

AssetCenter™

Version 3.5

Referenzhandbuch:

Verwaltung und Verwendung für fortgeschrittene Benutzer

Band 2

ITEM ACT-3.5X-DE-000808



© Peregrine Systems, Inc., 1999-2000. Alle Rechte vorbehalten.

Laufzeitversion von Sybase SQL Anywhere: © Copyright Sybase, Inc. 1992–1995 sowie für einige Teile © Rational Systems, Inc. 1992–1994.

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.

Das vorliegende Dokument enthält zahlreiche Markennamen. Dabei handelt es sich größtenteils um eingetragene Warenzeichen.

Peregrine Systems, ServiceCenter AssetCenter, InfraCenter for Workgroups und InfraTools 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 von Peregrine Systems, Inc. geändert werden.

Da die Software fortlaufenden Änderungen unterliegt, stimmt die mitgelieferte Dokumentation eventuell stellenweise nicht mit der verwendeten Softwareversion überein. Die vorgenommenen Änderungen beeinträchtigen jedoch in keiner Weise das Verständnis der Software und der Handbücher. Informationen zu den letzten Änderungen finden Sie in der Datei "readme.txt".

Die Namen der in diesem Handbuch, in der Beispieldatenbank 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.

Integrität der AssetCenter und InfraCenter for Workgroups Daten

AssetCenter und InfraCenter for Workgroups bieten eine breit gefächerte Funktionspalette. Für dieses umfangreiche Leistungsspektrum sind komplexe Datenbankstrukturen erforderlich: Die Datenbank umfasst zahlreiche Tabellen, Felder, Verknüpfungen und Indizes, wobei einige Interimstabellen nicht auf der grafischen Benutzeroberfläche angezeigt werden und bestimmte Verknüpfungen, Felder und Indizes von der Software automatisch erstellt, gelöscht und geändert werden.

Um die Integrität der Daten zu gewährleisten, darf der Datenbankinhalt nur mit den dazu vorgesehenen Schnittstellen (grafische Oberfläche, API, Importprogramm, Webschnittstelle, Gateways) geändert werden. **Ändern Sie die Struktur und den Inhalt der Datenbank niemals mit anderen Mitteln als den von AssetCenter oder InfraCenter for Workgroups zu diesem Zweck vorgesehenen**, da es dadurch mit größter Wahrscheinlichkeit zu schweren Fehlern in der Datenbank kommt: Verlust oder ungewollte Änderung von Daten oder

Verknüpfungen, Erstellen von Scheinverknüpfungen und –datensätzen, schwere Fehlermeldungen u.a. Änderungen dieser Art an der Datenbank führen überdies zum Erlöschen der Garantie- und Support-Ansprüche an die Peregrine Systems Gruppe.

Unterstützte Umgebungen

AssetCenter und InfraCenter for Workgroups können nur in bestimmten Umgebungen eingesetzt werden. Eine Liste dieser Umgebungen finden Sie im Handbuch "Installations- und Aktualisierungshandbuch". Bei Verwendung in anderen Umgebungen trägt der Benutzer das volle Risiko. Änderungen an der Datenbank im Rahmen einer Verwendung in anderen Umgebungen als den von AssetCenter und InfraCenter for Workgroups vorgesehenen führen zum Erlöschen der Garantie- und Support-Ansprüche an die Peregrine Systems Gruppe.

Vorwort

Einführung

AssetCenter ist ein umfassendes System zur Verwaltung technischer Aktiva. Das Softwarepaket setzt sich aus fünf voll integrierten Funktionen zusammen, die um eine Standardanwendung herum aufgebaut sind, das den Mittelpunkt des Informationssystems bildet:

- Vermögensverwaltung
- Helpdesk-Verwaltung
- Beschaffungsverwaltung
- Verwaltung von Verträgen, dem Leasing und der Finanzierung
- Kostenverwaltung

Die außerordentlich gelungene Integration ermöglicht den Zugriff auf sämtliche Funktionen über ein und dieselbe Benutzeroberfläche und das Arbeiten in einer einzigen Datenbank, wodurch das Duplizieren von Informationen entfällt.

Dank der funktionellen Dimension können Sie die Benutzeroberfläche durch eine gezielte Auswahl von Funktionen an Ihre Bedürfnisse anpassen. Die Oberfläche wird dadurch erheblich übersichtlicher.

Ziel der Referenzhandbücher

In den folgenden Handbüchern finden Sie eine detaillierte und systematische Beschreibung der Funktionsweise von AssetCenter.

- "Allgemeine Ergonomie und Referenztabellen"
- "Vermögensverwaltung"
- "Beschaffungsverwaltung"
- "Vertrags-, Leasing- und Finanzverwaltung"
- "Kostenverwaltung"
- "Verwaltung und Verwendung für fortgeschrittene Benutzer"

Die oben aufgeführten Handbücher sind auch als Online-Hilfe mit gleichem Inhalt verfügbar.

Weiterhin liegt ein Dokument vor, in dem ein Inhaltsverzeichnis sowie ein Index für sämtliche oben genannten Handbücher enthalten ist.

Ziel dieses Handbuchs

Dieses Handbuch richtet sich an fortgeschrittene Administratoren und Benutzer von AssetCenter. Dementsprechend finden Sie auf den folgenden Seiten eine detaillierte Beschreibung der Verwaltung der Asset Center Datenbank und der Verwendung von Funktionen, die die Mehrheit der Benutzer normalerweise nicht verwenden müssen.

Peregrine Systems kontaktieren

Hauptsitz

Peregrine Systems, Inc.
3611 Valley Centre Drive
San Diego, CA 92130
USA
Tel.: +1 858 481 5000 oder 800 638 5231
Fax: +1 858 481 1751
Website: <http://www.peregrine.com>

Support:
Tel.: +1 858 794 7402 oder 800 960 9998
Fax: +1 858 794 6028
E-Mail: support@peregrine.com
Website: <http://support.peregrine.com>
Öffnungszeiten: Montag bis Freitag von 05:00 bis 17:30 h (Pazifik-Normalzeit)

Deutschland und Osteuropäische Länder

Peregrine Systems GmbH
Bürohaus ATRICOM
Lyoner Straße 15
60528 Frankfurt
Deutschland
Tel.: +49 (0)(69) 6 77 34-0
Fax: +49 (0)(69) 66 80 26-26

Support:
Tel.: 0800 27 73 823
Fax: +49 (0)(69) 66 80 26-26
E-Mail: psc@peregrine.de
Öffnungszeiten: Montag bis Freitag von 08:00 bis 17:00 (Ortszeit)

Frankreich, Spanien, Griechenland und Afrika (mit Ausnahme von Südafrika)

Peregrine Systems
Tour Franklin - La Défense 8
92042 Paris - La Défense Cedex
Frankreich
Tel.: +33 (0)1 47 73 11 11
Fax: +33 (0)1 47 73 11 12

Support:
Tel.: +33 (0) 800 505 100
Fax: +33 (0)1 47 73 11 61
E-Mail: frsupport@peregrine.fr
Öffnungszeiten: Montag bis Freitag von 08:00 bis 18:00 (Ortszeit)

Großbritannien

Peregrine Systems, Ltd.
Ambassador House
Paradise Road
Richmond
Surrey TW9 1SQ
Großbritannien
Tel.: +44 (0)181 332 9666
Fax: +44 (0)181 332 9533

Support:
Tel.: +44 (0)181 334 5844 oder 0800 834 7700
Fax : +44 (0)181 334 5890
E-Mail: uksupport@peregrine.com
Öffnungszeiten: Montag bis Freitag von 08:00 bis 18:00 (Ortszeit)

Dänemark, Norwegen, Finnland und Island

Peregrine Systems A/S
Naverland 2, 12th fl.
DK-2600 Glostrup
Dänemark
Tel.: +45 43 46 76 76
Fax : +45 43 46 76 77

Support:
Tel.: +45 77 31 77 76
Fax: +45 43 46 76 77

E-Mail: support.nordic@peregrine.com
Öffnungszeiten: Montag bis Freitag von 08:30 bis 04:30 (Ortszeit)

