erstellen und das Schreiben von Skripten leichter gestalten.

## Definition eines Knotens vom Typ Finish

Mit einem Knoten vom Typ "Finish" wird der letzte Übergang beschrieben, der zur letzten Seite des Assistenten führt. Es handelt sich also um einen Sonderfall des Knotens vom Typ "Transition", da er weder über die Eigenschaft "FROM" noch über die Eigenschaft "TO" verfügt. Bis auf diese eine Ausnahme sind die Syntax und die Eigenschaften des Knotens vom Typ "Finish" identisch mit der Syntax und den Eigenschaften des Knotens vom Typ "Transition".

Die Eigenschaft CHAIN, die nur für den Knoten vom Typ Finish festgelegt werden kann, ermöglicht das Auslösen eines anderen Assistenten.

**Tabelle 14.9. Logische Eigenschaften eines Knotens vom Typ "Finish"**

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>	<b>Kommentar</b>
CHAIN=<SQL-Name des auszuführenden Assistenten>	Definiert den SQL-Namen des Assistenten, der am Ende des aktuellen Assistenten ausgeführt werden soll.  Wenn diese Eigenschaft nicht definiert wurde, wird kein Assistent ausgeführt.  Eigenschaft vom Typ "String"	CHAIN = "Umzug"	
CONDITION=<Skript>	Definiert die Bedingung, mit der die Schaltfläche <b>Fertig stellen</b> bestätigt wird.  Eigenschaft vom Typ "Skript".		
DO=<Skript>	Definiert das Skript, das am Ende des Assistenten ausgeführt werden muss.  Eigenschaft vom Typ "Skript", das einen booleschen Wert zurückgibt.		

 **Anmerkung:**

Der Knoten vom Typ PARAMS ermöglicht die Weitergabe der Parameter an den Assistenten, der ausgeführt werden soll.

**Tabelle 14.10. Physische Eigenschaft mit einem Knoten vom Typ "Finish"**

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>
----------------------------------	-------------------------------------	-----------------



SUMMARY=<TRUE FALSE>	<p>Zeigt die Seite mit der Zusammenfassung bei der Ausführung des Assistenten an (=TRUE) oder nicht (=FALSE).</p> <p>Über die Funktionen amLog et amProgress können Daten auf dieser Seite eingegeben werden.</p>
SHOWPROGRESS-BAR=<TRUE FALSE>	<p>Zeigt eine Verlaufsanzeige auf der Seite mit der Zusammenfassung an (=TRUE) oder nicht (=FALSE).</p>
SHOWLOG-LIST=<TRUE FALSE>	<p>Zeigt ein Verlaufsprotokoll auf der Seite mit der Zusammenfassung an (=TRUE) oder nicht (=FALSE).</p>
LABEL="Überschrift"	<p>Zeigt eine Überschrift für die Seite mit der Zusammenfassung an.</p>
ISHTML=<TRUE FALSE>	<p>Definiert die Art des Textes der Bezeichnung.</p> <p>Eigenschaft vom Typ "Boolean".</p>
TITLE="Titel"	<p>Definiert einen Titel für die Seite mit der Zusammenfassung</p> <p>Der Standardtitel für die Seite mit der Zusammenfassung entspricht dem Knoten "Root".</p>

## Definition eines Knotens vom Typ Start

Ein Knoten vom Typ "Start" beschreibt den Start des Assistenten. Es handelt sich also um einen Sonderfall des Knotens vom Typ "Transition", der weder über die Eigenschaft "FROM" noch über die Eigenschaft vom Typ "CONDITION" verfügt. Bis auf diese eine Ausnahme sind die Syntax und die Eigenschaften des Knotens vom Typ "Start" identisch mit der Syntax und den Eigenschaften des Knotens vom Typ "Transition".

**Tabelle 14.11. Logische Eigenschaft eines Knotens vom Typ "Start"**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
DO=<Skript>	Definiert das beim Start auszuführende Skript. Eigenschaft vom Typ "Skript", die einen booleschen Wert zurückgibt.	
TO="<Name der ersten Seite>"	Definiert den Namen der ersten Seite, die angezeigt werden soll. Eigenschaft vom Typ "String".	

**Anmerkung:**

Wenn dieser Knoten nicht existiert, startet der Assistent mit der ersten Seite.

## Definition eines Knotens vom Typ Timer

Ein Knoten vom Typ "Timer" ermöglicht das Ausführen einer Aufgabe in regelmäßigen Abständen.

**Tabelle 14.12. Logische Eigenschaften vom Typ "Timer"**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
AUTO= <TRUE FALSE>	Legt fest, ob die Zeituhr beim Anzeigen der Seite automatisch gestartet werden soll. Diese Eigenschaft kann zum Anhalten und Neustarten der Zeituhr verwendet werden. Eigenschaft vom Typ "Boolean".	

ENABLED=<TRUE FALSE>	Legt fest, ob die Zeituhr aktiviert werden soll (=TRUE) oder nicht (=FALSE). Eigenschaft vom Typ "Boolean".
INTERVAL=Frist	Definiert die Dauer des Intervalls (Frist) zwischen zwei Ausführungen der Zeituhr. In Millisekunden ausgedrückte Dauer.
TIMER="Script"	Legt fest, was bei Ablauf der Frist ("INTERVAL") der Zeituhr ausgeführt werden muss. Eigenschaft vom Typ Skript
VALUE=tickcount	Anzahl der abgelaufenen Intervalle. Sämtliche, von diesem Intervall abhängigen Eigenschaften, werden in regelmäßigen Abständen überprüft. Dieser Wert (tickcount) wird automatisch inkrementiert.

## Definition der Knoten vom Typ Long und String

Mit den Knoten vom Typ Long und String werden Variablen definiert. Diese beiden Knoten können in jedem beliebigen Knoten des Assistenten vorhanden sein. Über den Namen des Knotens wird der Name der Variablen festgelegt.

Diese Knoten verfügen nur über eine einzige Eigenschaft, die sich aus dem Typ des Knotens ergibt: Für einen Knoten vom Typ Long entspricht die Eigenschaft dem Typ LONG, und für einen Knoten vom Typ String entspricht die Eigenschaft dem Typ STRING. Mit der Eigenschaft VALUE wird der Wert der Variablen definiert.

**Tabelle 14.13. Logische Eigenschaft vom Typ Long oder String**

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>	<b>Kommentar</b>
VALUE=<Wert>	Definiert den Wert der Variablen, deren Namen mit dem Namen des Knotens identisch ist.  Eigenschaft vom Typ "Long" für einen Knoten vom Typ Long oder vom Typ "String" für einen Knoten vom Typ String.	VALUE=12	

**Anmerkung:**

Die Knoten vom Typ Long und String können in jedem beliebigen Knoten des Assistenten definiert werden. Sie verfügen über keinerlei untergeordneten Knoten.

## Definition eines Knotens vom Typ Steuerelement

Die Steuerelemente einer Seite ermöglichen den Dialogverkehr mit dem Benutzer. Sie können für jede einzelne Seite eine beliebige Anzahl von Steuerelementen definieren. AssetCenter übernimmt das Anordnen dieser Elemente auf der jeweiligen Seite, so dass Sie sich voll und ganz auf ihre Definition konzentrieren können.

Die Knoten vom Typ "Steuerelement" bestehen aus einem einzigen Block von Eigenschaften, der sich auf das definierte Steuerelement bezieht.

## Allgemeine Syntax eines Knotens vom Typ Steuerelement

Die allgemeine Syntax eines Knotens vom Typ "Steuerelement" lautet wie folgt:

```
' Beschreibung des Steuerelements
{ <Typ des Steuerelements> <Name des Steuerelements>
' Eigenschaft des Steuerelements
...
}
```

## Steuerelementtypen mit ihren zugeordneten Eigenschaften

Alle Steuerelemente verfügen über gemeinsame Eigenschaften. Dabei gibt es jedoch einige Eigenschaften, die nur für bestimmte Steuerelemente gelten.

### Gemeinsame Eigenschaften

In der nachstehenden Tabelle werden die optionalen Eigenschaften aufgeführt, die allen Steuerelementen gemein sind:

**Tabelle 14.14. Logische Eigenschaften, die allen Steuerelementen gemein sind**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
MANDATORY=<TRUE FALSE>	Zwingt den Benutzer zur Eingabe von Informationen, um einen Übergang zu bestätigen.  Diese Eigenschaft wird ignoriert, wenn das Steuerelement folgenden Wert aufweist: <ul style="list-style-type: none"> <li>• ausgeblendet</li> <li>• schreibgeschützt</li> <li>• deaktiviert</li> </ul>	MANDATORY=TRUE	Für die Steuerelemente "CHECKBOX" und "LABEL" ist diese Eigenschaft nicht verfügbar.

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
VALUE=<Wert>	<p>Definiert den Standardwert des Steuerelements bei seiner Erstellung. Der &lt;Wert&gt; ergibt sich aus dem betroffenen Steuerelement.</p> <p>Der Typ dieser Eigenschaft ergibt sich aus dem Typ des Steuerelements (Boolean, Text usw.).</p>	<p>Wenn es sich beispielsweise um das Steuerelement "CHECK-BOX" handelt, kann der &lt;Wert&gt; "TRUE" oder "FALSE" entsprechen.</p>	
PERMANENT=<TRUE FALSE>	<p>Die Steuerelemente werden beim Wechsel von einer Seite des Assistenten zur nächsten gelöscht.</p> <p>Definiert, ob das Steuerelement beim Wechsel von einer Seite zur nächsten beibehalten (=TRUE) oder gelöscht (=FALSE) werden soll.</p> <p>Eigenschaft vom Typ "Boolesch".</p>		
SERIALIZE=<TRUE FALSE>	<p>Diese Eigenschaft ermöglicht das Serialisieren des Assistenten auf der Ebene eines Steuerelements.</p> <p>Wenn die Serialisierung eines Knotens vom Typ "Root" aktiv ist (=TRUE), können Sie die Eigenschaft auf der Ebene des Steuerelements deaktivieren.</p>		<p>Standardmäßig weist diese Eigenschaft den Wert der Eigenschaft SERIALIZE für einen Knoten vom Typ "Root" auf.</p>

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
HELP="Hilfe"	Über diese Eigenschaft können Sie im Steuerelement des Assistenten einen Hilfetext im Format HTML einfügen. Der Zugriff erfolgt über die Tastenkombination Umschalttaste+F1.		

**Tabelle 14.15. Physische Eigenschaften, die allen Steuerelementen gemein sind**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
VISIBLE=<TRUE FALSE>	Definiert die Anzeige des Steuerelements (=TRUE) oder nicht (=FALSE). Eigenschaft vom Typ "Boolean".	Label1.Visible=TRUE	
ENABLED = <TRUE FALSE>	Definiert die Aktivierung des Steuerelements (=TRUE) oder nicht (=FALSE). Eigenschaft vom Typ "Boolean".	Choice1.Enabled=FALSE	
READONLY = <TRUE FALSE>	Definiert den Lesezugriff auf das Steuerelement (=TRUE), so dass festgelegt wird, ob der Benutzer das Element ändern darf oder nicht (=FALSE)	READONLY=TRUE	

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
LABEL = "<Text der Beschreibung>"	Definiert einen optionalen Text, der über oder links neben dem Steuerelement erscheint.  Eigenschaft vom Typ "String".	Choice1.Label="Auswahl der Person"	Diese Beschreibung unterstützt das Format HTML.
LABELLEFT	Über diese Eigenschaft können Sie die Bezeichnung des Steuerelements auf die linke Seite setzen.  Bei der Verwendung dieser Eigenschaft ist die Eingabe der Eigenschaft LABELLEFT erforderlich.  Eigenschaft vom Typ "Boolean".		
XOFFSET	Definiert den Abstand, der der Bezeichnung des Steuerelements vorbehalten ist, wenn sich die Bezeichnung aufgrund der Verwendung der Eigenschaft LABELLEFT links neben dem Steuerelement befindet.  Eigenschaft vom Typ <b>twip</b> .		



Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
ISHTML	Definiert die Art des Textes der Bezeichnung. Standardmäßig entspricht die Art des Textes dem Typ HTML. Eigenschaft vom Typ "Boolean".		Diese Beschreibung unterstützt das Format HTML.
INDENT	Diese Eigenschaft ermöglicht das Verschieben des Steuerelements und seiner Bezeichnung nach rechts. Eigenschaft vom Typ <b>twip</b> .		
YOFFSET	Definiert ein Leerzeichen vor dem Steuerelement und seiner Bezeichnung. Eigenschaft vom Typ <b>twip</b> .		
YOFFSET2	Definiert ein Leerzeichen nach dem Steuerelement und seiner Bezeichnung. Eigenschaft vom Typ <b>twip</b> .		

## Das Steuerelement CHECKBOX

Das Steuerelement "CHECKBOX" definiert ein Kontrollkästchen.

### Eigenschaften

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften erkennt das Steuerelement "CHECKBOX" die nachstehende Eigenschaft:

**Tabelle 14.16. Eigenschaft des Steuerelements "CHECKBOX"**

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>
CAPTION="<Text>"	Definiert den Text des Kontrollkästchens.  Dieser Text kann nicht im Format HTML geschrieben werden und darf nur eine einzige Zeile umfassen.  Eigenschaft vom Typ "String".	TEXT="Identifizierung durch den Namen"

## Das Steuerelement COMBOBOX

Das Steuerelement "COMBOBOX" definiert eine eindeutige Auswahl in der Aufzählung vordefinierter Werte.

### Eigenschaften

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften erkennt das Steuerelement "COMBOBOX" die nachstehende Eigenschaft:

**Tabelle 14.17. Physische Eigenschaften des Steuerelements "COMBOBOX"**

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>	<b>Kommentar</b>
VALUES="<Bezeichnung=Wert, Bezeichnung=Wert, Bezeichnung=Wert,...>"	Definiert die Wertepaare ("Bezeichnung"="Wert") für das Steuerelement "Combo". Die "Bezeichnung" definiert den Text, der im Steuerelement angezeigt wird, und der "Wert" definiert den dem Steuerelement zugeordneten Wert, wenn diese "Bezeichnung" vom Benutzer ausgewählt wurde. Eigenschaft vom Typ "String".	VALUES="Tabelle der Vermögen und Lose=asset, Benutzer=user"	Wenn Sie keinen "Wert" definieren, führt AssetCenter automatisch die Zuordnung eines Werts durch. Ein Beispiel: <b>Beispiel</b> VALUES = "A,B,C" entspricht VALUES="A=1,B=2,C=3"

## Das Steuerelement OPTIONBUTTONS

Das Steuerelement "OPTIONBUTTONS" definiert die Gruppe der Optionsfelder.

### Eigenschaften

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften erkennt das Steuerelement "OPTIONBUTTONS" die nachstehenden Eigenschaften:

**Tabelle 14.18. Physische Eigenschaften des Steuerelements "OPTIONBUTTONS"**

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>
VALUES="<Bezeichnung=Wert, Bezeichnung=Wert, Bezeichnung=Wert,....>"	Definiert das Wertepaar ("Bezeichnung"="Wert") für das Steuerelement "CHOICE". "Bezeichnung" definiert den Text für das Optionsfeld, und "Wert" definiert den Wert, der dem Steuerelement zugeordnet wird, wenn der Benutzer dieses Optionsfeld auswählt. Eigenschaft vom Typ "String".	VALUES="Tabelle der Vermögen und Lose=asset, Benutzer=user"
BORDER=<TRUE FALSE>	Legt fest, ob die Gruppe der Optionsfelder umrahmt wird (=TRUE) oder nicht (=FALSE).  Wenn sich die Schaltflächen in einem Rahmen befinden, ist der Text im Rahmenrand integriert. Dieser Text kann weder im Format HTML vorliegen noch mehrere Zeilen umfassen.  Eigenschaft vom Typ "Boolean".	BORDER= TRUE

## Das Steuerelement LISTBOX

Das Steuerelement "LISTBOX" definiert eine Liste auswählbarer Objekte. Die Steuerelemente "LISTBOX" können in mehreren Spalten erscheinen.

### Eigenschaften

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften erkennt das Steuerelement "LISTBOX" die nachstehenden Eigenschaften:

**Tabelle 14.19. Physische Eigenschaften des Steuerelements "LISTBOX"**

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>	<b>Kommentar</b>
LISTHEIGHT = <Prozentsatz>	Definiert die relative Größe des Steuerelements "LISTBOX" im Verhältnis zu den anderen Steuerelementen vom Typ "LISTBOX", die im Assistenten vorhanden sind. Eigenschaft des Typs "Long".	LISTHEIGHT=50	Wenn zwei Steuerelemente vom Typ "LISTBOX" vorhanden sind, und respektive die Werte "10" und "20" für diese Eigenschaft aufweisen, ist das zweite Steuerelement doppelt so hoch wie das erste.
MULTISEL = <TRUE FALSE>	Legt fest, ob das Steuerelement die Mehrfachauswahl unterstützt (=TRUE) oder nicht (=FALSE). Eigenschaft vom Typ "Boolean".	MULTISEL=TRUE	
COLTITLE = "<Spalte Spalte Spalte...>"	Definiert die Spaltenüberschrift in der Liste. Ersetzen Sie "Spalte" durch die Spaltenüberschrift. Eigenschaft vom Typ "String".	COLTITLE = "Name Vorname"	
COLWIDTH = "<Breite Breite Breite...>"	Definiert die Spaltenbreite proportional zur Breite des Steuerelements. Eigenschaft vom Typ "String".	COLWIDTH = "50 50"	

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
VALUES = "<Text Text ...=Wert, Text Text ...=Wert,...>"	Definiert die Wertepaare ("Text Text ..."="Wert") für das Steuerelement "LISTBOX". "Text Text ..." definiert den Text, der in jeder einzelnen Spalte für eine Zeile des Steuerelements "LISTBOX" erscheinen soll, und "Wert" den Wert, der dem Steuerelement zugeordnet wird, wenn der Benutzer diese Zeile auswählt. Eigenschaft vom Typ "String".	VALUES="Tabelle der Vermögen und Lose=asset, , Benutzer=user,"	Wenn Sie keinen "Wert" definieren, führt AssetCenter automatisch die Zuordnung eines Werts durch. Ein Beispiel: VALUES="A,B,C" entspricht VALUES="A=1,B=2,C=3" <b>Beispiel</b> Diese Eigenschaft kann direkt mit Hilfe der Funktion AmdbGetList eingegeben werden. Schreiben Sie zum Beispiel: VALUES= AmDbGetList ("SELECT Name, FirstName FROM amEmplDept WHERE Name Like 'A%', " , ", "=") Achten Sie darauf, die Werte "VALUES" und "VALUE" nicht zu verwechseln.
EDITABLE="<0 1>"	Legt fest, ob der Text in den Spalten bearbeitbar sein soll oder nicht. Eigenschaft vom Typ "String".	EDITABLE="0 1"	
TABLE="<Name der Tabelle>"	Definiert den Anwendungskontext des Spaltentitels, wenn dieser über die Eigenschaft "COLNAME" definiert wurde.	TABLE="amEmplDept"	

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
COLNAME="<Name Name Name>"	Definiert den Titel und die Eigenschaften der Listenspalten mit den SQL-Namen der Felder. Die Eigenschaft "TABELLE" muss definiert sein.  Ersetzt "Name" durch den SQL-Namen des Felds für den Spaltentitel.  Wenn der Spaltentitel über die Eigenschaft "COLTITLE" eingegeben wurde, hat diese Eigenschaft Vorrang vor der Eigenschaft "COLNAME", behält den Typ des SQL-Felds jedoch bei (Text, Datum usw.).	COLNAME="Name FirstName dtHi-re"	
MULTISEL = <TRUE FALSE>	Definiert für eine Liste die Verwendung der Mehrfachauswahl.  Eigenschaft vom Typ "Boolean".	MULTISEL=1	

**Tabelle 14.20. Methoden des Steuerelements "LISTBOX"**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
VALUES (i)	Gibt den Inhalt der Zeile (i) zurück	a b c	
VALUES.COUNT()	Berechnet die Anzahl der Zeilen in der Eigenschaft "VALUES".	retval = {listbox1.values.count()}	

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>	<b>Kommentar</b>
VALUES.CELL(h,v)	Gibt den Inhalt der Zellen zurück, die mit den entsprechenden Koordinaten (horizontal, vertikal) gekennzeichnet sind.	VALUES.CELL(2,4)	
VALUES.COLUMN(i)	Gibt den Inhalt Wert der Spalte (i) zurück. Wenn i=0 oder leer ist, gibt diese Anweisung die Kennungen (ID) der Spalte zurück.	VALUES.COLUMN(1)	
VALUES.LINE(i)	Gibt den Inhalt (Wert) der Zeile (i) zurück. Wenn i=0 oder leer ist, gibt diese Anweisung die Kennungen (ID) der Zeile zurück.	VALUES.LINE(1)	
VALUES.SORT(iCol, bAsc)	Sortiert die Spalte (i) in auf- oder absteigender Reihenfolge (bAsc=1 oder bAsc=0)	{ LISTBOX lb VALUES = "erster,zweiter,dritter" } { COMMANDBUTTON btn { CLICK =RetVal = {lb.Values.Sort(1)} } }	
VALUES (i,0)	Gibt den Wert der Zeilen-ID zurück (i).		

**Tabelle 14.21. Obligatorische, logische Eigenschaft des Steuerelements "LISTBOX"**

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>
TABLE=<Name der Tabelle>	Name der Tabelle, die zum Extrahieren der Spaltenüberschriften verwendet wird. Eigenschaft vom Typ "String".	TABLE= <b>amAsset</b>



Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
COLNAME=<Titel Titel ...>	<p>Definiert die Spaltenüberschrift anhand des SQL-Namens der Felder der Tabelle, die über die Eigenschaft "TABLE" definiert wird. Diese Eigenschaft ermöglicht das Definieren der zur Bearbeitung verwendeten Steuerelemente. Das Steuerelement ist identisch mit dem Element, das von AssetCenter zum Abrufen von Daten in das Feld verwendet wird.</p> <p>AssetCenter greift bevorzugt auf die Werte der Eigenschaft "COLTITLE" zurück, um die Spaltenüberschrift zu definieren, sofern eine solche Überschrift vorhanden ist.</p> <p>Eigenschaft vom Typ "String".</p>	COLNAME="Name FirstName"

## Das Steuerelement LABEL

Das Steuerelement "LABEL" definiert lediglich eine Bezeichnung. Dieses Element verfügt über die nachstehende Eigenschaft:

**Tabelle 14.22. Physische Eigenschaften des Steuerelements "LABEL"**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
CAPTION=<Text>	Enthält den Text, der für das Etikett angezeigt werden soll.	CAPTION="Wählen Sie einen Standort"	

## Das Steuerelement PROGRESSBAR

Das Steuerelement PROGRESSBAR definiert die Verlaufsanzeige.

## Eigenschaften

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften erkennt das Steuerelement "PROGRESSBAR" die nachstehende Eigenschaft:

**Tabelle 14.23. Physische Eigenschaften des Steuerelements "PROGRESSBAR"**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
MAXVALUE=<Max. Wert>	Definiert den maximalen Wert, der in der Verlaufsanzeige 100% entspricht. Die Eigenschaft "VALUE" zeigt den aktuellen Wert des Steuerelements an. Eigenschaft des Typs "Long".	MAXVALUE=200

## Das Steuerelement COMMANDBUTTON

Das Steuerelement "COMMANDBUTTON" definiert eine Aktionsschaltfläche.

### Eigenschaften

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften erkennt das Steuerelement "COMMANDBUTTON" die nachstehenden Eigenschaften:

**Tabelle 14.24. Physische Eigenschaften des Steuerelements "COMMANDBUTTON"**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
WIDTH=<Breite>	Definiert die Breite der Schaltfläche in Twips. Eigenschaft des Typs "Long".	WIDTH=250
HEIGHT=<Höhe>	Definiert die Höhe der Schaltfläche in Twips. Eigenschaft des Typs "Long".	HEIGHT=125

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
CAPTION=<Text>	Definiert den Text (nicht HTML) innerhalb der Schaltfläche. Eigenschaft vom Typ "String".	CAPTION="Starten"
CLICK=<Script BASIC>	Definiert das BASIC-Skript, das ausgeführt wird, wenn der Benutzer auf die Schaltfläche klickt.	

## Das Steuerelement DBLISTBOX

Das Steuerelement "DBLISTBOX" definiert eine Liste von Datensätzen, die in der AssetCenter-Datenbank ausgewählt werden kann. Das Steuerelement kann mehrere Spalten aufweisen. Die im Steuerelement angezeigte Liste ist das Ergebnis einer teilweisen AQL-Abfrage in der AssetCenter-Datenbank, bei der nur die Klausel WHERE verwendet wird.

 Anmerkung:

Die Eigenschaft "VALUE" gibt die Liste der Kennungen ("Id") der ausgewählten Zeilen zurück. Der Zugriff auf die Werte in den Zellen der Liste ist nicht möglich. Sie müssen zu diesem Zweck entweder eine andere Abfrage ausführen oder ein Steuerelement vom Typ "LISTBOX" verwenden.

## Eigenschaften

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften erkennt das Steuerelement "DBLISTBOX" die nachstehenden Eigenschaften:

**Tabelle 14.25. Physische Eigenschaften des Steuerelements "DBLISTBOX"**

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>	<b>Kommentar</b>
TABLE="<SQL-Name der Tabelle>"	Definiert die Tabelle, in der die Abfrage durchgeführt werden soll.  Eigenschaft vom Typ "String".	TABLE=amAsset	Diese Eigenschaft ist obligatorisch.
COLNAME="<SQL-Name des Felds oder der Verknüpfung oder Verknüpfung SQL-Name des Felds oder der Verknüpfung ...>"	Definiert die Daten, die aus der Datenbank extrahiert werden sollen. Die Identifizierung der Daten erfolgt anhand der SQL-Namen.  Eigenschaft vom Typ "String".	COLNAME = "Name FirstName"	
COLWIDTH = "<Breite Breite Breite ...>"	Definiert die Spaltenbreite in der Datenbank, und zwar als Prozentsatz der Gesamtgröße des Steuerelements "DBLISTBOX".  Eigenschaft vom Typ "String".	COLWIDTH="40 60"	
LISTHEIGHT = <Prozentsatz>	Definiert die relative Größe des Steuerelements "DBLISTBOX" im Verhältnis zu den anderen Steuerelementen vom Typ "DBLISTBOX", die im Assistenten vorhanden sind.  Eigenschaft des Typs "Long".	LISTHEIGHT=50	Wenn zwei Steuerelemente vom Typ "DBLISTBOX" vorhanden sind, und respektive die Werte "10" und "20" für diese Eigenschaft aufweisen, ist das zweite Steuerelement doppelt so hoch wie das erste.

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
TREE=<TRUE FALSE>	Legt fest, ob das Steuerelement die Mehrfachauswahl (=TRUE) unterstützt oder nicht (=FALSE). Eigenschaft vom Typ "Boolean".	TREE=TRUE	Standardmäßig erscheint für diese Eigenschaft der Wert "FALSE".
MULTISEL = <TRUE FALSE>	Legt fest, ob das Steuerelement die Mehrfachauswahl unterstützt (=TRUE) oder nicht (=FALSE). Eigenschaft vom Typ "Boolean".	MULTISEL=TRUE	
DBLCLICK = <TRUE FALSE>	Wenn für diese Eigenschaft der Wert TRUE gilt, simuliert AssetCenter bei einem Doppelklick auf dieser Zeile einen Mausklick auf die Schaltfläche <b>Weiter</b> der aktuellen Seite.	DBLCLICK=FALSE	
FILTER = "<Klausel>"	Definiert die AQL-Klausel "WHERE", um die Datensätzen zu filtern, die bei der Abfrage bearbeitet werden sollen. Eigenschaft vom Typ "String".	FILTER = "User.ImplDeptId='Colombo, Gérard' "	

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
MAXSEL = <TRUE FALSE>	Definiert die Möglichkeit, mehr als 99 Elemente einer AQL-Abfrage zu wählen (=TRUE) oder nicht (=FALSE).  Standardmäßig ist die Auswahl auf 99 Elemente begrenzt (=TRUE).		
VALUES.SORT(iCol, bAsc)	Sortiert die Spalte (iCol) in auf- oder absteigender Reihenfolge (bAsc=1 oder bAsc=0)	VALUES.SORT(2, 0)	
[Value.]ISSELECTION()	Wenn diese Auswahl andere Werte enthält als 0, der Benutzer also keine Auswahl in der Liste vorgenommen hat, wird bei Verwendung dieser Methode ein Wert ungleich 0 zurückgegeben.	MANDATORY = not {dblistbox1.IsSelection()}  (ein Übergang ist nicht Möglich, wenn der Benutzer keine Auswahl in der Liste getroffen hat)	

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
TABLE.LABEL([iNameType])	<p>Diese Methode gibt die Überschrift der Tabelle zurück, in der das Steuerelement definiert ist.</p> <p>Im Folgenden sind die Überschriftentypen (iNameType) aufgeführt:</p> <ul style="list-style-type: none"> <li>• 1 - Systemname</li> <li>• 2 - SQL-Name</li> <li>• 3 - Standardüberschrift</li> <li>• 4 - Beschreibung (Hilfetext)</li> </ul> <p>Die Eigenschaft "TABLE" muss eingegeben worden sein.</p>	RetVal = {dblist-box1.table.label(2)}	

## Das Steuerelement DBQUERYBOX

Das Steuerelement "DBQUERYBOX" definiert eine Liste von Datensätzen, die in der AssetCenter-Datenbank ausgewählt werden können. Das Steuerelement kann aus mehreren Spalten bestehen. Die im Steuerelement angezeigte Liste ist das Ergebnis einer vollständigen AQL-Abfrage in der AssetCenter-Datenbank.