Niederlande, Belgien und Luxemburg

Peregrine Systems BV
Botnische Golf 9a
Postbus 244
3440 AE Woerden
Niederlande
Tel.: +31 (0) 348 43 7070
Fax : +31 (0) 348 43 7080

Support:
Tel.: 0800 0230889 (Niederlande)
oder 0800 74747575 (Belgien und Luxemburg)
Fax: +31 (0) 348 43 7080
E-Mail: benelux.support@peregrine.com
Öffnungszeiten: Montag bis Freitag von 08:00 bis 18:00 (Ortszeit)

Singapur

Peregrine Systems Pte. Ltd
#03-16
CINTECH III
77 Science Park Drive
Singapore Science Park
118256
Singapur
Tel.: +65 778 5505
Fax: +65 777 3033

Italien

Peregrine Systems, S.r.l.
Via Monte di Pietà, 21
I-20121 Mailand
Italien
Tel.: +39 (02) 86337230
Fax: +39 (02) 86337720

Support:
Tel.: +39 (02) 86337230
Fax: +39 (02) 86337400

Japan

Peregrine Systems K.K.
Level 32, Shinjuku Nomura Building
1-26-2 Nishi-shinjuku, Shinjuku-ku
Tokio 163-0532
Japan

Tel.: +81 (3) 5322-1350
Fax: +81 (3) 5322-1352

Support:
Tel.: +81 (3) 5322-1350
Fax: +81 (3) 5322-1352
E-Mail: glipper@Peregrine.com

Schweden

Peregrine Systems AB
Frösundaviks Allé 15, 4th floor
S-169 70 Solna
Schweden

Tel.: +46 (0)8-655 36 04
Fax : +46 (0)8-655 26 10

Support:
Tel.: +45 77 31 77 76
Fax: +45 43 46 76 77
E-Mail: nordic@peregrine.com
Öffnungszeiten: Montag bis Freitag von 08:30 bis 04:30 (Ortszeit)

Schreibweisenkonventionen

Die Befehle werden in der Dokumentation folgendermaßen wiedergegeben:

[]	Fakultative Parameter stehen in eckigen Klammern, die bei der Eingabe des Befehls jedoch wegfallen. Ausnahme: In BASIC-Skripts umschließen eckige Klammern den Zugriffspfad auf die Datenbankdaten. Diese Klammern müssen im Skript vorhanden sein. Ein Beispiel: <code>[Verknüpfung.Verknüpfung.Feld]</code>
< >	Wörtlich zu nehmende Parameter sind mit spitzen Klammern dargestellt. Bei der Eingabe des Befehls werden sie nicht eingegeben und der eingeklammerte Text wird durch die erforderlichen Informationen ersetzt.
{ }	Geschweifte Klammern umschließen mehrere Parameter, von denen einer gewählt werden kann. Bei der Eingabe des Befehls werden sie nicht verwendet.
	Die in geschweiften Klammern stehenden Parameter werden durch einen vertikalen Strich voneinander getrennt.
*	Ein Stern rechts neben einer Klammer gibt an, dass die eingeklammerte Formel mehrmals wiederholt werden kann.

Die nachstehenden Formate haben besondere Bedeutungen:

Feste Schrift	DOS-Befehl
<code>Beispiel</code>	Beispielcode oder -befehl
...	Ausgelassener Code oder Befehl
Objektname	Die Namen von Feldern, Registerkarten, Menüs und Dateien sind fett gedruckt.
Hinweis	Wichtige Anmerkung

Teilen Sie uns Ihre Anregungen mit

Unser Ziel ist es, Ihnen in unserer Dokumentation aktuelle und hilfreiche Informationen zur Verfügung zu stellen.

Für Ihre Anregungen sind wir Ihnen sehr dankbar.

Richten Sie Ihre Kommentare und Anregungen bitte an folgende Adresse:
documentation@peregrine.com.

Inhaltsverzeichnis

Kapitel 1 - Datenbankoptionen	1
Kapitel 2 - Struktur der AssetCenter-Datenbank	5
Einführung	5
Definition einer Datenbank	7
Struktur der Dateien database.txt und tables.txt	7
Beschreibung der Tabellen	8
Identifizierung der AssetCenter-Tabellen	9
Beschreibungsfolgen der AssetCenter-Tabellen	9
Beschreibung der Felder	10
Identifizierung der AssetCenter-Felder	11
Feldtypen und Eingabeformate	11
Felder mit dem Namen <i>dtLastModif</i>	15
Felder vom Typ <i>FullName</i>	16
Beschreibung der Verknüpfungen	17
Identifizierung der Verknüpfungen	17
Typen von Verknüpfungen	19
Kardinalcharakter einer Verknüpfung	19
Übergangstabellen	20
Spezifische Verknüpfung	22
Beschreibung der Indizes	23
Identifizierung der Indizes	23
Einmaligkeit der Feldwerte eines Index in einer Tabelle	23
Datenbank anpassen	24
Tabellen anpassen	24
Felder und Verknüpfungen anpassen	24
Index anpassen	25
Standardwerte für Felder und Verknüpfungen	26
Zähler in den Standardwerten für Felder	27
Fehler bei der Beschreibung von Standardwerten	28
HTML-Befehle in der Hilfe zu Feldern und Verknüpfungen	28
Kapitel 3 - Erstellung, Anpassung und Beschreibung der AssetCenter-Datenbank	31
Arbeiten mit <i>AssetCenter Database Administrator</i>	32
<i>AssetCenter Database Administrator</i> starten	33
Beschreibung des Menüs <i>Datei</i> in <i>AssetCenter Database Administrator</i>	33
Beschreibung des Menüs <i>Bearbeiten</i> in <i>AssetCenter Database Administrator</i>	36

Beschreibung des Menüs <i>Aktion</i> in <i>AssetCenter Database Administrator</i>	37
Datenbanken erstellen	45
Datenbanken anpassen	45
Beschädigte Datenbanken reparieren	51
AssetCenter-Datenbanken beschreiben	52
Parameter zur Beschreibung der Datenbank	53
Syntax der Beschreibungsmodelle	54
Feste Texte	55
Kommentare	55
Andere Modelle berücksichtigen	56
Komponenten durchsuchen, sortieren und filtern	56
Bedingungen <i>\$if...\$else...\$elseif...\$endif</i>	59
In den Beschreibungsmodellen verwendbare Funktionen	59
Werte einer Eigenschaft mit Hilfe einer im Modell definierten Funktion bearbeiten	63
Markierung am Ende des Abschnitts löschen	64
Anzahl der durchsuchten Komponenten zählen	65
Globale Variablen auf der Ebene eines Modells definieren	65
Informationen zu bestimmten Parametern zur Beschreibung der Datenbank	66
Instanz <i>Database</i>	67
Instanz <i>Table</i>	67
Instanz <i>Field</i>	69
Instanz <i>Link</i>	73
Instanz <i>Index</i>	75
Instanz <i>Script</i>	76
Globale Variablen	77
Beim Export generierte Werte	78
<hr/>	
Kapitel 4 - Verwaltung von Aufzählungen	79
Anpassbare Aufzählungen	79
Systemaufzählungen	81
<hr/>	
Kapitel 5 - Funktionsweise der Historienerstellung	83
<hr/>	
Kapitel 6 - Verwaltung der Zugriffe von Benutzern auf die AssetCenter-Datenbank	89
Nutzen und Funktionsweise der Zugriffsverwaltung	90
Sicherheit und Zuverlässigkeit der Daten gewährleisten	92
Definitionen	92
Definition eines Benutzerprofils	92
Definition einer Zugriffsberechtigung	92
Definition einer Zugriffsbeschränkung	93
Zugriffsbedingungen definieren	93
Benutzerprofile definieren	93
Zugriffsberechtigungen definieren	94
Zugriffsbeschränkungen definieren	96
AssetCenter-Benutzer definieren	97
Verbindungen von Benutzern verwalten	99
Arten des Zugriffs auf die Datenbank	99
Funktionsweise von Login-Scheiben	101
Login-Scheiben löschen	102
Kennwörter verwalten	103