### Eigenschaften

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften erkennt das Steuerelement "DBQUERYBOX" die nachstehenden Eigenschaften:

**Tabelle 14.26. Physische Eigenschaften des Steuerelements "DBQUERYBOX"**

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>	<b>Kommentar</b>
QUERY="<Vollständige AQL-Abfrage>"	Definiert die AQL-Abfrage, die die im Steuerelement "DB-LIST" anzuzeigenden Informationen zurückgibt. Eigenschaft vom Typ "String".	QUERY="SELECT Name, FirmenName FROM amEmplDept WHERE Location='Meteor-Gebäude'"	
COLTITLE="<Spalte Spalte ...>"	Definiert die Spaltenüberschrift in der Liste. Eigenschaft vom Typ "String".	COLTITLE = "Name FirstName"	
COLWIDTH="<Breite Breite ...>"	Definiert die Breite der Spalte proportional zur Breite des Steuerelements. Die Angaben erfolgen in Prozenten. Eigenschaft vom Typ "String".	COLWIDTH = "50 50"	



Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
LISTHEIGHT=<Prozentsatz>	Definiert die relative Größe des Steuerelements "DBQUERY-BOX" im Verhältnis zu den anderen Steuerelementen vom Typ "DBQUERY-BOX", die im Assistenten vorhanden sind. Eigenschaft des Typs "Long".	LIST-HEIGHT=50	Wenn zwei Steuerelemente vom Typ "DBQUERY-BOX" vorhanden sind, und respektive die Werte "10" und "20" für diese Eigenschaft aufweisen, ist das zweite Steuerelement doppelt so hoch wie das erste.
TREE=<TRUE FALSE>	Legt fest, ob das Steuerelement die Mehrfachauswahl (=TRUE) unterstützt oder nicht (=FALSE). Eigenschaft vom Typ "Boolean".	TREE=TRUE	Standardmäßig erscheint für diese Eigenschaft der Wert "FALSE".
MAXSEL = <TRUE FALSE>	Definiert die Möglichkeit der Auswahl von mehr als 99 Elementen (=TRUE) oder nicht (=FALSE). Standardmäßig ist die Auswahl auf 99 Elemente begrenzt (=TRUE).		

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
MULTISEL=<TRUE FALSE>	Legt fest, ob das Steuerelement die Mehrfachauswahl unterstützt (=TRUE) oder nicht (=FALSE). Eigenschaft vom Typ "Boolean".	MULTI-SEL=TRUE	
DBLCLICK=<TRUE FALSE>	Wenn für diese Eigenschaft der Wert TRUE gilt, simuliert AssetCenter bei einem Doppelklick auf diese Zeile einen Mausklick auf die Schaltfläche <b>Weiter</b> der aktuellen Seite. Eigenschaft vom Typ "Boolean".	DBLCLICK=FAL-SE	
[Value.]ISSELECTION()	Wenn diese Auswahl andere Werte enthält als 0, der Benutzer also keine Auswahl in der Liste vorgenommen hat, wird bei Verwendung dieser Methode ein Wert ungleich 0 zurückgegeben.		


## Das Steuerelement DBEDIT

Das Steuerelement "DBEDIT" erstellt ein weiteres Steuerelement, das identisch ist mit dem Element, das zum Abrufen von Daten in ein Feld der

AssetCenter-Datenbank verwendet wird. Je nach Feldtyp gilt ein anderes Steuerelement (Datum, Währung, ...).



**Anmerkung:**

Die Schaltfläche  neben diesem Steuerelement ermöglicht die Auswahl der Werte, die tatsächlich in der Datenbank vorhanden sind. Der Benutzer kann jedoch auch einen anderen Wert eingeben.

Für dieses Steuerelement gilt eine gemeinsame Eigenschaft "VALUE" vom Typ "Variant" (in Abhängigkeit vom Steuerelement).

## Eigenschaften

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften muss das Steuerelement "DBLIST" obligatorisch die nachstehenden Eigenschaften aufweisen:

**Tabelle 14.27. Physische Eigenschaften des Steuerelements "DBEDIT" im Modus "Normal"**

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>
TABLE="<SQL-Name der Tabelle>"	SQL-Name der Ausgangstabelle Eigenschaft vom Typ "String".	TABLE="amAsset"
FIELD="<SQL-Name des Felds>"	AQL-Name des Felds, das für das Steuerelement verwendet wird. Eigenschaft vom Typ "String".	FIELD="seAcquMethod"

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
TABLE.LABEL([iNameType])	<p>Diese Methode gibt die Überschrift einer bestimmten Tabelle zurück.</p> <p>Im Folgenden sind die Überschriftentypen (iNameType) aufgeführt:</p> <ul style="list-style-type: none"> <li>• 1 - Systemname</li> <li>• 2 - SQL-Name</li> <li>• 3 - Standardüberschrift</li> <li>• 4 - Beschreibung (Hilfertext)</li> </ul> <p>Die Eigenschaften "TABLE" und "NAME" müssen eingegeben worden sein.</p>	<p>► Physische Eigenschaften des Steuerelements "DBQUERY-BOX" [Seite 312]</p>
FIELD.LABEL([iNameType])	<p>Diese Methode gibt die Überschrift eines bestimmten Felds zurück.</p> <p>Im Folgenden sind die Überschriftentypen (iNameType) aufgeführt:</p> <ul style="list-style-type: none"> <li>• 1 - Systemname</li> <li>• 2 - SQL-Name</li> <li>• 3 - Standardüberschrift</li> <li>• 4 - Beschreibung (Hilfertext)</li> </ul> <p>Die Eigenschaft "TABLE" muss eingegeben worden sein.</p>	

## Das Steuerelement DBTABLE

Das Steuerelement "DBTABLE" erstellt ein weiteres Steuerelement für die Eingabe einer Tabelle der AssetCenter-Datenbank.

### Eigenschaften

Dieses Steuerelement verfügt über keinerlei zusätzliche Eigenschaften.

## Das Steuerelement DBPATH

Das Steuerelement "DBPATH" erstellt ein weiteres Steuerelement für die Eingabe eines Pfads der AssetCenter-Datenbank.

### Eigenschaften

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften muss das Steuerelement "DBPATH" obligatorisch die nachstehenden Eigenschaften aufweisen:

**Tabelle 14.28. Obligatorische, physische Eigenschaften des Steuerelements "DBPATH"**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
TABLE=<SQL-Name der Tabelle>"	Name der Tabelle, in der ein Pfad ausgewählt werden soll. Eigenschaft vom Typ "String".	TABLE= <b>amAsset</b>

## Das Steuerelement LINKEDIT


Das Steuerelement "LINKEDIT" erstellt ein weiteres Steuerelement für die Eingabe einer Verknüpfung der AssetCenter-Datenbank

### Eigenschaften

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften verfügt das Steuerelement "LINKEDIT" über die nachstehenden Eigenschaften:

**Tabelle 14.29. Logische Eigenschaft des Steuerelements "LINKEDIT"**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel / Kommentar
TABLE=<SQL-Name der Tabelle>"	Name der Tabelle, in der eine Verknüpfung ausgewählt werden soll. Eigenschaft vom Typ "String".	TABLE="amAsset"
FILTER=<WHERE-Klausel einer AQL-Klausel>	Definiert einen Filter AQL-Filter. Eigenschaft vom Typ "String".	Diese Eigenschaft ist optional.

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel / Kommentar
LINK=<SQL-Name der Verknüpfung>"	SQL-Name einer Verknüpfung der Tabelle, die in der Eigenschaft 'TABLE' definiert ist.  Optionale Eigenschaft.	LINK="POrLIne"
ZOOM=<TRUE FALSE>	Zeigt die Lupe an (=TRUE) oder nicht (=FALSE).  Diese Eigenschaft gilt nur für nicht modale Assistenten (Eigenschaft MODAL=FALSE auf der Ebene des Knotens "Root").	
SRCCHOICE=<TRUE FALSE>	Zeigt das Symbol  an (=TRUE) oder nicht (=FALSE).  Diese Eigenschaft gilt nur für nicht modale Assistenten (Eigenschaft MODAL=FALSE auf der Ebene des Knotens "Root").	
TABLE.LABEL([iNameType])	Diese Methode gibt die Überschrift der Ausgangstabelle der Verknüpfung zurück.  Im Folgenden sind die Überschriftentypen (iNameType) aufgeführt: <ul style="list-style-type: none"> <li>• 1 - Systemname</li> <li>• 2 - SQL-Name</li> <li>• 3 - Standardüberschrift</li> <li>• 4 - Beschreibung (Hilfertext)</li> </ul> Die Eigenschaft "TABLE" muss eingegeben worden sein.	► Physische Eigenschaften des Steuerelements "DBQUERY-BOX" [Seite 312]

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel / Kommentar
LINK.LABEL([iNameType])	<p>Diese Methode gibt die Überschrift der Verknüpfung zurück.</p> <p>Im Folgenden sind die Überschriftentypen (iNameType) aufgeführt:</p> <ul style="list-style-type: none"> <li>• 1 - Systemname</li> <li>• 2 - SQL-Name</li> <li>• 3 - Standardüberschrift</li> <li>• 4 - Beschreibung (Hilfertext)</li> </ul> <p>Die Eigenschaften "TABLE" und "LINK" müssen eingegeben worden sein.</p>	

## Das Steuerelement TEXTBOX

Das Steuerelement "TEXTBOX" erstellt ein Steuerelement für die Eingabe eines Textes.

### Eigenschaften

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften kann das Steuerelement "TEXTBOX" über die nachstehenden Eigenschaften verfügen:

**Tabelle 14.30. Physische Eigenschaft des Steuerelements "TEXTBOX"**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
MULTILINE=<Anzahl>	Für diese Eigenschaft erscheint der Wert 0, wenn das Steuerelement "TEXTBOX" nur eine Zeile aufweist. Wenn das Steuerelement mehrere Zeilen aufweist, erscheint ein numerischer Wert, der die angezeigte Höhe des Steuerelements als Prozentsatz ausdrückt.	MULTILINE=50

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
PASSWORD=<TRUE FALSE>	Diese Eigenschaft blendet den eingegebenen Text aus (=TRUE) oder nicht (=FALSE).	

## Das Steuerelement CHART

Das Steuerelement "CHART" ermöglicht das Anzeigen einer Grafik, die sich aus mehreren Serien zusammensetzen kann.

### Eigenschaften

Neben den optionalen Eigenschaften, die allen Steuerelementen gemein sind, besitzt das Steuerelement "CHART" folgende Eigenschaften:

**Tabelle 14.31. Logische Eigenschaften des Steuerelements "CHART"**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
SERIES	Definiert die Liste der Seriennamen, wobei die Namen durch einen senkrechten Strich voneinander getrennt sind.  Diese Liste darf nicht leer sein, da sonst keine Elemente in der Grafik erscheinen.	SERIES="Einkaufspreis Verkaufspreis"
VALUES	Definiert die numerischen Werte der Grafikserien.  Zweidimensionale Tabelle	VALUES="1 2,1 4"
FORMAT	Definiert den Datentyp: <ul style="list-style-type: none"> <li>• Lange Ganzzahl (Long integer)</li> <li>• Doppelte Gleitkommazahl (Double Floating)</li> <li>• Nummer (Number)</li> <li>• Prozentsatz (Percent)</li> </ul>	



Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
SERIE	Nummer der Serie, auf die Sie geklickt haben Die Eigenschaft CHART muss dem Dialogverkehr entsprechen (=TRUE).	
INDEX	Nummer der Spalte, auf die Sie geklickt haben. Die Eigenschaft CHART muss dem Dialogverkehr entsprechen (=TRUE).	
CLICK	Ruft das Skript dieser Eigenschaft beim einem Mausklick auf die Grafik auf.	

**Tabelle 14.32. Physische Eigenschaften des Steuerelements "CHART"**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
MODE	Definiert den Grafiktyp: <ul style="list-style-type: none"> <li>• MODE=0: vertikale Striche</li> <li>• MODE=1: horizontale Striche</li> <li>• MODE=3: Sektoren</li> </ul>	
LABELS	Definiert den Spaltentitel.	Januar Februar
3D	Legt fest, ob die Grafik in 3D angezeigt wird (=TRUE) oder nicht (=FALSE).	
COLORS	Definiert die Liste der Farben für jede Serie. Die Elemente der Liste sind durch einen vertikalen Strich voneinander getrennt. RGB-Wert mit Dezimal stellen.	255 16777215 16711680 Zeigt die Farben Blau, Weiß und Rot an.
INTERACTIVE	Legt fest, ob die Grafik interaktiv sein soll (=TRUE) oder nicht (=FALSE), also ob sie beim Darübergleiten des Mauszeiger aktiviert werden soll oder nicht.	
POPUP	Zeigt das Popup-Menü an (=TRUE) oder nicht (=FALSE).	

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
BACKGROUND	Definiert, ob es für die Grafik einen Hintergrund gibt (=TRUE) oder nicht (=FALSE).	
BACKIMAGE	Definiert den Pfad des Bild, das der Grafik als Hintergrund dient. Die Eigenschaft "BACKGROUND" muss bestätigt werden (=TRUE), um das Anzeigen des Hintergrunds zu ermöglichen.	
STACKED	Legt fest, ob die grafischen Striche übereinandergelegt werden sollen (TRUE) oder nicht (=FALSE).	
CHART-HEIGHT	Definiert die relative Größe des Steuerelements "CHART" im Vergleich zu den anderen Steuerelementen des Assistenten.	
CAPTION	Zeigt den Titel an.	
ELEVATION	Definiert die Neigung der 3D-Ansicht für die Sektoren.	
ROTATION	Definiert den Rotationswinkel für die Sektorgrafik. In Grad ausgedrückter Wert.	
DISPLAYLABELS	Zeigt die Spaltentitel (LABELS) an (=TRUE) oder nicht (=FALSE).	
DISPLAYSLegend	Zeigt die Legende der Serien an (=TRUE) oder nicht (=FALSE).	

## Das Steuerelement FILEEDIT

Das Steuerelement zeigt das Dialogfeld an, mit dem eine Datei oder ein Verzeichnis gespeichert oder geladen werden können.