Kennwort des Logins <i>Admin</i>	104
Kennwort des Logins <i>Admin</i> ändern	104
Benutzerkennwort	105
Verlust des Kennworts	105
<hr/>	
Kapitel 7 - Schreiben von AQL-Abfragen	107
Einführung	107
Abfragen in AssetCenter	107
AQL	108
Empfehlungen zum Schreiben von AQL-Abfragen	111
AQL-Beziehungen	112
Vorzüge der Datensätze mit dem Primärschlüssel <i>0</i>	113
Verwendung des Werts NULL	115
Self	116
CurrentUser	116
Systemaufzählungen	117
Hierarchische Tabellen	118
Vereinfachte AQL-Notationen	119
Sortiervorgänge und Indizes	121
Abfrageeditor	124
Allgemeine Funktionsweise	124
Zugriff auf den Abfrageeditor	125
Abfragen mit Hilfe des Abfrageeditors erstellen	126
Bei Abfragen verwendete Felder	128
Ausdrücke schreiben	129
Konstanten	130
Syntax der Programmiersprache AQL	130
Konventionen	131
Abfragesyntax	131
Klausel FROM – In einer Abfrage verwendete Tabellen	132
Elemente einer Abfrage	134
Klausel WHERE	140
Klausel GROUP BY	140
Klausel HAVING	141
Klausel ORDER BY	142
AQL-Funktionen	142
AQL-Funktionen vom Typ <i>Aggregat</i>	142
AQL-Funktionen vom Typ <i>String</i>	143
AQL-Funktionen vom Typ <i>Datum</i>	144
AQL-Funktionen vom Typ <i>Numerisch</i>	146
AQL-Funktionen vom Typ <i>Test</i>	147
Abfragebeispiele	147
<hr/>	
Kapitel 8 - Verwenden von Formularen	153
Definition eines Formulars	153
Formulare erstellen	154
Formulare und Objekte bearbeiten	154
Eigenschaften von Formularobjekten	157
Eingaberaster	159
Formularseiten einrichten	160
Einfache Berichte mühelos erstellen	161

Kapitel 9 - Bearbeiten von Berichten	163
Funktionsprinzip und Installation des Berichtgenerators	163
Detail eines Berichts	164
Crystal Reports-Berichte ändern	165
Crystal Reports-Statistiken	166
Detailberichte erstellen	167
<hr/>	
Kapitel 10 - Lesezugriff auf die Datenbank mit Hilfe des ODBC-Treibers von AssetCenter	171
Allgemeine Funktionsweise des Zugriffs auf die AssetCenter-Datenbank	172
Beispiel - Crystal Reports-Berichte mit Hilfe des ODBC-Treibers erstellen	174
<hr/>	
Kapitel 11 - Definition von Aktionen	177
Definition einer Aktion	178
Aktionen erstellen	178
Aktionstypen	178
Vorgehensweise	181
Informationen auf der Registerkarte <i>DDE</i> eingeben	182
Informationen auf der Registerkarte <i>Nachrichtensystem</i> eingeben	183
Beispiele für Aktionen	186
Beispiel für eine Aktion vom Typ <i>Programm</i>	186
Beispiel für eine Aktion vom Typ <i>DDE</i>	187
Beispiel für eine Aktion vom Typ <i>Nachricht</i>	191
Beispiel einer Aktion vom Typ <i>Skript</i>	191
Arbeiten mit Variablen	193
Sonderfall der Sybase SQL Anywhere-Engine	194
Aktionen testen	194
Aktionen ausführen	195
<hr/>	
Kapitel 12 - Verwaltung von Fälligkeiten mit Hilfe von AssetCenter Server	197
Allgemeine Funktionsweise von <i>AssetCenter Server</i>	198
<i>AssetCenter Server</i> ausführen	199
Hauptbildschirm von <i>AssetCenter Server</i>	202
Allgemeine Überwachungsoptionen von <i>AssetCenter Server</i>	203
Von <i>AssetCenter Server</i> überwachte Module	206
Zu überwachende Module wählen	206
Uhrzeiten der Überwachung durch <i>AssetCenter Server</i>	207
Überwachung der Historien durch <i>AssetCenter Server</i>	209
Überwachung des Lagers durch <i>AssetCenter Server</i>	209
Von <i>AssetCenter Server</i> überwachte Warnungen	210
Von <i>AssetCenter Server</i> berechnete Zahlungen	213
Von <i>AssetCenter Server</i> aktualisierte Verlustwerte	215
Von <i>AssetCenter Server</i> umgelegte Aufwandszeilen	215
Zeitzone des Datenbankservers überwachen	217
Präsenzsignal von <i>AssetCenter Server</i>	218
Suche nach neuen Ausführungsgruppen für Workflow-Modelle durch <i>AssetCenter Server</i>	218
Ausführung der Workflow-Regeln durch <i>AssetCenter Server</i>	218
Überwachung durch <i>AssetCenter Server</i> unmittelbar auslösen	220
<i>AssetCenter Server</i> über das Web verwalten	221

<i>AssetCenter Server</i> als NT-Dienst starten	221
Über das WEB auf <i>AssetCenter Server</i> zugreifen	222
<i>AssetCenter Server</i> -Dienst über das Web steuern	223
<hr/>	
Kapitel 13 - Verwaltung von Nachrichten	227
Allgemeine Funktionsweise des Nachrichtensystems	228
Zum Senden von Nachrichten über das externe Nachrichtensystem erforderliche Umgebung	230
<i>AssetCenter</i> für die Verwendung von Nachrichtensystemen konfigurieren	230
Informationen im Personendetail zur Verwendung des externen Nachrichtensystems eingeben	231
Nachrichten erstellen	235
Probleme bei der Herstellung einer Verbindung	235
<hr/>	
Kapitel 14 - Workflow-Modelle	239
Definition der Konzepte im Rahmen der Workflow-Modelle	239
Definition des Workflows	240
Definition einer Workflow-Aktivität	241
Definition eines Workflow-Ereignisses	241
Definition eines Workflow-Übergangs	242
Definition einer Workflow-Aufgabe	242
Definition des Bearbeiters einer Workflow-Aktivität	242
Definition einer Workflow-Ausführungsgruppe	242
Allgemeine Funktionsweise	243
Wichtigste Tabellen des Workflows	245
Arbeiten mit dem Workflow-Grafikeditor	246
Aktivitäten	247
Ereignisse	248
Übergänge	249
Weitere Funktionalitäten	250
Workflow-Modelle einrichten	250
Beispiel eines Workflow-Modells für die Bewilligung von Anforderungen	251
Ziel und Zweck	251
Vorbereitung	252
Aktivitäten erstellen	258
Ausgangsereignisse erstellen	263
Übergänge erstellen	264
Beispiele zum Auslösen des Workflows	265
Kontext eines Workflow-Modells	270
Kontext eines Workflows definieren	270
Referenzobjekt eines Workflow-Modells	270
Aktuelle Workflow-Instanzen für ein Objekt einschränken	271
Workflow-Rollen	272
Workflow-Aktivitäten	274
Aktivitäten vom Typ <i>Frage</i>	274
Aktivität vom Typ <i>Benutzeraktion</i>	275
Aktivitäten vom Typ <i>Automatische Aktion</i>	276
Aktivitäten vom Typ <i>Test / Skript</i>	277
Aktivität <i>Beginn</i>	278
Aktivitätenmodelle	278
Aktivitäten auslösen	279
Aufgaben	279