**Tabelle 14.33. Eigenschaften des Steuerelements "FILEEDIT"**

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>
OPENMODE	Definiert den Typ des Dialogfelds: <ul style="list-style-type: none"> <li>• OPENMODE=1: Datei öffnen</li> <li>• OPENMODE=2: Datei speichern</li> <li>• OPENMODE=4: Verzeichnis öffnen</li> <li>• OPENMODE=8: Verzeichnis speichern</li> </ul>	
FILTERS	Definiert die Anzeigekriterien für die im Dialogfeld aufgeführten Dateien.	(*.* *.txt *.scn) *.scn
DEFEXT	Definiert die Standarderweiterung der Datei.	(*.*) *.scn

## Das Steuerelement TICKEDIT

Dieses Steuerelement ermöglicht das Einfügen eines Steuerprogramms.

**Tabelle 14.34. Eigenschaften des Steuerelements TICKEDIT**

<b>Name der Eigenschaft=Wert</b>	<b>Beschreibung der Eigenschaft</b>	<b>Beispiel</b>
VALUE	Darstellung der vom Benutzer im Steuerprogramm definierten Parameter in Form eines Strings.	
LISTHEIGHT	Definiert die relative Höhe des Steuerelements "TICKEDIT" im Vergleich zu den anderen im Assistenten vorhandenen Steuerelemente. Eigenschaft vom Typ "Long".	

## Das Steuerelement CALENDAR

Dieses Steuerelement ermöglicht das Einfügen eines Kalenders.

## Das Steuerelement TIMESPANEDIT

Dieses Steuerelement ermöglicht das Einfügen einer Eingabezone für die Dauer.

## Das Steuerelement NUMBOX

Dieses Steuerelement ermöglicht das Einfügen eines Steuerelements vom Typ Zahl.

**Tabelle 14.35. Eigenschaften des Steuerelements NUMBOX**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
MINVALUE	Definiert den minimalen Wert der Zahl. Standardmäßig ein unendlicher Wert.	
MAXVALUE	Definiert den maximalen Wert der Zahl Standardmäßig ein unendlicher Wert.	
FORMAT	Definiert das Format der Zahl: <ul style="list-style-type: none"> <li>• LONG in dem Format, das in der Systemsteuerung des Betriebssystems definiert ist.</li> <li>• RAWLONG</li> <li>• DOUBLE in dem Format, das in der Systemsteuerung des Betriebssystems definiert ist.</li> </ul>	

## Das Steuerelement COMBOEDIT

Dieses Steuerelement ermöglicht das Einfügen eines Steuerelements vom Typ Dropdown-Liste.

**Tabelle 14.36. Eigenschaft des Steuerelements COMBOEDIT**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
VALUES	Definiert Wertepaare für das Steuerelement. "Titel" definiert den Text, der im Steuerelement angezeigt ist und "Wert" definiert den Wert, der dem Steuerelement zugeordnet ist, wenn "Titel" vom Benutzer ausgewählt wurde. Eigenschaft vom Typ "String".	

## Das Steuerelement DATETIMEEDIT

Dieses Steuerelement ermöglicht das Einfügen eines Steuerelements vom Typ Datum.

**Tabelle 14.37. Eigenschaften des Steuerelements**

Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
FORMAT	Definiert das Format des Steuerelements: <ul style="list-style-type: none"> <li>• Datum (date)</li> <li>• Uhrzeit (time)</li> <li>• Datum und Uhrzeit</li> </ul> Das Format des Steuerelements ist abhängig von den vom Benutzer definierten Systemparametern.	2002/02/07 13:37:19 2002/02/07 13:37:19

## Arbeiten mit dem Grafikeditor

AssetCenter bietet die Möglichkeit, die Assistenten mit Hilfe eines integrierten Grafikeditors zu erstellen. Dieser Editor soll das Schreiben eines Assistenten erleichtern und beschleunigen, ist jedoch unter keinen Umständen als Ersatz der Skriptsprache zu verstehen, deren Verwendung eine entsprechende Fachkenntnis voraussetzt.

- Überblick über die Benutzeroberfläche zur Bearbeitung des Assistenten
- Neue Knoten erstellen
- Eigenschaften eines Knotens bearbeiten
- Assistenten ausführen und Fehler beheben

 **Anmerkung:**

Wenn Sie den Grafikeditor verwenden möchten, muss die Aktion beim Erstellen und Ändern dem Typ "Assistent" entsprechen.

## Überblick über die Benutzeroberfläche zur Bearbeitung des Assistenten

Verwenden Sie das Menü **Extras/ Aktionen/ Bearbeiten**, um auf die grafische Benutzeroberfläche zur Bearbeitung des Assistenten zuzugreifen. Der Grafikeditor erscheint auf der Registerkarte **Assistent** im Detailfenster der Aktion. Die Registerkarte setzt sich aus drei verschiedenen Abschnitten zusammen:





- In der Symbolleiste sind die am häufigsten verwendeten Funktionen zusammengefasst.
- Der Abschnitt **Hierarchie** zeigt die Baumstruktur des Assistenten.
- Im rechten Bereich erscheinen die Eigenschaften der Knoten, die in der Baumstruktur ausgewählt sind.

### Symbolleiste

Klicken Sie auf ein Symbol in dieser Leiste, um den entsprechenden Befehl zur Bearbeitung des Assistenten zu aktivieren. Wenn die Maus über eines dieser Symbole gleitet, wird der entsprechende Befehl in einem Hilfetext angezeigt.

### Befehle zur Bearbeitung

Dem Benutzer stehen vier Schaltflächen zur Verfügung:

-  schaltet zwischen dem Text- und dem Grafikmodus des Editors um.
-  versetzt den gewählten Knoten innerhalb des übergeordneten Knotens um eine Ebene nach oben.
-  versetzt den gewählten Knoten innerhalb des übergeordneten Knotens um eine Ebene nach unten.
-  löscht den gewählten Knoten.

### Befehle zur Ausführung und Fehlerbehebung

Mit Hilfe dieser Befehle kompilieren Sie das Skript des Assistenten, um es ausführen zu können und dabei eventuell vorhandene Fehler zu erkennen:

**Abbildung 14.2. Schaltflächen zur Ausführung und Fehlerbehebung**



## Suchwerkzeug

Die Symbolleiste enthält ein Suchwerkzeug, mit dem Sie Zeichenfolgen in der Baumstruktur des Assistenten auffinden können. Verwenden Sie die Tastenkombination "Strg+F", um auf dieses Werkzeug direkt zuzugreifen.:  
Klicken Sie auf dieses Feld und geben Sie den gesuchten Text ein. Bei einer erfolgreichen Suche verschiebt AssetCenter die Auswahl automatisch auf den entsprechenden Eintrag. Verwenden Sie die Tastenkombinationen "F3" und "Umschalttaste+F3", um die vorigen bzw. nächsten Einträge zu suchen.

---

### Anmerkung:

Im Textmodus wird der gesamte Text durchsucht. Im Grafikmodus ist die Suche auf den Namen eines Knotens begrenzt.

---

## Anzeige der Baumstruktur des Assistenten

In der linken Hälfte des Grafikeditors erscheint die Baumstruktur des Assistenten:

Wenn Sie einen Knoten in der Baumstruktur auswählen, führt AssetCenter in der rechten Hälfte des Grafikeditors alle dem Knoten zugeordneten Eigenschaften auf.

## Liste der Eigenschaften eines bestimmten Knotens

In der rechten Bildschirmhälfte können Sie die Werte für die Eigenschaften eines Knotens eingeben:

Jede Eigenschaft verfügt über einen festen Wert oder ein Skript. Dabei kommen nachstehende Farbcodes zur Anwendung:

- Wenn die Eigenschaft ihren Standardwert verwendet, erscheinen Name und Wert in Grau. Sie können für diese Eigenschaft auch einen anderen Wert erzwingen. Name und Wert erscheinen daraufhin in Schwarz.
- Wenn die Eigenschaft einen benutzerdefinierten Wert oder ein Skript verwendet, werden Name und Wert in Schwarz angezeigt.
- Wenn die Eingabe eines Werts für die Eigenschaft obligatorisch ist, erscheinen Name und Wert in Rot.
- Die geänderten Werte werden in Blau angezeigt.

## Neue Knoten erstellen

In diesem Abschnitt werden die Vorgänge beschrieben, die mit einem Knoten durchgeführt werden können. Das Verschieben eines Knotens auf eine höhere bzw. tiefere Ebene oder das Löschen eines Knotens erfolgt über die Symbolleiste.

---

 **Anmerkung:**

Das Verschieben eines Knotens auf eine höhere bzw. tiefere Ebene oder das Löschen eines Knotens kann auch über das Kontextmenü durchgeführt werden. Klicken Sie dazu mit der rechten Maustaste auf den ausgewählten Knoten.

---

Bei der Erstellung eines Knotens müssen Sie zunächst den Hauptknoten wählen. Ein Beispiel: Wenn Sie einen neuen Knoten vom Typ "Page" wählen möchten, müssen Sie zunächst den Knoten "Root" wählen. Sobald Sie den übergeordneten Knoten gewählt haben, drücken Sie die rechte Maustaste, um das Kontextmenü anzuzeigen. Der Eintrag "Neu" im Kontextmenü fasst alle Knoten zusammen, die von Ihnen erstellt werden können:

AssetCenter fügt den Knoten daraufhin in der Baumstruktur des Assistenten ein.

## Eigenschaften eines Knotens bearbeiten

Nach dem Erstellen des Knotens müssen Sie den Eigenschaften dieses Knotens bestimmte Werte zuordnen. Diese Zuordnung erfolgt in der rechten Hälfte des Grafikeditors.

Der Wert einer Eigenschaft kann auf zwei verschiedene Weisen definiert werden:

- Eingabe eines festen Werts
  - Definition eines Skripts
- 

 **Anmerkung:**

Ein Skript hat grundsätzlich Vorrang vor einem festen Wert. Wenn Sie der Eigenschaft ein Skript und einen Wert zuordnen, ignoriert AssetCenter den festen Wert und interpretiert das Skript.

---



## Einer Eigenschaft einen festen Wert zuordnen

Klicken Sie neben der betreffenden Eigenschaft direkt in die Spalte "Wert". Je nach dem Typ der von der Eigenschaft akzeptierten Werte (Text, boolesch, Dezimalstellen usw.) werden Sie von AssetCenter zur Auswahl einer Liste von möglichen Werten bzw. zur Eingabe von Informationen in einem Textfeld aufgefordert.


## Einer Eigenschaft ein Skript zuordnen

Wählen Sie eine Eigenschaft, der sie ein Skript zuordnen möchten. Die Eingabe des Skripts erfolgt über das Feld **Skript**, das sich unter der Liste mit den Eigenschaften befindet.

 **Anmerkung:**

Wenn Sie mit der rechten Maustaste auf eine Eigenschaft klicken und in dem daraufhin erscheinenden Kontextmenü die Option **Standardwert wiederherstellen** wählen, bricht AssetCenter die Eingabe eines festen Werts oder eine Skripts ab, und führt eine erneute Initialisierung der Eigenschaft mit ihrem Standardwert durch. Dieser Vorgang kann nur mit den Eigenschaften ausgeführt werden, für die ein Wert oder ein Skript vom Benutzer definiert wurden, und die dementsprechend in Schwarz erscheinen.

## Assistenten ausführen und Fehler beheben

Klicken Sie in der Symbolleiste des Grafikeditors auf die Schaltfläche , um den Assistenten auszuführen. Wenn es bei der Ausführung zu Fehlern kommen sollte, werden sie im Fenster mit der Fehlerhistorie angezeigt. Der Zugriff auf dieses Fenster erfolgt mit dem im Assistenten integrierten Debugger. Verwenden Sie die Tastenkombination Umschalttaste+F9, um die Ausführung des Assistenten zu unterbrechen, sofern es sich um einen modalen Assistenten handelt, und aktivieren Sie den Debugger.

Auf diese Weise können Sie sämtliche Fehler im Assistenten mühelos erkennen und beheben.

 **Anmerkung:**

Bei kontextbezogenen Assistenten steht diese Schaltfläche nicht zur Verfügung.

## Beispiel für die Erstellung eines Assistenten

Zur Veranschaulichung der theoretischen Darstellung der Programmierung eines Assistenten wird auf den nun folgenden Seiten die Ausführung eines Umzugsassistenten erläutert. Der Assistent erleichtert den Umzug eines Benutzers und seiner Vermögensgegenstände von einem Standort zum anderen. Die Erstellung des Assistenten wird schrittweise beschrieben. Wir empfehlen Ihnen, diesen Vorgang nachzuvollziehen und die nächsten Seiten gegebenenfalls bei Problemen zu Rate zu ziehen.

- 1 Schritt 1 - Analysieren der Anforderungen
- 2 Schritt 2 - Definieren der Struktur des Assistenten
- 3 Schritt 3 - Schreiben der Struktur des Assistenten mit Hilfe der Skriptsprache

### Schritt 1 - Analysieren der Anforderungen

Mit Hilfe dieses Assistenten werden Vermögensgegenstände von einem Standort an einen anderen gebracht. Gehen Sie wie folgt vor:

- 1 Identifizieren Sie die Vermögensgegenstände, die an einen anderen Standort gebracht werden sollen.
- 2 Wählen Sie einen neuen Standort für die Gegenstände.

### Gegenstände identifizieren, die beim Umzug berücksichtigt sollen

Bei der Identifizierung der Gegenstände, die beim Umzug berücksichtigt werden sollen, haben Sie die Wahl zwischen drei Möglichkeiten:

- Identifizieren der Gegenstände anhand ihres Benutzers. Sie wählen zuerst den Benutzer und dann die Gegenstände, die beim Umzug berücksichtigt werden sollen.
- Direktes Identifizieren der betreffenden Vermögensgegenstände durch Auswahl in den Datensätzen der Tabelle der Vermögen und Lose.
- Identifizieren der Gegenstände anhand ihres Standorts. Sie wählen zuerst den Standort und dann die Gegenstände, die beim Umzug berücksichtigt werden sollen.

---

 **Anmerkung:**

Wir werden also eine Auswahlseite erstellen, auf der der Benutzer des Assistenten die Methode zur Auswahl der Vermögensgegenstände wählen muss, die beim Umzug berücksichtigt werden sollen.

---

## Wahl eines neuen Standorts

Zur Auswahl eines neuen Standorts für die Vermögensgegenstände brauchen Sie nur einen Datensatz in der Tabelle der Standorte auszuwählen.