Aufgaben erstellen	280
Aktivität vom Typ <i>Automatische Aktion</i> oder <i>Test / Skript</i>	280
Aktuelle Aufgabenliste anzeigen	281
Benutzeraufgaben ausführen	281
Benutzeraufgaben zuordnen	282
Workflow-Aufgaben verwalten	282
Ereignisse	283
Ereignisse vom Typ <i>System</i>	283
Ereignisse vom Typ <i>Warnung</i>	284
Ereignisse vom Typ <i>Benutzer</i>	284
Allgemeine Bedingungen zur Aktivierung von Ereignissen	287
Bearbeitung der Ereignisse	288
Anwendung – Ausführung eines synchronen Workflows	291
Abschließendes Ereignis	292
Workflow-Übergänge	293
Warnungen und Fristen innerhalb eines Workflow-Modells	294
Fristen	294
Warnungen	295
Workflow-Ausführungsgruppen	296
Workflow-Protokoll	297
<hr/>	
Kapitel 15 - Import von Daten	299
Funktionsprinzip des Datenimports	300
Empfehlungen	301
Auflagen, die beim Import von Daten in ein Feld zu beachten sind	302
Textdateien oder ODBC-Datenbank importieren	306
Vorbereitung des Imports von Textdateien	307
Schritt 1 - Auswahl der zu importierenden Textdatei bzw. der ODBC-Datenbank	308
Schritt 2 - Beschreibung des Parsings der zu importierenden Dateien bzw. Tabellen	309
Schritt 3 - Beschreibung der zu importierenden Felder	314
Schritt 4 - Zuordnung der zu importierenden Felder zu den Feldern der AssetCenter-Datenbank	316
Schritt 4 - Zuordnung der einzelnen Textdateien oder Ausgangstabellen zu einer Zieltabelle	317
Schritt 4 - Zuordnung der zu importierenden Felder zu den Feldern der AssetCenter-Datenbank	318
Schritt 4 - Zuordnung zusätzlicher berechneter Felder zur Ausgangsdatei	323
Schritt 4 - Sonderfälle des Imports	327
Schritt 4 - Anwendungsbeispiele für Identifizierungsschlüssel	328
Schritt 5 - Parameter für die Übertragung der Daten	333
Schritt 6 - Übertragung der Daten	335
AssetCenter-Datenbank importieren	336
Importskripte speichern und ausführen	336
Importskript über eine 32-Bit-Eingabeaufforderung ausführen	339
<hr/>	
Kapitel 16 - Export von Daten und Verwaltung von SQL-Ansichten	343
Definitionen eines Skripts und einer Exportabfrage	343
Daten der AssetCenter-Datenbank exportieren	344
SQL-Ansichten der AssetCenter-Datenbank verwalten	346
Empfehlungen	347
Exportskripte definieren	347
Vorgehensweise	347
Exportabfragen definieren	348

Ausgabeformat eines Exportskripts definieren	351
SQL-Ansichten bearbeiten	352
Exportskripte ausführen	353
Exportskripte über <i>AssetCenter Export</i> ausführen	353
Exportskripte unter DOS ausführen	354
<hr/>	
Kapitel 17 - Arbeiten mit Skripten	357
Definition eines Skripts	358
Anwendungsbereiche von Skripten	359
Einführung in die Funktionen	360
Definition einer Funktion	360
Vordefinierte und programmierbare Funktionen	361
Typ der Funktionen und Funktionsparameter	363
Klassifizierung der BASIC-Funktionen	365
Erste Schritte zum Schreiben eines Skripts	365
Beispielszenario	365
Schritt 1 – Erstellen des Merkmals <i>Lernprogramm</i>	366
Schritt 2 – Öffnen des Bearbeitungsfensters	367
Schritt 3 – Analysieren und Schreiben des Algorithmus	367
Schritt 4 – Schreiben des BASIC-Programms	367
Schritt 5 – Testen des BASIC-Programms	368
Vorsichtsmaßnahmen und Tipps	369
Vorsichtsmaßnahmen beim Arbeiten mit programmierbaren Funktionen	369
Format der Konstanten vom Typ <i>Datum+Uhrzeit</i> in den Skripten	369
Format der Konstanten vom Typ <i>Dauer in den Skripten</i>	370
Lese- und Schreibzugriff auf den Wert einer Systemaufzählung	370
Die virtuelle Verknüpfung <i>CurrentUser</i>	372
Kommentar zu einem BASIC-Skript eingeben	373
Fehlermeldungen auslösen	373
Erstes Beispiel für ein Skript	374
Darstellung des Problems	374
Schritt 1 – Analysieren und Schreiben des Algorithmus	375
Schritt 2 – Schreiben des BASIC-Skripts	375
Schritt 3 – Testen des BASIC-Programms	376
Zweites Beispiel für ein Skript	376
Darstellung des Problems	377
Schritt 1 - Analysieren und Schreiben des Algorithmus	377
Schritt 2 – Schreiben des BASIC-Skripts	377
Schritt 3 – Testen des BASIC-Programms	378
<hr/>	
Kapitel 18 - Verwaltung von Kalendern	379
Allgemeine Übersicht und Funktionsweise eines Kalenders	379
Auswirkung der Kalender auf bestimmte Funktionalitäten	380
Vorgehensweise beim Erstellen von Kalendern	380
Beschreibung der Erstellung eines Kalenders	381
Allgemeine Informationen zur Eingabe des Kalenders	381
Eingaben auf der Registerkarte <i>Arbeitszeiten</i>	381
Eingaben auf der Registerkarte <i>Ausnahmen</i> eines Kalenders	383
Funktionsweise des Kalenders prüfen	386
<hr/>	
Kapitel 19 - Verwaltung von Zeitzonen	389
Vorteile der Verwaltung von Zeitzonen	389

Zeitzone einrichten	391
Zeitzone erstellen	391
Zeitzone verwalten	392
Format des Felds <i>SZ</i> -> <i>WZ</i>	392
Werte für das Argument <i><Year></i>	393
Werte für das Argument <i><DaylightInfo></i>	393
Beispiel zur Verwaltung der Sommerzeit	396
Verwaltung von Zeitzone in <i>AssetCenter Server</i>	399
Auswirkungen der Zeitzone auf verschiedene Vorgänge	400
Datenbanken erstellen	401
Verbindung zu einer Datenbank und Zeitzone	402
Import und Export	403
<hr/>	
Kapitel 20 - Arbeiten mit AssetCenter als DDE-Befehlsserver	405
Definition eines DDE-Servers	406
Dynamic Data Exchange aufrufen	406
DDE-Dienste (Service)	406
DDE-Thema	407
DDE-Befehl	407
Einführung in DDE-Befehle	408
Globale Befehle	409
Connect(<i><Cnx></i> , <i><User></i> , <i><Password></i>)	409
Disconnect()	410
ExecuteAction(<i><ActionName></i>)	411
ListAllTables(<i>[Mask]</i>)	411
ListOpenTables(<i>[Mask]</i>)	412
OpenView(<i><ViewName></i>)	413
Einer Tabelle zugeordnete Befehle	414
OpenTable(<i><Table></i>)	414
CloseTable(<i><Table></i>)	415
<i><Table>.GetRecordCount()</i>	416
<i><Table>.SetViewMode(<Mode>)</i>	416
<i><Table>.SetRecordMode(<Mode>)</i>	417
<i><Table>.ListAllFields([Mask])</i>	418
<i><Table>.ListAllLinks([Mask])</i>	419
<i><Table>.SetFilter(<Condition>)</i>	420
<i><Table>.SetSelection(<Condition>)</i>	421
<i><Table>.GetSelectionId()</i>	422
Einer Tabelle und einem Feld bzw. einer Verknüpfung zugeordnete Befehle	422
<i><Table>:<Objet>.AddLink()</i>	423
<i><Table>:<Item>.GetValue()</i>	424
<i><Table>:<Objet>.Highlight()</i>	425
<i><Table>:<Objet>.RemoveLink()</i>	425
<i><Table>:<Item>.SetFocus()</i>	426
<i><Table>:<Item>.SetValue(<Value>)</i>	427
<i><Table>:<Link>.SetValueWhere(<Condition>)</i>	428
<i><Table>:<Objet>.Show()</i>	429
Beispiele für DDE-Aufrufe	430
Erstes Szenario – AssetCenter interne DDE-Aufrufe	430
Einführung	431
Eingaben	431
Merkmal erstellen	432
Zweites Szenario – DDE-Aufrufe über Excel 97	432