## Schritt 2 - Definieren der Struktur des Assistenten

In diesem Schritt müssen Sie je nach den im Schritt 1 definierten Anforderungen die Struktur des Assistenten festlegen. Dazu gehören folgende Elemente:

- 1 Die Anzahl der Seiten
  - 2 Die Abfolge der verschiedenen Seiten
  - 3 Der Inhalt einer jeden Seite
- 

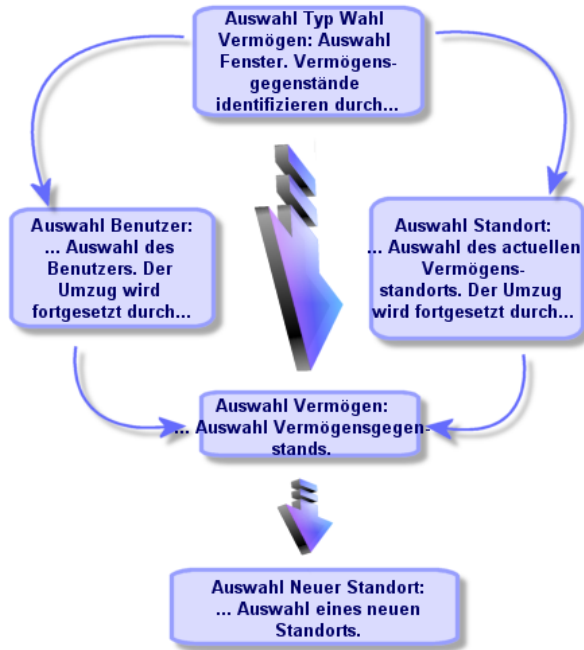
 **Anmerkung:**

Im Schritt 1 haben wir festgelegt, dass wir eine Auswahlseite erstellen müssen. Diese Seite ist die erste Seite des Assistenten, und wir nennen sie "AuswahlTypWahlVermögen".

---

Im Folgenden definieren wir die vollständige Struktur des Assistenten mit Hilfe des Schemas und der nachstehenden Tabelle:

Abbildung 14.3. Assistent - Beispiel



Unter Verwendung des oben abgebildeten Schemas definieren wir die Übergänge für jede Seite:

Seite	Seitenabfolge
AuswahlTypWahlVermögen	AuswahlVermögen, AuswahlBenutzer, AuswahlStandort
AuswahlVermögen	AuswahlNeuerStandort
AuswahlBenutzer	AuswahlVermögen
AuswahlStandort	AuswahlVermögen
AuswahlNeuerStandort	Keine

Jetzt legen wir den Inhalt der Seiten und damit die Steuerelemente fest, die dem Benutzer die Auswahl ermöglichen:

Seite	Über die Seite mögliche Aktionen	Zu verwendendes Steuerelement
AuswahlTypWahlVermögen	Der Benutzer hat die Wahl zwischen drei verschiedenen Möglichkeiten.	Steuerelement "CHOICE-BOX"
AuswahlVermögen	Der Benutzer kann die Vermögensgegenstände in der Liste der Datensätze in der Tabelle mit den Vermögensgegenständen wählen.	Steuerelement "DBLISTBOX"
AuswahlBenutzer	Sie können einen Benutzer wählen, dessen Vermögensgegenstände in der Liste der Datensätze der Tabelle mit den Abteilungen und Personen für einen anderen Standort gespeichert werden sollen.	Steuerelement "DBLISTBOX"
AuswahlStandort	Sie können einen aktuellen Standort in der Liste der Datensätze der Tabelle mit den Standorten wählen.	Steuerelement "DBLISTBOX"
AuswahlNeuerStandort	Der Benutzer kann einen neuen Standort für die Vermögensgegenstände in der Tabelle mit den Standorten wählen.	Steuerelement "DBLISTBOX"

## Schritt 3 - Schreiben der Struktur des Assistenten mit Hilfe der Skriptsprache

In diesem Schritt schreiben Sie das Skript für den Assistenten, und verwenden dabei die Beschreibungen der Struktur eines jeden einzelnen Knotens. Im Folgenden finden Sie den kommentierten Quellcode des für den Umzug verwendeten Assistenten. Dieser Code ist selbstverständlich nur eine von mehreren Möglichkeiten zum Schreiben eines Assistenten, der diese spezifische Aufgabe erfüllt.

```

=====
; (c) Peregrine Systems 1999
=====
NAME = "Move"
TITLE = "Umzug von Personen"
VERSION = "699"
;=====
=====

```

```

;Ask which user to move. By default, use selection in amEmplDept if co
ntext is on this table
;=====
=====
{ PAGE pgUser
  TITLE = "Wählen Sie die Personen, die am Umzug teilnehmen"
  { DBLISTBOX Users
    COLNAME = "Name|FirstName"
    COLWIDTH = "50|50"
    DBLCLICK = 1
    LABEL = "Beim Umzug zu berücksichtigende Personen"
    MULTISEL = 1
    TABLE = "amEmplDept"
    { VALUE =
      if [CurrentTable] = "amEmplDept" then
        RetVal = [CurrentSelection]
      else
        RetVal = ""
      end if
    }
    VISIBLE = 1
  }
  { TRANSITION trPersonToNewLoc
    TO = "pgNewLoc"
  }
}

;=====
=====
;Ask for new location
;=====
=====
{ PAGE pgNewLoc
  TITLE = "Wählen Sie den neuen Standort"
  { STRING UserName
    VALUE = AmDbGetString("SELECT FirstName + ' ' + Name FROM amEmplDe
pt WHERE lEmplDeptId IN (" & {pgUser.Users} & ")")
  }
  { LABEL LABEL1
    CAPTION = "Benutzer: " & {UserName}
  }
  { DBLISTBOX NewLocId
    COLNAME = "Name"
    COLWIDTH = "100"
    DBLCLICK = 1
    TABLE = "amLocation"
    VALUE = "-1"
  }
  { TRANSITION trNewLocToAssets
    TO = "pgRecap"
  }
}

;=====
=====

```

```

=====
;Recapitulation
;=====
=====
{ PAGE pgRecap
  TITLE = "Zusammenfassung"
  { LISTBOX Users
    COLTITLE = "Name"
    COLWIDTH = "100"
    LABEL = "Beim Umzug zu berücksichtigende Personen"
    MANDATORY = 0
    MULTISEL = 1
    READONLY = 1
    VALUE = ""
    VALUES = AmDbGetList("SELECT FullName FROM amEmplDept WHERE FullNa
me LIKE LikeParam(amEmplDept_2:FullName)+'%' AND amEmplDept_2:lEmplDep
tId IN(" & {pgUser.Users} & ")","|","", "=",)
  }
}
;=====
;Finish
;=====
=====
{ FINISH FINISH
  { DO =
    On Error Goto ErrHandler
    Dim lErr as long

    dim hRecord as Long

    dim iEmplCount as Integer
    iEmplCount = {pgRecap.Users.VALUES.Count()}
    dim iMax as Long
    iMax = iEmplCount

    dim lLocaId as long
    lLocaId = {pgNewLoc.NewLocId}

    lErr = amStartTransaction()

    dim i as Integer
    For i = 1 To iEmplCount
      lErr = AmProgress((100 * i ) / iMax)
      lErr = AmLog("Umzug der Person" + {pgRecap.Users.VALUES(i,1)})
      hRecord = AmGetRecordFromMainId("amEmplDept", {pgRecap.Users.VAL
UES(i,0)} )
      If hRecord <> 0 then
        lErr = AmSetFieldLongValue( hRecord, "lLocaId", lLocaId)
        lErr = AmUpdateRecord(hRecord)
        lErr = AmReleaseHandle(hRecord)
      End If
    Next i
  }
}

```

```

lErr = amCommit()

RetVal = 0
Exit Function

ErrorHandler:
    On Error Goto 0
    AmLog(AmLastError() & " - " & AmLastErrorMsg())
    AmLog("Die Transaktion wurde abgebrochen")
    RetVal = 1
    Exit function
}
SUMMARY = 1
}
}

```

## Beispiel für die Erstellung eines Abfrageassistenten (QBE)

Abfrageassistenten werden speziell zum Filtern der Daten bestimmter Bildschirme eingesetzt.

Im nachstehend beschriebenen Beispiel wird ein Teil der Skripte eines vorgegebenen Abfrageassistenten erläutert.

### Schritt 1 - Analysieren der Anforderungen

Der Assistent soll die Daten auf dem Bildschirm der Interventionen filtern.

Daher muss Folgendes festgelegt werden:

- 1 Die vom Abfrageassistenten verwendeten Felder
- 2 Der Feldtyp

### Schritt 2 - Definieren der Struktur des Assistenten

In diesem Schritt müssen Sie je nach den im Schritt 1 definierten Anforderungen die Struktur des Assistenten festlegen. Dazu gehören folgende Elemente:

- 1 Die Anzahl der Seiten  
Damit der Filtervorgang nicht zu umfangreich wird, ist es ratsam, nur eine Seite zu erstellen.
- 2 Der Inhalt einer jeden Seite



### 3 Die Abfolge der verschiedenen Seiten

Seite	Über die Seite mögliche Aktionen	Zu verwendende Steuerelemente
Filter	Unterschiedliche Felder zum Filtern der Daten des anzuzeigenden Bildschirms	TEXTBOX COMBOBOX LABEL LINKEDIT DATETIMEEDIT CHECKBOX
Advanced	Fakultative Felder, die einen tief greifenderen Filtervorgang ermöglichen	COMBOBOX LINKEDIT DATETIMEEDIT LABEL

## Schritt 3 - Schreiben der Struktur des Assistenten mit Hilfe der Skriptsprache

In diesem Abschnitt werden bestimmte Teile des Assistentenskripts für den Bildschirm der Interventionen im Detail erläutert.

Die beschriebenen Skriptteile dienen zur Erstellung von QBE-Assistenten.

### WICHTIG:

Bei der Definition der Objektnamen des Assistenten sollten die in der Assistentensyntax vorkommenden Namen (beispielsweise 'name' oder 'title') vermieden werden.

- ```

NAME = "searchamWorkOrder"
TITLE = "Interventionen filtern"
VERSION = "3152"
{ PAGE Filter
  TITLE = "Schnellsuche"

```

Definition des Assistententitels und der -seite

- ```

{ TEXTBOX WONO
  LABEL = AmGetFieldLabelFromName ("amWorkOrder", "WONO")
  LABELLEFT = 1
  VALUE = ""
  XOFFSET = 1500
}

```

## Erstellen des Objekts **Interventionsnummer** aus der Tabelle der Interventionen und Anzeigen der entsprechenden Bezeichnung im Assistenten

- ```

{ COMBOBOX seStatus
  LABEL = AmGetFieldLabelFromName("amWorkOrder", "seStatus")
  LABELLEFT = 1
  VALUE = ""
  { VALUES =
    Dim strFormat As String
    strFormat = AmGetFieldFormat(AmGetFieldFromName(AmGetTablePro
mName("amWorkOrder"), "seStatus"))
   RetVal = SysEnumToComboBox(strFormat)
  }
  XOFFSET = 1500
}
{ COMBOBOX Priority
  LABEL = AmGetFieldLabelFromName("amWorkOrder", "Priority")
  LABELLEFT = 1
  VALUE = ""
  { VALUES =
    Dim strValues As String
    strValues = AmDbGetList("SELECT Value FROM amItemListVal WHER
E ItemizedList.Identifier = 'amWOPriority'", "", ",", "")
   RetVal = EnumToComboBox(strValues)
  }
  XOFFSET = 1500
}

```

### Definition der Aufzählungswerte

#### 1 Für das Objekt **Status**:

- Übernahme der Bezeichnung aus dem Feld **Status** der Tabelle der Interventionen und Anzeigen im Assistenten
- Definition des Felds als Feld vom Typ "Text"
- Erstellen eines Objekts vom Typ "Aufzählung"
- Übernahme des Inhalts aus der diesem Feld zugeordneten Systemaufzählung und Formatieren durch die in den Bibliotheken zur Assistentenkonfiguration definierte Funktion **SysEnumToComboBox**

#### 2 Für das Feld **Priorität**:

- Definition des Felds als Feld vom Typ "Text"
- Übernahme des Inhalts aus der diesem Feld zugeordneten Aufzählung
- Erstellen eines Objekts vom Typ "Systemaufzählung" und Formatieren des entsprechenden Inhalts im Format "Aufzählung" durch die in den Bibliotheken zur Assistentenkonfiguration definierte Funktion **EnumToComboBox**

- ```

{ LINKEDIT Model
  FILTER = "Nature.seBasis = 3"
  LABEL = AmGetFieldLabelFromName("amWorkOrder", "Model")
  LABELLEFT = 1
  TABLE = "amModel"
  VALUE = 0
  XOFFSET = 1500
  YOFFSET = 0
  YOFFSET2 = 0
}

```

Erstellen eines Objekts zur Eingabe einer Verknüpfung:

- 1 Erstellen eines Filters für alle Modelle, deren Art zur Erstellung einer Intervention führt
- 2 Übernahme der Bezeichnung aus dem Feld **Modell** der Tabelle der Interventionen und Anzeigen im Assistenten
- 3 Definition der Tabelle, aus der die Verknüpfung gewählt wird

- ```

{ LABEL Empty2
}
{ COMBOBOX OPdtResolLimit
  LABEL = AmGetFieldLabelFromName("amWorkOrder", "dtResolLimit")
  LABELLEFT = 1
  VALUE = ""
  VALUES = "Vor le=<\=, Nach le=>\=, Le=\=, Sauf le=<>"
  XOFFSET = 1500
}
{ DATETIMEEDIT dtResolLimit
  FORMAT = "DateTime"
  LABEL = ""
  LABELLEFT = 1
  VALUE = ""
  XOFFSET = 1500
}
}

```

Erstellen eines Objekts vom Typ "Werteauzählung"

- 1 Übernahme der Bezeichnung aus dem Feld **Frist** der Tabelle der Interventionen und Anzeigen im Assistenten
- 2 Definition einer Liste möglicher Operatoren

Erstellen eines Objekts vom Typ "Datum"

- ```

{ CHECKBOX bAdvanced
  CAPTION = "Erweiterte Suche"
  VALUE = 0
}
{ TRANSITION Transition
  CONDITION = 1
  { TO =
    If {Filter.bAdvanced} = 0 Then
      RetVal = "Finish"
    Else
      RetVal = "Advanced"
  }
}

```

```

    End If
  }
}

```

Erstellen eines Kontrollkästchens:

- 1 Definition der Kontrollkästchen-Bezeichnung
- 2 Definition des Kontrollkästchen-Verhaltens: standardmäßig nicht markiert
- 3 Definition des Übergangs: Wenn das Kontrollkästchen markiert ist, wird die Schaltfläche "Weiter" aktiviert

```

• { FINISH FINISH
  { DO =
    Dim strFilter As String

    Dim strOperator As String
    strOperator = ""

    'WONo condition
    If {Filter.WONo} <> "" Then
      strFilter = strFilter & " AND WONo LIKE " & AmSqlTextConst({F
ilter.WONo})
    End If

    'Model condition
    If {Filter.Model} <> "0" Then
      strFilter = strFilter & " AND lModelId = " & {Filter.Model}
    End If

    If strFilter<> "" then
      strFilter = RightPartFromLeft(strFilter, "AND", 0)
      RetVal = AmOpenScreen("amWorkOrder", "", strFilter, 1, "")
    Else
      'ERR CANCEL
      RetVal = 2
    End if
  }
}

```

Erstellen der Filterbedingungen (FINISH):

- 1 Das Objekt FINISH wird erstellt.  
Es dient zum Schreiben des WHERE-Teils der SQL-Abfrage.
- 2 Für jedes zuvor erstellte Objekt wird eine Filterbedingung definiert:
  - Objekt vom Typ "Text", **WONo condition**:
    - 1 Wenn das Feld leer ist, wird nach dem eingefügten Wert gefiltert.
  - Objekt vom Typ "Verknüpfung", **Model condition**:
    - 1 Wenn das Feld nicht leer ist, wird nach der Verknüpfungs-ID gefiltert, und die Modellverknüpfungen werden angezeigt.