Beschreibung des Makros	433
Quelle des Makros	433
Drittes Szenario – DDE-Aufrufe über Visual BASIC 5.0	434
Vorsichtsmaßnahmen	434
Programm ausführen	435
Quelle des Programms	437
<hr/>	
Kapitel 21 - Berechnete Felder	439
Definition eines berechneten Felds	440
Vorteile der berechneten Felder	440
Berechnete Felder erstellen	441
Einführung	441
Vorgehensweise	443
Arbeiten mit berechneten Feldern	446
<hr/>	
Kapitel 22 - Präsentation der Assistenten	449
Definition eines Assistenten	450
An wen richten sich die Assistenten?	450
Beschreibung verschiedener Assistenten	451
Assistenten zum Austauschen von Daten mit der AssetCenter-Datenbank	451
Unabhängige Assistenten	452
<hr/>	
Kapitel 23 - Erstellen von Assistenten	453
Verwendete Konventionen	455
Definitionen zu Assistenten	455
Strukturmodelle	458
Modell der Seite eines Assistenten	460
Allgemeine Informationen zu Assistenten	461
Allgemeine Struktur und Syntax eines Assistentenknotens	462
Eigenschaften eines Knotens	463
Verkettung von Assistenten	468
BASIC-Funktionen	469
Definition eines Knotens vom Typ <i>Root</i>	469
Syntax eines Knotens vom Typ <i>Root</i>	469
Eigenschaften des Knotens <i>Root</i>	470
Dem Knoten <i>Root</i> untergeordnete Knoten	474
Definition des Knotens <i>Page</i>	474
Syntax eines Knotens vom Typ <i>Page</i>	474
Eigenschaften eines Knotens vom Typ <i>Page</i>	475
Dem Knoten <i>Page</i> untergeordnete Knoten	477
Definition eines Knotens vom Typ <i>Transition</i>	477
Syntax eines Knotens vom Typ <i>Transition</i>	477
Eigenschaften eines Knotens vom Typ <i>Transition</i>	478
Besonderheiten eines Knotens vom Typ <i>Transition</i>	480
Definition eines Knotens vom Typ <i>Finish</i>	480
Definition eines Knotens vom Typ <i>Start</i>	481
Definition der Knoten vom Typ <i>Long</i> und <i>String</i>	481
Definition eines Knotens vom Typ <i>Steuerelement</i>	482
Allgemeine Syntax eines Knotens vom Typ <i>Steuerelement</i>	483

Steuerelementtypen mit ihren zugeordneten Eigenschaften	483
Gemeinsame Eigenschaften	484
Das Steuerelement <i>CheckBox</i>	486
Das Steuerelement <i>ComboBox</i>	486
Das Steuerelement <i>ListBox</i>	487
Das Steuerelement <i>Label</i>	490
Das Steuerelement <i>OptionButtons</i>	491
Das Steuerelement <i>ProgressBar</i>	491
Das Steuerelement <i>CommandButton</i>	492
Das Steuerelement <i>DBListBox</i>	493
Das Steuerelement <i>DBQueryBox</i>	497
Das Steuerelement <i>DBEdit</i>	500
Das Steuerelement <i>DBTable</i>	501
Das Steuerelement <i>DBPath</i>	501
Das Steuerelement <i>LinkEdit</i>	502
Das Steuerelement <i>TextBox</i>	502
Beispiel für die Erstellung eines Assistenten	503
Schritt 1 – Analysieren der Anforderungen	504
Schritt 2 – Definieren der Struktur des Assistenten	504
Schritt 3 – Schreiben der Struktur des Assistenten mit Hilfe der Skriptsprache	508
Arbeiten mit dem Grafikeditor	511
Überblick über die Benutzeroberfläche zur Bearbeitung des Assistenten	511
Neue Knoten erstellen	513
Eigenschaften eines Knotens bearbeiten	514
Assistenten ausführen und Fehler beheben	515
Häufig gestellte Fragen	515
<hr/>	
Kapitel 24 - Einstellen der Leistungsstärke von AssetCenter in einem WAN	519
Kapitel 25 - .ini-Dateien	523
Liste der verfügbaren <i>.ini</i> -Dateien	523
<i>.ini</i> -Dateien ändern	525

Kapitel 14 - Workflow-Modelle

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

- Definition der Konzepte im Rahmen der Workflow-Modelle
- Allgemeine Funktionsweise
- Wichtigste Tabellen des Workflows
- Arbeiten mit dem Workflow-Grafikeditor
- Workflow-Modelle einrichten
- Beispiel eines Workflow-Modells für die Bewilligung von Anforderungen
- Kontext eines Workflow-Modells
- Workflow-Rollen
- Workflow-Aktivitäten
- Aufgaben
- Ereignisse
- Workflow-Übergänge
- Warnungen und Fristen innerhalb eines Workflow-Modells
- Workflow-Ausführungsgruppen
- Workflow-Protokoll

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:

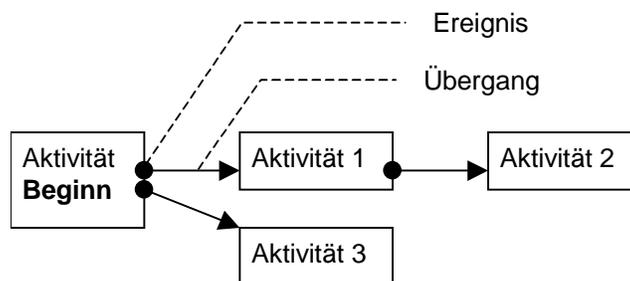
- Prozesse zur Bestätigung von Beschaffungsanforderungen
- Umzug von Vermögensgegenständen
- Andere

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



Vereinfachtes Modell eines Workflows

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:

- Eine auszuführende Arbeit. Diese Arbeit kann sowohl von einem Benutzer als auch automatisch vom 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 Aktivitätenausgabe 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 die Tabelle mit den Workflow-Rollen 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.

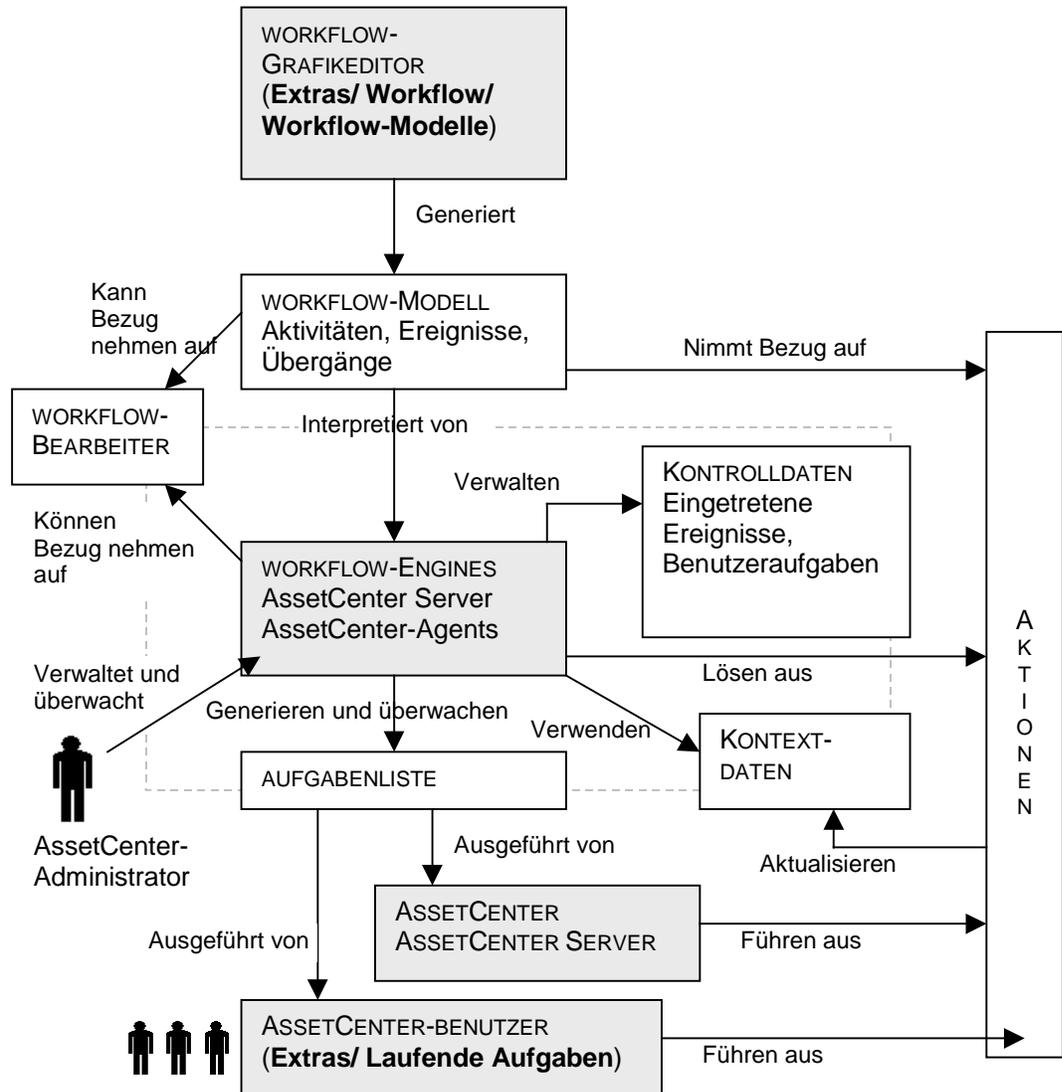
Die Workflow-Modelle werden von Workflow-Engines interpretiert. Bei den Workflow-Engines von AssetCenter handelt es sich entweder um AssetCenter Server-Programme, oder um AssetCenter-Agenten.

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:

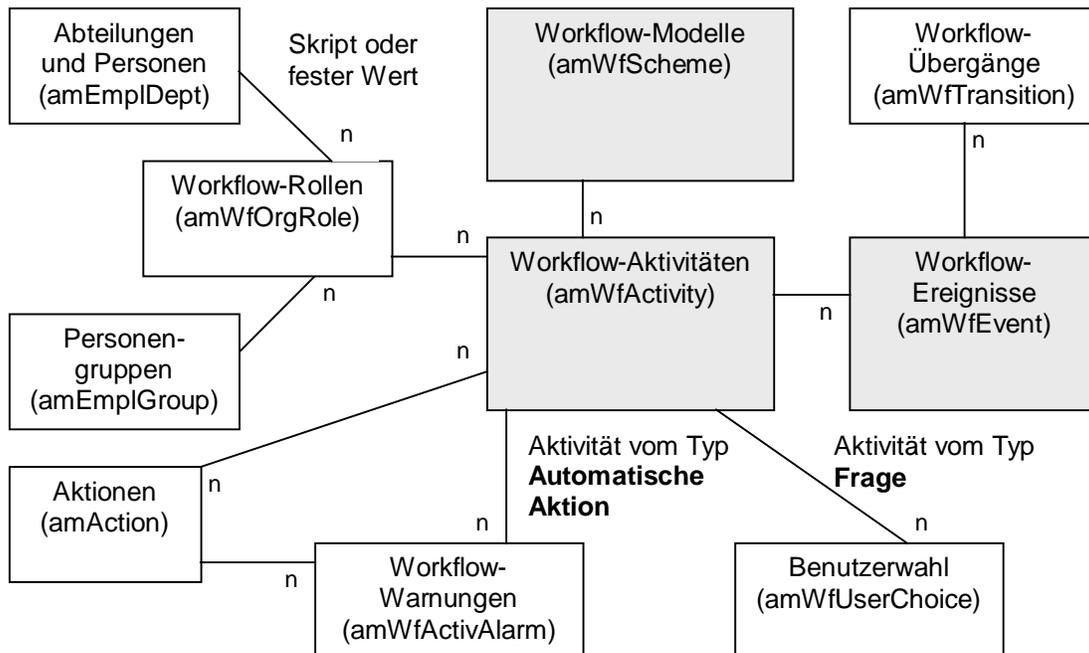


Übersicht über den Workflow in AssetCenter

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.

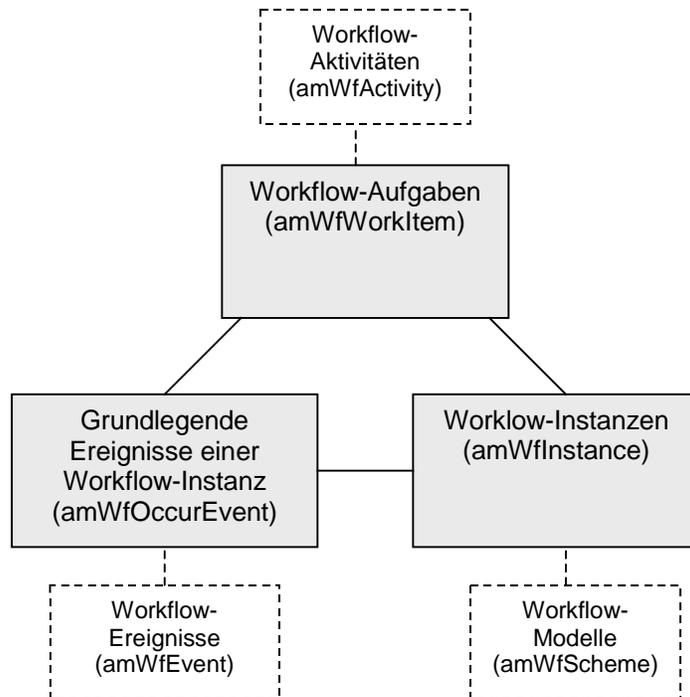
Wichtigste Tabellen des Workflows

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:



Zum Definieren eines Workflow-Modells hauptsächlich verwendete Tabellen

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



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:

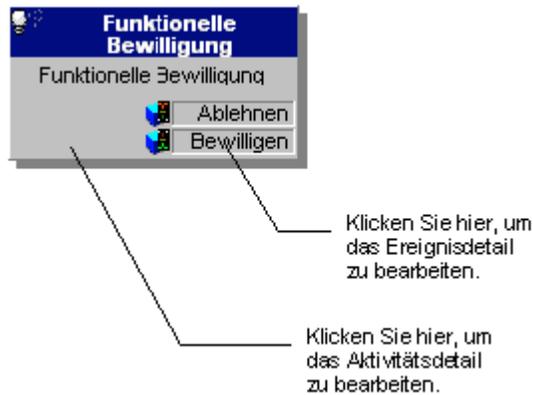
- Klicken Sie mit der rechten Maustaste auf einen leeren Bereich auf der Registerkarte **Aktivitäten**, und wählen Sie in dem daraufhin erscheinenden Kontextmenü die Option **Aktivität hinzufügen**. Es erscheint das Detailfenster der Aktivitäten; oder
- Klicken Sie auf die Schaltfläche  und dann auf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

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:

- Positionieren Sie den Cursor auf der 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:

- Positionieren Sie den Cursor auf der 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 Art der Bearbeitung eines Ereignisses (Feld **Bearbeitung** (SQL-Name: seProcessingMode) im Ereignisdetail), gehen Sie so 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:

- Wählen Sie durch Drücken der linken Maustaste das Ausgangsereignis für den Übergang.
- 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:

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

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

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

- ↗ Analysieren Sie die Geschäftsprozesse, für die Sie Modelle definieren wollen.
- ↗ Erstellen Sie folgende Elemente:
 - ❖ Workflow-Rollen
 - ❖ Aktionen
- ↗ Beim Erstellen der Workflow-Modelle definieren Sie folgende Elemente:
 - ❖ Aktivitäten, Ereignisse und Übergänge
 - ❖ Warnungen
- ↗ Definieren Sie die für die Workflow-Modelle geeigneten Ausführungsgruppen. Ordnen Sie die einzelnen Workflow-Modelle einer Ausführungsgruppe zu.
- ↗ Starten Sie das Programm 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.

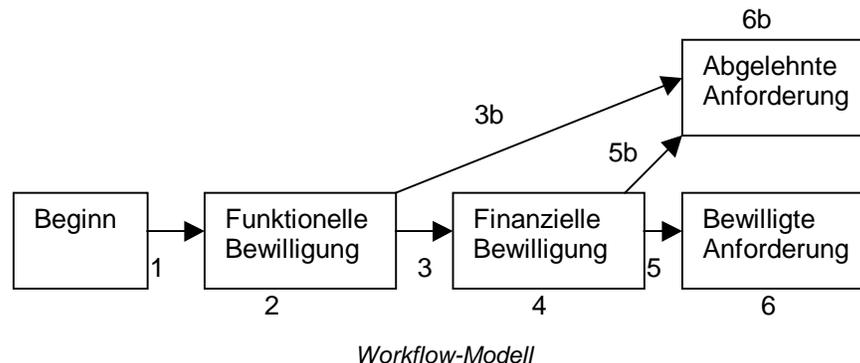
Beispiel eines Workflow-Modells für die Bewilligung von Anforderungen

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

- Ziel und Zweck
- Vorbereitung
- Aktivitäten erstellen
- Ausgangsereignisse erstellen
- Übergänge erstellen
- Beispiele zum Auslösen des Workflows

Ziel und Zweck

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



Im Folgenden sind die einzelnen Schritte des Workflows aufgeführt:

- ↳ Die Workflow-Instanz beginnt mit dem Vorliegen einer Anforderung, die bewilligt werden soll, also sobald im Feld **Bewilligungsstatus** (SQL-Name: seApprStatus) der Registerkarte **Bewilligung** des Anforderungsdetails der Wert **Zu bewilligen** erscheint.