- 3 Für den Wert des anzuzeigenden Bildschirms wird eine Filterbedingung definiert.  
Diese Bedingung nimmt die definierte WHERE-Klausel wieder auf und berücksichtigt lediglich den Teil, der nach dem ersten AND beginnt (zur Vereinfachung der Abfrage).
- 4 Der Bildschirm wird gefiltert und von der Funktion **AmOpenScreen** aufgerufen.

## Häufig gestellte Fragen

In diesem Kapitel finden Sie die Antworten auf die Fragen, die bei der Erstellung eines Assistenten am häufigsten auftauchen.

### Frage

Der folgende Code ist ungültig:

```
{lBxMyListBox.Values.Count}
```

### Antwort

Für die Methode müssen Sie eine öffnende und eine schließende Klammer eingeben. Der richtige Code wird wie folgt geschrieben:

```
{lBxMyListBox.Values.Count ()}
```

### Frage

Der folgende Code ist ungültig:

```
{lBxMyListBox.Line (lRow)}
```

### Antwort

Die Methode "LINE" ist der Eigenschaft "VALUES" des Steuerelements "LISTBOX" zugeordnet. Der richtige Code wird wie folgt geschrieben:

```
{lbxMyListBox.Values.Line(lRow)}
```

## Frage

Der folgende Code ist ungültig:

```
{lbxMyListBox.Values.Line({lbxTmp})}
```

## Antwort

Sie können keine Eigenschaften verwenden, auf die in einer Methode verwiesen wird. Der richtige Code wird wie folgt geschrieben:

```
Dim lRow As Long
lRow = {lbxTmp}
{lbxMyListBox.Values.Line(lRow)}
```

## Frage

Der nachstehende Code, mit dem einer Eigenschaft ein fester Wert zugeordnet wird, scheint nicht zu funktionieren:

```
{Eigenschaft} = 123
```

## Antwort

Dieser Code ist für die von den Assistenten verwendeten Datenmodelle nicht besonders gut geeignet. Wir empfehlen Ihnen, bei der Zuordnung eines Werts zu einer Eigenschaft die dedizierte Funktion "Am SetProperty()" zu verwenden, wie aus dem nachstehenden Beispiel zu entnehmen ist!

```
Dim irc as Integer
irc= AmSetProperty("Eigenschaft", 123)
```

 **Anmerkung:**

Denken Sie daran, den Rückgabecode (in diesem Beispiel "irc") zu notieren, auch wenn Sie ihn nicht verwenden.

---

## Frage

Beim Ausführen eines Assistenten, über den ein Vermögensgegenstand in der Datenbank erstellt wird, erscheint die nachstehende Fehlermeldung:

```
12001 - Sie verfügen über keinerlei Berechtigung für den Schreibzugriff
```

Diese Meldung erscheint auch dann, wenn der Assistent von einem Benutzer ausgeführt wird, der die Verbindung zur Datenbank als Administrator hergestellt hat.

## Antwort

Diese Meldung wird angezeigt, wenn der Schreibzugriff außerhalb des Knotens vom Typ "FINISH.DO" eines Assistenten erfolgt. Das Funktionsschema eines Assistenten lautet wie folgt:

- 1 Sammeln von Informationen mit Hilfe aufeinanderfolgender Seiten. Der Schreibzugriff ist auch für den AssetCenter-Administrator unzulässig.
- 2 Ausführen des Skripts, das im Knoten FINISH.DO enthalten ist. Der Schreibzugriff wird unter Berücksichtigung der entsprechenden Berechtigungen autorisiert.

## Frage

Die Fehlermeldungen, die beim Ausführen eines Assistenten erscheinen, sind zum Teil unvollständig.

## Antwort

Zeigen Sie mit Hilfe der Tastenkombination "Umschalttaste+F9" das Fenster mit dem Debugger an. Die Meldungen, die im Historienfenster erscheinen, sind in der Regel ausführlicher und damit leichter verständlich.

## Frage

Sobald das Steuerelement "DBLISTBOX" auf einer Seite des Assistenten verwendet wird, sinkt die Leistung der Software. Ist das normal?

## Antwort

Dieses Problem tritt auf, wenn Sie das Steuerelement "DBLISTBOX" zusammen mit einem Filter verwenden. Bei dieser Konfiguration wird jedes Mal, wenn Sie die Auswahl ändern, eine Abfrage an die Datenbank gesendet, um sicherzustellen, daß der Filter bei der Auswahl berücksichtigt wird. Diese zusätzliche Abfrage wird nicht ausgeführt, wenn die Auswahl vom Benutzer festgelegt wurde.

## Frage

Wie kann die Bearbeitung bestimmter Spalten des Steuerelements LISTBOX verhindert werden?

## Antwort

Verwenden Sie die Eigenschaft "EDITABLE" dieses Steuerelements. Der Wert, der dieser Eigenschaft zugeordnet ist, ist eine Folge von "0" und "1", die durch einen senkrechten Strich "|" voneinander getrennt sind. "0" definiert die Spalte als nicht bearbeitbar, und "1" definiert die Spalte als bearbeitbar. Wenn Sie einen Wert auslassen, kann die entsprechende Spalte nicht bearbeitet werden. Dementsprechend sind im nachfolgenden Beispiel die Spalten 2 und 4 nicht bearbeitbar:

```
EDITABLE = " |1| |1"
```

## Frage

Was ist zu tun, damit ein Assistent ein Detailfenster öffnet?

## Antwort

Sie müssen die DDE-Aufrufe mit Hilfe einer Funktion innerhalb des Assistenten verwenden. Der Assistent darf nicht modal sein. Im Folgenden finden Sie ein Beispiel zum Öffnen der Tabelle der Vermögen und Lose innerhalb eines Assistenten:

```
Dim irc as Long
irc = AmActionDDE("am", "AssetCenter", "OpenTable(amAsset)")
```



## Frage

Was ist der Unterschied zwischen den Eigenschaften "COLNAME" und "COLTITLE" des Steuerelements "LISTBOX"?

## Antwort

Die Bezeichnung der Spalten eines Steuerelements vom Typ "LISTBOX" kann sowohl automatisch als auch manuell definiert werden:

- Die Eigenschaft "COLNAME", die der Eigenschaft "TABLE" zugeordnet ist, ermöglicht das automatische Definieren der Spaltenüberschrift für ein Steuerelement vom Typ "LISTBOX" und zwar anhand der Feldbeschreibungen der Datenbank.
- Sofern eingegeben, erzwingt die Eigenschaft "COLTITLE" die Spaltenüberschrift. Wenn diese Eigenschaft nicht definiert ist, werden die Spaltenüberschriften über die Eigenschaft "COLNAME" definiert.

Dementsprechend gilt folgendes Beispiel:

```
...  
TABLE = "amEmplDept"  
COLNAME = "Name| |FirstName"  
COLTITLE = "|A|B"  
...
```

In diesem Beispiel werden die Überschriften in den Spalten des Steuerelements "LISTBOX" angezeigt: Name, A, B.

Die Eigenschaft "COLNAME" definiert zusätzlich den Typ des Steuerelements, das zur Bearbeitung herangezogen wird, wenn die Werte in den Spalten des Steuerelements nicht bearbeitet werden können.




# 15 | Mitteilungen

## KAPITEL

In diesem Kapitel wird das Verteilen und Verwalten von Mitteilungen mit AssetCenter beschrieben.

Der Zugriff auf die Liste mit den Mitteilungen erfolgt über das Menü **Extras/ Mitteilungen**.

Die Anzeige der Mitteilungen wird über das Menü **Fenster/ Mitteilungen anzeigen** oder die Schaltfläche  in der Symbolleiste aktiviert bzw. deaktiviert.

## Definition einer Mitteilung

Eine Mitteilung ist eine aktuelle Information, die zu einem bestimmten Zeitpunkt an eine Reihe von Personen gesendet werden soll.

Diese Personen gehören zu Personengruppen.

Normalerweise werden Mitteilungen nur für Kurzinformationen verwendet.

Ein Beispiel: "Der Server XXX wird am 10.02.2002 zwischen 11:00 und 12:00 Uhr heruntergefahren".

# Allgemeine Funktionsweise von Mitteilungen

## Mitteilungen erstellen

Jeder Benutzer mit den entsprechenden Rechten kann über das Menü **Extras/Mitteilungen** Mitteilungen in der Tabelle der Mitteilungen erstellen.

Im Detailfenster einer Mitteilung erscheinen folgende Informationen:

- Mitteilungstext
- Empfängerliste
- Gültigkeitsperiode

## Mitteilung lesen

Alle Benutzer können die Mitteilungen in der Anzeigeleiste lesen, sofern ihr Name in der jeweiligen Empfängerliste geführt wird.

## Bedeutung der Mitteilungen definieren

Um die Bedeutung einer Mitteilung festzulegen, geben Sie einfach im Detailfenster der Mitteilung im Feld **Bedeutung (Nom SQL : seSeverity)** einen entsprechenden Wert ein.

Jeder Bedeutung ist eine bestimmte Farbe zugeordnet. Die Zuordnung erfolgt im Feld **Farbe (SQL-Name: IColor)** der Mitteilungsanzeige.

## Mitteilungen verteilen

Die Registerkarte **Mitteilungen** im Detailfenster einer Mitteilung enthält den Text, der veröffentlicht werden soll.

Eine Mitteilung kann max. 255 Zeichen umfassen.




## Empfängerliste für Mitteilungen

Die Registerkarte **Empfänger** im Detailfenster einer Mitteilung zeigt die Personengruppen, die die Mitteilung lesen können.

### Kontrollkästchen Alle Personengruppen (SQL-Name: bAllGroups)

Wenn Sie dieses Kontrollkästchen aktivieren, kann die Mitteilung von allen Personengruppen gelesen werden.

Ansonsten kann sie nur von den Mitgliedern der Personengruppen gelesen werden, die in der Liste der Registerkarte **Empfänger** markiert wurden.

Verwenden Sie die Schaltflächen ,  und  zum Hinzufügen, Löschen, Anzeigen und Ändern von Personengruppen in der Empfängerliste.

### Kontrollkästchen Mit Untergruppen (SQL-Name: bChildGrps)

Eine Mitteilung kann an alle Personengruppen oder an eine Auswahl von Gruppen gesendet werden. Wenn Sie dieses (standardmäßig markierte) Kontrollkästchen aktivieren, kann die Mitteilung von allen Untergruppen der ausgewählten Gruppe gelesen werden, da die Tabelle der Personengruppen hierarchisch aufgebaut ist.

## Mitteilungen anzeigen

Zum Anzeigen von Mitteilungen gehen Sie wie folgt vor:


- 1 Aktivieren Sie die Mitteilungsanzeige.
- 2 Verwenden Sie die Schaltflächen in der Mitteilungsanzeige, um die Mitteilungen einzusehen.

Sie können Folgendes definieren:





- Farbe zur Anzeige von Mitteilungen unter Berücksichtigung ihrer Bedeutung
- Automatischer Aktualisierungsmodus für die Mitteilungsanzeige.

## Mitteilungsanzeige aktivieren

Die Mitteilungsanzeige kann von allen AssetCenter-Benutzern aktiviert werden, die dort die ihnen gesendete Mitteilungen lesen können. Das Aktivieren bzw. Deaktivieren der Mitteilungsleiste erfolgt auf zwei verschiedene Weisen:

- Über das Menü **Fenster/ Mitteilungen anzeigen**
- Über das Symbol  in der Symbolleiste

**Tabelle 15.1. Schaltflächen in der Mitteilungsanzeige**

	Klicken Sie auf diese Schaltfläche, um die aktuellen Mitteilungen zu lesen, ohne die in den Optionen (Menü <b>Bearbeiten/ Optionen</b> ) festgelegte Frist abzuwarten.
	Klicken Sie auf diese Schaltfläche, um die vorherige Mitteilung anzuzeigen.
	Klicken Sie auf diese Schaltfläche, um die nächste Mitteilung anzuzeigen.
	Klicken Sie auf diese Schaltfläche, um das Anzeigen einer Mitteilung in der Leiste zu unterbrechen oder wiederaufzunehmen.