- ↪ 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 darauf folgenden Tages nicht reagiert hat, sendet das System eine Warnmeldung.
- ↪ Wenn der Abteilungsleiter die Beschaffungsanforderung bewilligt, besteht der nächste Schritt in der finanziellen Bewilligung. Ansonsten ist die Anforderung abgelehnt.
- ↪ 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.
- ↪ Wenn der Finanzprüfer die Anforderung bewilligt, gilt die Anforderung als genehmigt. Ansonsten ist die Anforderung abgelehnt.
- ↪ Wenn die Anforderung bewilligt ist, ändert AssetCenter das Feld **Bewilligungsstatus** auf der Registerkarte **Bewilligung** des Anforderungsdetails, so dass der Wert **Bewilligt** erscheint. Wenn die Anforderung abgelehnt wird, ändert AssetCenter den Wert **Bewilligungsstatus** (SQL-Name: seApprStatus) auf der Registerkarte **Bewilligung** des Anforderungsdetails in den Wert **Abgelehnt**.
In beiden Fällen erhält der AssetCenter-Benutzer eine entsprechende Nachricht.

Vorbereitung

Wir empfehlen Ihnen, das Workflow-Modell in der Beispieldatenbank zu erstellen, die im Lieferumfang von AssetCenter enthalten ist, und eine Verbindung unter dem Login **Admin** herzustellen.

Bearbeiter erstellen

Bei den 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 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 die Schaltfläche .

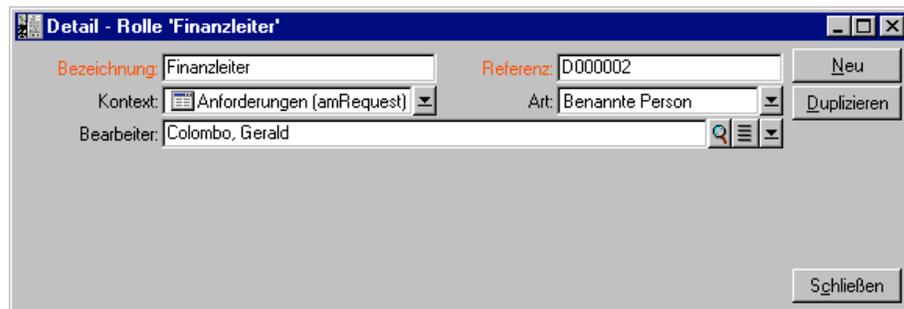
Leiter der Abteilung, in der der Anforderer tätig ist

Der Leiter wird mit Hilfe eines Skripts berechnet. Geben Sie zum Definieren des Leiters folgende Informationen im Detailbildschirm ein:



Finanzprüfer des Unternehmens

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 configuration window for a message. The 'Name' field contains 'Request approval reminder (functional)'. The 'Kontext' dropdown is set to 'Grundlegende Aufgaben einer Workflow-Instanz'. The 'Art' dropdown is set to 'Nachricht'. Below this, there are tabs for 'Nachrichtensystem', 'Beschreibung', and 'His'. The 'Referenzobjekt' dropdown is set to 'RequestDocRecord' and the 'Priorität' dropdown is set to 'Normal'. There is an unchecked checkbox for 'Empfangsbestätigung'. The 'An:' field contains '[Assignee]'. The 'Cc:', 'Bcc:', and 'Betreff:' fields are empty. The 'Text' field contains the following warning message: 'Warnung: Bewilligung für Anforderung 'RequestDocRecord.RegNumber' ausstehend ('RequestDocRecord.RegPurpose') ausgestellt von 'RequestDocRecord.Requester.Self'. Laufende Programme/Aufgaben verwenden.'

Aktion **Funktionelle Bewilligung**

Aufforderung zur finanziellen Bewilligung an den Finanzprüfer des Unternehmens

Die Aktion **Finanzielle Bewilligung** wird bei der finanziellen Bewilligung verwendet und gleicht der oben beschriebenen Aktion **Funktionelle Bewilligung**. Sie sendet eine Nachricht an den für die finanzielle Bewilligung zuständigen Mitarbeiter.

Bewilligung der Beschaffungsanforderung

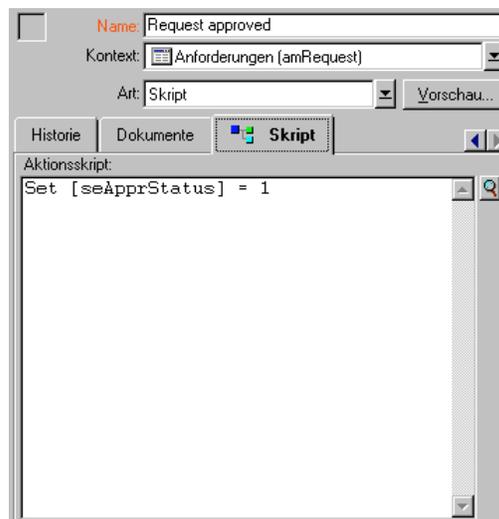
Diese Aktion kommt auf der Ebene der Aktivität **Anforderung bewilligt** zur Anwendung.

Über die Aktion wird der Wert im Feld **Bewilligungsstatus** (SQL-Name: seApprStatus) der Registerkarte **Bewilligung** im Anforderungsdetail in den Wert **Bewilligt** geändert. Dabei handelt es sich um eine Aktion vom Typ **Skript**.

Das Feld **Bewilligungsstatus** ist eine Systemaufzählung. Zum Zugreifen auf die verschiedenen Werte müssen Sie die Hilfe zu diesem Feld anzeigen:

- ❖ Positionieren Sie den Cursor im Detailfenster der Anforderung auf dem Feld **Bewilligungsstatus**.
- ❖ Drücken Sie die rechte Maustaste.
- ❖ Wählen Sie in dem daraufhin erscheinenden Kontextmenü die Option **Hilfe zu diesem Feld**: Der angezeigte Wert **Bewilligt** wird in der Datenbank als **1** gespeichert.

Die Aktion erscheint in folgendem Format:

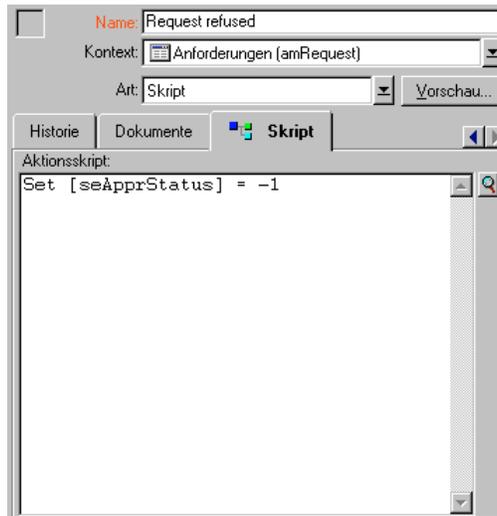


Aktion Anforderung bewilligt

Ablehnung der Beschaffungsanforderung

Diese Aktion kommt auf der Ebene der Aktivität **Anforderung abgelehnt** zur Anwendung.