# Index

## Stichwortverzeichnis

- \$C (Formulare), 74
- \$D (Formulare), 74
- \$N (Formulare), 74
- \*= (AQL), 52
- < (AQL), 52
- <> (AQL), 51
- =! (AQL), 51
- = (AQL), 51
- =\* (AQL), 52
- =< (AQL), 52
- = ALL (AQL), 53
- = ANY (AQL), 53
- = SOME (AQL), 53
- > (AQL), 52
- >= (AQL), 52
- Übergang (SZ-WZ) (Feld), 235
- Übergang von Assistenten, 263
- Übergeordnete Objekte, 261
- A**
- Abfrageeditor (Siehe Abfragen)
- Abfragen, 23-70
  - Administrator, 42
  - Anzahl der Tabellen, 54
  - AQL und SQL im Vergleich, 25
  - Assistent, 336
  - Aufzählungen, 46
  - Ausdrücke, 44
  - Beispiele, 66 , 65
  - alias, 69
  - Ausdrücke, 68
  - Fehlende Verknüpfung, 68
  - FullName, 67
  - FullName und sLvl, 67
  - Hierarchische Tabellen, 67
  - Leere Felder, 68
  - Merkmale, 68
  - Mit der Haupttabelle verbundene Tabellenfelder, 66
  - Verknüpfungen, 66 , 66
  - Zahlen, Daten und Texte, 67
  - Zwei Bedingungen, 67
- Berechnete Felder, 37
- Beschreibungsfolgen, 36
- Beziehungen, 27
- CurrentUser, 32
- Datensätze mit dem Primärschlüssel 0, 28
- DELETE, 60
- DUPLICATE, 60
- Editor, 40
  - Funktionsprinzip, 40
  - Zugriff, 41
- Einführung, 23
- Elemente, 48
  - Ausdrücke, 50
  - Auswahlliste, 53
  - Felder und Verknüpfungen, 48
  - Konstanten, 49

- Operatoren, 51
- Empfehlungen, 26
- Erforderliche Kenntnisse, 24
- Erstellung, 42
- Externe Beziehungen
  - Operatoren, 31
  - ZAHL, 28
- Externe Schlüssel, 35
- Felder, 44
- Filter, 43
  - Direkteingabe, 43
- Freie Verwendung, 42
- FROM, 54
- FullName (Feld), 34
- Funktionen (Siehe AQL-Funktionen)
- Funktionsweise, 25
- GROUP BY, 56
- HAVING, 57
- Hierarchische Tabellen, 34
- Index, 37
- Index - Hinweise, 39
- INSERT, 58
- Konstanten, 45
- Merkmale, 36
- NULL, 31
- Numerische Felder, 31
- ORDER BY, 58
- Self, 32
- sLvl (Feld), 35
- Sortieren, 37
  - Aufzählungen, 38
  - Reihenfolge, 38
  - Sonstige Hinweise, 39
  - Verwendung von Indizes erzwingen, 38
- Syntax, 46
  - Einfache Abfragen, 47
  - Konventionen, 46
  - Nebenabfragen, 47
  - UNION, 48
- Systemaufzählungen, 33 , 33
- UPDATE, 59
- Vereinfachte Notationen, 35
- Vorschau, 44
- Vorteile von AQL, 24
- WHERE, 55
- Abfragen (Menü), 41
- Abfragesyntax (Siehe Abfragen)
- Abs (AQL), 64
- Abschließende Workflow-Ereignisse, 175 , 175
- AddDays (AQL), 63
- AddHours (AQL), 63
- AddMinutes (AQL), 63
- AddSeconds (AQL), 63
- Adresse - Aktionen vom Typ "Nachricht", 105
- Advanced Query Language (Siehe Abfragen)
- Aktion (Aktionstyp), 101
- Aktionen, 95-117
  - (Siehe auch Assistenten)
  - Ausführung, 116
  - Beispiele, 107
    - DDE, 108
    - Nachrichtensystem, 112
    - Programm, 107
    - Skript, 112
  - Definition, 95
  - Druck - Kontext, 100
  - Erstellung, 97
  - Funktionsbereiche, 96
  - Nachrichtensystem, 120
  - Registerkarte "DDE", 103
  - Registerkarte "Nachrichtensystem", 104
  - Schaltflächen - Zuordnen, 117
  - Sybase SQL Anywhere, 115
  - Tests, 115
  - Typen, 97
  - Variablen, 114
- Aktionstypen, 97
- alias (AQL), 69
- am.ini, 245
- amdb.ini, 28
- amexpl.exe, 187
- AND (AQL), 51
- Anlegen von Historien, 17-22
  - 1-Verknüpfungen, 19
  - Allgemeine Konzepte, 17
  - Anlegen von Historienzeilen, 18



- Datensatz, 19
- Felder, 19
- Merkmale, 20
  - Aktivierung, 22
  - Änderung, 21
  - Anlegen, Ändern und Löschen einer Historienzeile, 22
  - Hinzufügen, 20
  - Löschen, 21
- n-Verknüpfung, 20
- n-Verknüpfungen, 20 , 19
- AQL (Siehe Abfragen)
- AQL-Funktionen, 60
  - Aggregate, 61
    - Avg, 61
    - Count, 61
    - Countdistinct, 61
    - Max, 61
    - Min, 61
    - Sum, 61
- Daten, 62
  - AddDays, 63
  - AddHours, 63
  - AddMinutes, 63
  - AddSeconds, 63
  - Day, 62
  - DayOfYear, 62
  - DaysDiff, 63
  - DbToLocalDate, 63
  - Getdate, 63
  - Hour, 63
  - HoursDiff, 63
  - LocalToDate, 63
  - Minute, 63
  - MinutesDiff, 63
  - Month, 62
  - Second, 63
  - SecondsDiff, 63
  - WeekDay, 63
  - Year, 62
- Numerisch, 64
  - Abs, 64
  - Ceil, 64
  - Floor, 64
  - Mod, 64
  - Round, 64
  - Trunc, 64
- Strings, 62
  - Ascii, 62
  - Char, 62
  - Left, 62
  - Length, 62
  - Lower, 62
  - Ltrim, 62
  - Right, 62
  - Rtrim, 62
  - Substring, 62
  - Upper, 62
- Test, 65
  - IsNull, 65
- Arbeitszeiten, 225
- ASCII (AQL), 62
- AssetCenter Server - Zeitzonen, 242
- Assistent (Aktionstyp), 100
- Assistenten, 259-345
  - { - Syntax, 267
  - Abfrageassistent, 336
  - Allgemeine Informationen, 266
  - Ausführung, 329
  - Ausführung - Beschränkungen, 329
  - BASIC-Funktionen, 273
  - Editor
    - Arbeiten mit, 325
    - Benutzeroberfläche, 326
    - Symbolleiste, 326
    - Text - Suche, 327
    - Zugriff, 325
  - Eigenschaften, 293
  - Erstellung - Beispiele, 330
  - Fehlerbehebung, 329
  - Finish (Knoten), 287
  - Fragen, 341
  - Knoten
    - Beschreibendes Modell, 267
    - CurrentSelection, 271
    - CurrentTable, 271

- Eigenschaften, 328 , 268 , 267
  - Eigenschaften - Beschränkungen, 267
  - Erstellung, 328
  - Konstanten, 268
  - Methoden, 270
  - Namen - Beschränkungen, 261
  - Skripte, 269
  - Skripte oder feste Werte - Vorrang, 328
  - Struktur und Syntax, 266
  - Tabellen, 270
  - Zahlen, 268
  - Long (Knoten), 291
  - Page (Knoten)
    - Definition, 282
    - Eigenschaften, 283
    - Syntax, 282
    - Untergeordnete Knoten, 284
  - Parameter, 288
  - Root (Knoten)
    - Definition, 273
    - Eigenschaften, 274
    - Syntax, 274
    - Untergeordnete Knoten, 281
  - Schreibweise - Konventionen, 259
  - Seiten - Modelle, 265
  - Skripte - Beschränkungen, 273
  - Start (Knoten), 289
  - Start - Seite, 290
  - Steuerelement (Knoten), 292 , 292
  - Steuerelemente, 293
    - CALENDAR, 323
    - CHART, 320
    - CHECKBOX, 297
    - COMBOBOX, 298
    - COMBOEDIT, 324
    - COMMANDBUTTON, 306
    - DATETIMEEDIT, 325
    - DBEDIT, 314
    - DBLISTBOX, 307
    - DBPATH, 317
    - DBQUERYBOX, 311
    - DBTABLE, 316
    - FILEEDIT, 322
    - LABEL, 305
    - LINKEDIT, 317
    - LISTBOX, 300
    - NUMBOX, 324
    - OPTIONBUTTONS, 299
    - PROGRESSBAR, 305
    - TEXTBOX, 319
    - TICKEDIT, 323
    - TIMESPANEDIT, 323
  - String (Knoten), 291
  - Strukturmodelle, 263
  - Timer (Knoten), 290
  - Transition (Knoten)
    - Besonderheiten, 287
    - Definition, 285
    - Eigenschaften, 285
    - Syntax, 285
  - Verkettung, 272
  - Zellen - Werte, 307
  - Aufgabe speichern (Option), 176
  - Aufzählungen, 13-350 , 13-16
    - Änderungen - Berücksichtigung, 15
    - Anpassbare Aufzählungen, 13
      - Geschlossene Aufzählungen, 15
      - Offene Aufzählungen, 14
      - Werte, 14
    - Skripte in Assistenten, 337
    - Systemaufzählungen, 15 , 15
  - Ausdrücke - Abfragen, 50
  - Ausführbares Programm (Aktionstyp), 97
  - Ausführen (Schaltfläche), 116
  - Ausnahmen (Registerkarte), 227
  - Auswahllisten - Abfragen, 53
  - Automatische Aktion (Workflow-Aktivitäten), 157
  - Avg (AQL), 61
- B**
- BASIC (Siehe Skripte)
    - (Siehe auch Assistenten)
    - Informationen, 200
  - BASIC-Funktionen
    - (Siehe auch Assistenten)
    - (Siehe auch Skripte)

- Definition, 202
- Bearbeiter einer Workflow-Aktivität
  - Bezeichnung, 154
  - Definition, 126
- Bedeutung der Mitteilungen definieren, 348
- Beginn (Workflow-Aktivitäten), 159
- Benutzeraktion (Workflow-Aktivitäten), 156
- Berechnen (Schaltfläche), 115
- Berechnete Felder, 249-258
  - (Siehe auch Skripte)
- Abfragen, 37
- Arbeiten mit, 257
- Berechnungsformeln, 255
- Datenbank - Speichern von Werten, 250
- Definition, 249
- Erstellung, 250
- Filter, 257
- Listen, 257
- Referenz, 257
- SQL-Name - Beschränkungen, 255
- Typen, 251
  - Anwendungsbereich, 252
  - Empfehlungen, 254
  - Verwendete Sprache, 256
- Verwendungskontext, 255
- Vorzüge, 250
- Zugriffsberechtigungen, 256
- Berichte
  - (Siehe auch Formulare)
  - Änderung, 88
  - CurrentUserId, 90
  - Detail, 87
  - Detailberichte
    - Erstellung, 89
    - Parametrierung, 90
    - Verwendung - Beispiel, 89
  - Druck, 92
  - Funktionsweise, 83
  - Identifikation, 93
  - Installation
    - Crystal Reports-Version (Voll- oder Runtime-Version), 85
    - Neue Datenbank, 86

- Vorhandene Datenbank, 86
- Vorkonfigurierte Berichte, 85
- Kontextunabhängige Berichte, 92
- Listen - Beschränkungen, 78
- Schaltflächen - Zuordnen, 93
- SQL-Namen (Beschränkungen), 90
- Statistiken, 88
- UNIX - Beschränkungen, 83
- Vorkonfigurierte Berichte, 85
- Vorschau, 88
- Beschreibende Modelle - Assistenten, 267
- Beschreibungsfolgen - Abfragen, 36
- Beziehungen, 27
- Bild (Formulare), 78
- Bilder in Formularen, 75

## C

- CALENDAR (Assistentenknoten), 323
- Ceil (AQL), 64
- Char (AQL), 62
- CHART (Assistentenknoten), 320
- CHECKBOX (Assistentenknoten), 297
- COMBOBOX (Assistentenknoten), 298
- COMBOEDIT (Assistentenknoten), 324
- COMMANDBUTTON (Assistentenknoten), 306
- Count (AQL), 61
- Countdistinct (AQL), 61
- Crystal Reports (Siehe Berichte)
- Crystal Reports - Einstellung der Berichtparameter, 90
- Crystal Reports-Statistiken, 88
- CurrentSelection (Assistentenknoten), 271
- CurrentTable (Assistentenknoten), 271
- CurrentUser (AQL), 32
- CurrentUser (Skripte), 215
- CurrentUserId (Berichte), 90

## D

- database.txt, 215, 27, 16
- Daten
  - Abfragen, 67
  - Skripte, 213
- Datenbank

(Siehe auch Anlegen von Historien)  
 Vorkonfigurierte Formulare, 71  
 Daten - Export (Siehe Export)  
 DATETIMEEDIT (Assistentenknoten), 325  
 Dauer - Skripte, 213  
 Day (AQL), 62  
 DayOfYear (AQL), 62  
 DaysDiff (AQL), 63  
 DBEDIT (Assistentenknoten), 314  
 DBLISTBOX (Assistentenknoten), 307  
 DBPATH (Assistentenknoten), 317  
 DBQUERYBOX (Assistentenknoten), 311  
 DBTABLE (Assistentenknoten), 316  
 DbToLocalDate (AQL), 63  
 DDE (Aktionstyp), 98  
 DDE (Registerkarte), 103  
 Definitionen  
 Abschließende Workflow-Ereignisse, 175  
 Aktionen, 95  
 BASIC-Funktionen, 202  
 Bearbeiter einer Workflow-Aktivität, 126  
 Berechnete Felder, 249  
 Exportabfragen, 186  
 Exportskripte, 185  
 Finish (Assistentenknoten), 287  
 Formulare, 71  
 Knoten, 261  
 Kontextunabhängige Berichte, 92  
 Long (Assistentenknoten), 291  
 Mitteilungen, 347  
 Objekte, 261  
 Page (Assistentenknoten), 282  
 Root (Assistentenknoten), 273  
 Start (Assistentenknoten), 289  
 Steuerelement (Assistentenknoten), 292  
 Steuerelemente, 261  
 String (Assistentenknoten), 291  
 Timer (Assistentenknoten), 290  
 Transition (Assistentenknoten), 285  
 Twips, 260  
 Übergänge, 263  
 Übergeordnete Objekte, 261  
 Untergeordnete Objekte, 261

Variablen, 262  
 Vollständiger Name eines Objekts, 262  
 Workflow-Aktivitäten, 125  
 Workflow-Aufgaben, 126  
 Workflow-Ausführungsgruppen, 126  
 Workflow-Ereignisse, 125  
 Workflow-Instanzen, 125  
 Workflow-Modelle, 124  
 Workflows, 124  
 Workflow-Übergänge, 125  
 DELETE (AQL), 60  
 Druck (Aktionstyp), 100  
 Druck von Berichten, 92  
 DUPLICATE (AQL), 60

## E

Einrichtung (Aktionstyp), 100  
 Empfängerlisten - Mitteilungen, 349  
 Empfangsbestätigungen, 121 , 105  
 Export, 185-197  
 Abfragen, 193  
 Ansichten (Registerkarte) - Anzeigen, 195  
 Ausgabeformat von Skripten, 193  
 Empfehlungen, 189  
 Exportabfragen, 190  
 Exportskripte  
 Ausführung, 195  
 Ausführung - DOS, 196  
 Ausführung - Prozess, 196  
 Konzeption, 189  
 Filter, 192  
 Optionen (Registerkarte) - Anzeigen, 193  
 Prozess, 186  
 Kontextmenü, 187  
 Skripte, 187  
 Sortieren, 191  
 SQL-Ansichten, 188  
 Erstellung, 195  
 Löschen, 195  
 Vorgehensweise, 189  
 Vorschau, 193  
 Zeitzonen, 247  
 Zu exportierende Spalten, 191

- Zu extrahierende Daten, 191
- Zugriffsberechtigungen, 187
- Exportabfragen
  - Definition, 186
  - Konzeption, 190
- Exportskripte
  - Ausführung, 195
  - Definition, 185
  - Konzeption, 189
- Externe Beziehungen
  - Operatoren, 31
  - ZAHL, 28
- Externe Schlüssel - Abfragen, 35

## F

- Fehlermeldungen - Skripte, 217
- Fehler - Skripte, 217
- Felder
  - (Siehe auch Anlegen von Historien)
  - Abfragen, 66 , 48
  - Leere Felder, 68
- Feste Texte - Formulare, 74
- FILEEDIT (Assistentenknoten), 322
- Filter - berechnete Felder, 257
- Finish (Assistentenknoten), 287
- Floor (AQL), 64
- Formel (Formular), 77
- Formeln - Formulare, 75
- Formulare
  - (Siehe auch Berichte)
  - Anführungszeichen, 75
  - Bearbeiten, 73
  - Bilder, 75
  - Definition, 71
  - Eigenschaften von Objekten, 76
  - Erstellung, 73
  - Feste Texte, 75 , 74
  - Filtern von Formularen, 73
  - Listen, 75
  - Module, 81
  - Objekte
    - Bild, 78
    - Eigenschaften, 76

- Farbe, 77
- Formel, 77
- Listeninhalt, 78
- Mehrfachauswahl, 76
- Position und Größe, 76
- Rahmen, 77
- Schriftart, 77
- Seitenausrichtung, 77
- Text, 77
- Textausrichtung, 77
- Verknüpfung - Liste, 78
- Raster, 79
- Regelmäßig verwendete Berichte, 80
- Schaltflächen - Zuordnen, 81
- Seiteneinrichtung, 79
  - Fußzeilen, 79
  - Kopfzeilen, 79
  - Ränder, 79
- Symbole, 74
- Variablen, 74
- Vorkonfigurierte Formulare, 71
- Formularraster, 79
- Formularsymbole, 74
- Frage (Workflow-Aktivitäten), 155
- FROM (AQL), 54
- FullName (AQL), 67
- FullName (Feld), 34
  - Abfragen, 67
- Funktionsbereiche - Aktionen, 96
- Fußzeilen - Formulare, 79

## G

- g\_lTimeZoneCheckInMns (am.ini), 246
- gbbase.wiz, 276
- gbbase.xml, 86 , 72
- Getdate (AQL), 63
- GROUP BY (AQL), 56

## H

- HAVING (AQL), 57
- Hierarchische Tabellen - Abfragen, 67
- Hintergrundfarbe (Formulare), 77
- Historien

Datenbank - Import, 18  
 Löschen, 21  
 Historienzeilen (Siehe Anlegen von Historien)  
 Hochformat - Formulare, 79  
 Hour (AQL), 63  
 HoursDiff (AQL), 63

## I

Import von Zeitzonen, 247  
 Index  
 (Siehe auch Abfragen)  
 INSERT (AQL), 58  
 IS NOT NULL (AQL), 52  
 IsNull (AQL), 65  
 IS NULL (AQL), 52

## K

Kalender, 223-231  
 Allgemeine Informationen, 225  
 Änderungen - Berücksichtigung, 224  
 Auswirkung auf andere Funktionalitäten, 224  
 Erstellung, 225 , 224  
 Funktionsweise, 224  
 Genauigkeit, 227  
 Registerkarte "Arbeitszeiten", 225  
 Steuerelement, 230  
 Vorschau, 223  
 Knoten (Assistent)  
 Definition, 261  
 Struktur und Syntax, 266  
 Kommentare - Skripte, 216  
 Konfiguration der Felder  
 (Siehe auch Skripte)  
 Konstanten  
 Abfragen, 49  
 Assistentenknoten, 268  
 Kopfzeilen von Formularen, 79

## L

LABEL (Assistentenknoten), 305  
 Left (AQL), 62  
 Length, 62  
 LIKE (AQL), 52

SQL Anywhere - Beschränkungen, 53  
 LINKEDIT (Assistentenknoten), 317  
 LISTBOX (Assistentenknoten), 300  
 Liste exportieren (Menü), 187  
 Liste konfigurieren (Menü), 41 , 38  
 Listen  
 Berechnete Felder, 257  
 Formulare, 75  
 Listeninhalt (Formulare), 78  
 LocalToDbDate (AQL), 63  
 Long (Assistentenknoten), 291  
 Lower (AQL), 62  
 Ltrim (AQL), 62

## M

MAPI (Siehe Nachrichtensystem)  
 Max (AQL), 61  
 Merkmale  
 (Siehe auch Anlegen von Historien)  
 Abfragen, 68 , 36  
 Methoden - Assistenten, 270  
 Microsoft SQL Server  
 Abfragen  
 Sortierreihenfolge, 39  
 Min (AQL), 61  
 Minute (AQL), 63  
 MinutesDiff (AQL), 63  
 Mitteilungen  
 Anzeigen, 349  
 Bedeutung, 348  
 Definition, 347  
 Empfängerlisten, 349  
 Erstellung, 348  
 Funktionsweise, 348  
 Lesen, 348  
 Mitteilungen, 348  
 Mod (AQL), 64  
 Module  
 Berichte, 93  
 Formulare, 81  
 Month (AQL), 62

**N**

Nachricht (Aktionstyp), 98  
 Nachrichten (Menü), 120  
 Nachrichtensystem, 119-121  
   Abfragen, 120  
   Empfangsbestätigungen, 121  
   Funktionsweise, 119  
   Internes Nachrichtensystem - Adressen, 99  
   Senden, 120  
 Nachrichtensystem (Registerkarte), 104  
 Nebenabfragen, 47  
 normal.dot, 109  
 NOT LIKE (AQL), 52  
 NULL (AQL), 31  
 NUMBOX (Assistentenknoten), 324

**O**

Objekte, 261  
 Operatoren - Abfragen, 51  
 OPTIONBUTTONS (Assistentenknoten), 299  
 OR (AQL), 51  
 Oracle for WorkGroups  
   Abfragen  
     Sortierreihenfolge, 39  
 ORDER BY (AQL), 58  
 Ortszeit mit Serverzeit vergleichen (Option), 246, 243

**P**

Page (Assistentenknoten), 282  
 Programmierbare Funktion, 212  
 PROGRESSBAR (Assistentenknoten), 305

**Q**

QBE  
   Beispiel für die Erstellung eines  
   Abfrageassistenten, 336  
 Querformat - Formulare, 79

**R**

Rahmen (Formulare), 77  
 Referenzobjekt (Feld), 105

Referenzobjekt - Workflow, 151  
 reports.txt, 84, 84  
 Right (AQL), 62  
 Root (Assistentenknoten), 273  
 Round (AQL), 64  
 Rtrim (AQL), 62

**S**

Schriftart (Formulare), 77  
 Second (AQL), 63  
 SecondsDiff (AQL), 63  
 Seitenansicht von Berichten, 88  
 Seitenausrichtung (Formulare), 77  
 Seiteneinrichtung - Formulare, 79  
 SELECT (AQL)  
   Beschränkungen, 54  
   Nebenabfragen - Warnung, 47  
 Self (AQL), 32  
 Skript (Aktionstyp), 99  
 Skriptbibliotheken, 210  
 Skripte, 199-221  
   Anwendungsbereich, 201  
   Assistentenknoten, 269  
   BASIC-Funktionen  
     Einführung, 202  
     Funktionstypen, 205  
     Hilfe, 200  
     Klassifizierung, 207  
     Namen - Beschränkungen, 211  
     Parametertypen, 205  
     Programmierbare Funktion, 204  
     Typen - Beschränkungen, 206  
     Vordefinierte Funktionen, 203  
 Bearbeiten, 199  
 Beispiele, 220, 217  
 Bibliotheken, 210  
 CurrentUser, 215  
 Daten, 213  
 Dauer, 213  
 Erste Schritte, 207  
 Fehlermeldungen, 217  
 Groß-/Kleinschreibung, 209  
 Kommentare, 216

Schreibweise - Konventionen, 201  
 Sonstige Hinweise, 212  
 Stunden, 213  
 Systemaufzählungen, 214  
 Tipps, 212  
 Verknüpfungen - Identifizierung, 201  
 sLvl (Feld) - Abfragen, 67  
 sLvl - Abfragen, 35  
 SMTP (Siehe Nachrichtensystem)  
 Sommerzeit, 239  
 Sortieren  
 (Siehe auch Abfragen)  
 SQL  
 Abfragen, 61  
 Datenbank - Änderung, 25  
 SQL-Ansichten (Siehe Export)  
 SQL Anywhere - Sortieren und Abfragen, 37  
 SQL und AQL im Vergleich, 25  
 Standardbrief.doc, 109  
 Standardwerte  
 Felder (Siehe Skripte)  
 Merkmale (Siehe Skripte)  
 Standardwert wiederherstellen (Option), 329  
 Start (Assistentenknoten), 289  
 Steuerelement (Assistentenknoten), 292 , 292  
 Steuerelemente, 261  
 String (Assistentenknoten), 291  
 Stunden - Skripte, 213  
 Substring (AQL), 62  
 Sum (AQL), 61  
 Sybase SQL Anywhere  
 Abfragen  
 Sortierreihenfolge, 39  
 Aktionen, 115  
 Sybase SQL Server  
 Abfragen  
 Sortierreihenfolge, 39  
 Systemaufzählungen - Skripte, 214  
 Assistenten, 337  
 Systemaufzählungen - Werte, 34

## T

Tabellen

(Siehe auch Anlegen von Historien)  
 Workflows, 128  
 Test / Skript (Workflow-Aktivitäten), 158  
 Test von Aktionen, 115  
 Text (Formulareigenschaft), 77  
 Textausrichtung (Formulare), 77  
 TEXTBOX (Assistentenknoten), 319  
 Textfarbe (Formulare), 77  
 TICKEDIT (Assistentenknoten), 323  
 Timer (Assistentenknoten), 290  
 TIMESPANEDIT (Assistentenknoten), 323  
 Transition (Assistentenknoten), 285  
 Trunc (AQL), 64  
 Twips, 260  
 tz.scr, 235

## U

UNION (AQL), 48  
 UNIX - Berichte - Beschränkungen, 83  
 Untergeordnete Objekte, 261  
 UPDATE (AQL), 59  
 Upper (AQL), 62  
 useSQL92Join (amdb.ini), 28

## V

Variablen  
 Aktionen, 114  
 Definition, 262  
 Formulare, 74  
 Verknüpfungen  
 (Siehe auch Anlegen von Historien)  
 Abfragen, 66 , 66 , 48  
 Fehlende Verknüpfung, 68  
 Verknüpfung - Liste (Formulare), 78  
 VIM (Siehe Nachrichtensystem)  
 Vollständiger Name eines Objekts, 262

## W

Warnungen, 177 , 176  
 WeekDay (AQL), 63  
 Werteliste (Siehe Aufzählungen)  
 WHERE (AQL), 55  
 Workflow-Aktivitäten, 154



- Definition, 125
- Grafikeditor, 130
- Workflow-Aktivitätenmodelle, 160
- Workflow-Aufgaben, 161 , 126
- Workflow-Ausführungsgruppen, 178 , 126
- Workflow-Editor, 130
- Workflow-Ereignisse, 164
  - Definition, 125
  - Editor, 131
- Workflow-Fälligkeiten, 177
- Workflow-Fristen, 176 , 176
- Workflow-Instanzen
  - Beschränkungen, 152
  - Definition, 125
  - Löschen, 179
- Workflow-Kontext, 151 , 151
- Workflow-Modelle (Siehe Workflows)
  - Definition, 124
- Workflow-Modelle (Menü), 127
- Workflow neu starten, wenn bereits ein anderer
- Workflow läuft (Option), 168
- Workflow-Rollen, 153
- Workflows, 123-184
  - Aktionen - Auslösung, 156
  - Aktivitäten, 154
    - Auslösung, 160
    - Automatische Aktion, 157
    - Bearbeiter - Aktivitätentypen, 153
    - Beginn, 159
    - Benutzeraktion, 156
    - Frage, 155
    - Modelle, 160
    - Registerkarte "Fristen" - Beschränkungen, 176
    - Registerkarte "Warnungen" - Beschränkungen, 177
    - Test / Skript, 158
  - Aktivitätenmodelle - Beschränkungen, 160
- Aufgaben, 161
  - Automatische Aktionen, 162
  - Benutzeraufgaben, 163
  - Erstellung, 161
  - Laufende Aufgaben, 162
- Skript, 162
- Test, 162
- Verwaltung, 164
- Warnungen - Beschränkungen, 162
- Zuordnung, 164
- Ausführungsgruppen, 178
- Auslösung - Beschränkungen, 166
- Beispiel, 133
  - Aktivitäten, 140
  - Ausgangsereignis, 146
  - Auslösung, 148
  - Einstellung der Ereignisparameter, 145
  - Übergänge, 147
  - Voraussetzungen, 135
- Dauer, 177
- Definition, 124
- Definitionen, 123
- Einrichtung, 133
- Ereignisse, 164
  - Abschließendes Ereignis, 175
  - Aktivierung, 168
  - Aktivität Beginn, 165
  - Bearbeitung, 169
  - Bei Datenbankänderung, 165
  - Benutzer, 165
  - Regelmäßige Überwachung, 167
  - System, 164
  - Warnung, 165
- Fälligkeiten, 177
- Fristen, 178 , 176
- Funktionsweise, 127
- Grafikeditor, 130
- Instanzen
  - Automatisierung des Löschvorgangs, 180
  - Löschen, 179 , 179
- Komplexe Eingabebedingungen, 161
- Kontext, 151
- Protokoll, 179
- Rollen, 153
- SQL-Namen - Beschränkungen, 159
- Synchroner Workflow, 173
- Tabellen, 128
- Übergänge, 176

Warnungen, 176  
Workflow-Übergänge, 176  
  Definition, 125  
  Editor, 132

## Y

Year (AQL), 62

## Z

Zahlen - Abfragen, 67  
Zeitzone des Datenbankservers prüfen (Option),  
243  
Zeitzone  
  AssetCenter Server - Beschränkungen, 243  
  Auswirkung auf bestimmte Vorgänge, 243  
  Erstellung, 235  
  Export, 247  
  Funktionsprinzip, 242  
  Import, 247  
  Sommerzeit, 235  
  Uhrzeit des Clients, 245  
  Verbindungen, 245  
  Verfügbarkeit, 233  
  Verwaltung, 235  
  Winterzeit, Sommerzeit, 242  
  Zweck, 233  
Zeitzone - Einrichtung, 234  
Zeitzone verwenden (Option), 244 , 234  
Zugriffsberechtigungen - berechnete Felder, 256