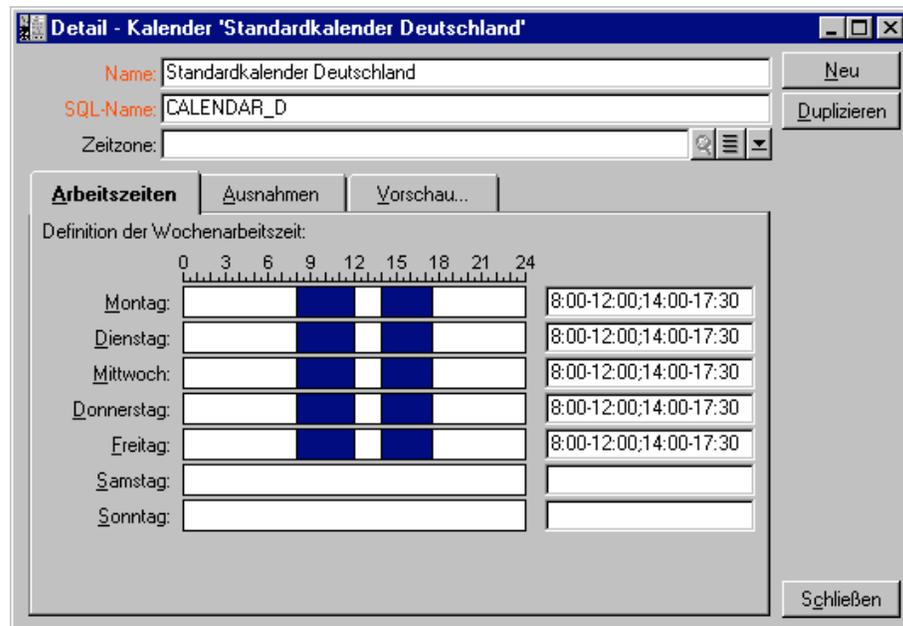
Diese Aktion ähnelt der Aktion **Anforderung bewilligt**, wobei jedoch im Feld **Bewilligungsstatus** (SQL-Name: seApprStatus) der Registerkarte **Bewilligung** des Anforderungsdetails der Wert **Abgelehnt** erscheint.



*Aktion **Anforderung abgelehnt***

Kalender erstellen

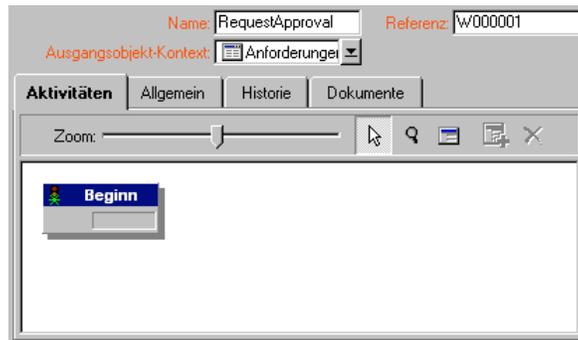
Der Zugriff auf die Liste mit den Kalendern erfolgt über das Menü **Extras/ Kalender**. Der Kalender wird den Aktivitäten des Workflow-Modells zugeordnet. Über diesen Kalender lassen sich die Fälligkeiten für das Workflow-Modell festhalten:



Kalender *Standardkalender Deutschland*

Workflow-Modell vorbereiten

- ↪ Wählen Sie das Menü **Extras/ Workflow/ Workflow-Modelle**.
- ↪ Klicken Sie auf .
- ↪ Geben Sie dem Modell einen Namen.
- ↪ 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: amRequest).
- ↪ Klicken Sie auf die Schaltfläche : Die Ausgangsaktivität (**Beginn**) wird von AssetCenter im Grafikeditor auf der Registerkarte **Aktivitäten** erstellt.



Das Workflow-Modell **Anforderungsbewilligung**

Aktivitäten erstellen

Die grafische Erstellung der Aktivitäten erfolgt auf der Registerkarte **Aktivitäten** im Detailfenster des Workflow-Modells:

- ↵ Drücken Sie die rechte Maustaste.
- ↵ Wählen Sie in dem daraufhin erscheinenden Kontextmenü die Option **Aktivität hinzufügen**, um den Detailbildschirm der Aktivität anzuzeigen.

Funktionelle Bewilligung

- ↵ Geben Sie der Aktivität einen Namen.
- ↵ 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).
- ↵ Das Feld **Kontext** (SQL-Name: ContextTable) auf der Registerkarte **Allgemein** bleibt unverändert.

Auszuführende Aufgabe

- ↵ Geben Sie die erforderlichen Informationen auf der Registerkarte **Fragen** ein, wie in der nachstehenden Abbildung zu sehen ist:

- ↳ Legen Sie den genauen Wortlaut der Frage fest:
 - ❖ Der Text der Frage bezieht sich auf die Nummer der Anforderung, die bewilligt werden soll.
 - ❖ Sie haben die Wahl zwischen 2 möglichen Antworten: Ablehnung oder Bewilligung. Wenn Sie eine Unterregisterkarte mit der Antwort auf die Frage hinzufügen möchten, positionieren Sie den Cursor auf der Ebene der Unterregisterkartennamen, und drücken die rechte Maustaste, um das Kontextmenü anzuzeigen. Wählen Sie die Option **Verknüpfung hinzufügen** oder **Verknüpfung duplizieren**.
- ↳ 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.
- ↳ Wenn der Bearbeiter automatisch über die Notwendigkeit der Bewilligung einer Anforderung informiert werden soll, gehen Sie wie folgt vor:
 - ❖ Wählen Sie rechts neben dem Feld **Person benachrichtigen** (SQL-Name: bNotifAssignee) den Wert **Ja**.

- ❖ Legen Sie die Aktion fest, die ausgeführt werden soll: In unserem Fall handelt es sich um die im vorherigen Schritt erstellte Aktion **Funktionelle Bewilligung**. Diese Aktion wird automatisch ausgelöst, sobald eine Anforderung zur funktionellen Bewilligung vorgelegt wird.

Fristen

Auf der Registerkarte **Fristen** des Aktivitätendetails geben Sie folgende Informationen ein:

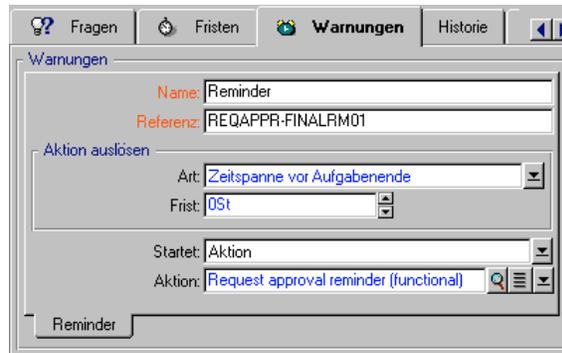
- Legen Sie den Kalender mit den gearbeiteten Tagen fest, der der Aktivität zugeordnet werden soll. Dieser Kalender wird bei der Berechnung der Workflow-Fälligkeiten berücksichtigt. Wählen Sie den Kalender **Standardkalender Deutschland**, den Sie im vorherigen Schritt erstellt haben.
- Definieren Sie die Frist, innerhalb derer eine Entscheidung gefallen sein muss, und beziehen Sie sich dabei auf den Zeitpunkt, zu dem die Aktivität ausgelöst wurde. In unserem Beispiel muss der Bearbeiter die Frage vor Ende des nächsten Tages beantwortet haben.

*Frist für die Aktivität **Funktionelle Bewilligung***

Warnung

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 unser Beispiel so einfach wie möglich zu gestalten, führt die Warnung zum Auslösen der Aktion **Funktionelle Bewilligung**:



Warnung **Bewilligung nicht ausgeführt**

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 mit den Ereignissen der Workflow-Instanz gespeichert (SQL-Name **amWfOccurEvent**).

AssetCenter Server übernimmt das Auslösen der darauf folgenden Aktivitäten (standardmäßig erscheint im Feld **Bearbeitung** (SQL-Name: `seProcessingMode`) auf der Registerkarte **Allgemein** des Ereignisdetails der Wert **Ereignis speichern und vom Server bearbeiten**.

Finanzielle Bewilligung

Diese Aktivität ähnelt der vorherigen.

Der Bearbeiter ist jedoch eine andere Person: Es handelt sich um den Finanzprüfer des Unternehmens, Herrn Gerald Colombo. Der Finanzprüfer wurde im vorherigen Schritt in der Tabelle der Workflow-Rollen erstellt. Wählen Sie die Aktion **Finanzielle Bewilligung**, um dem Bearbeiter eine entsprechende Nachricht zu schicken.

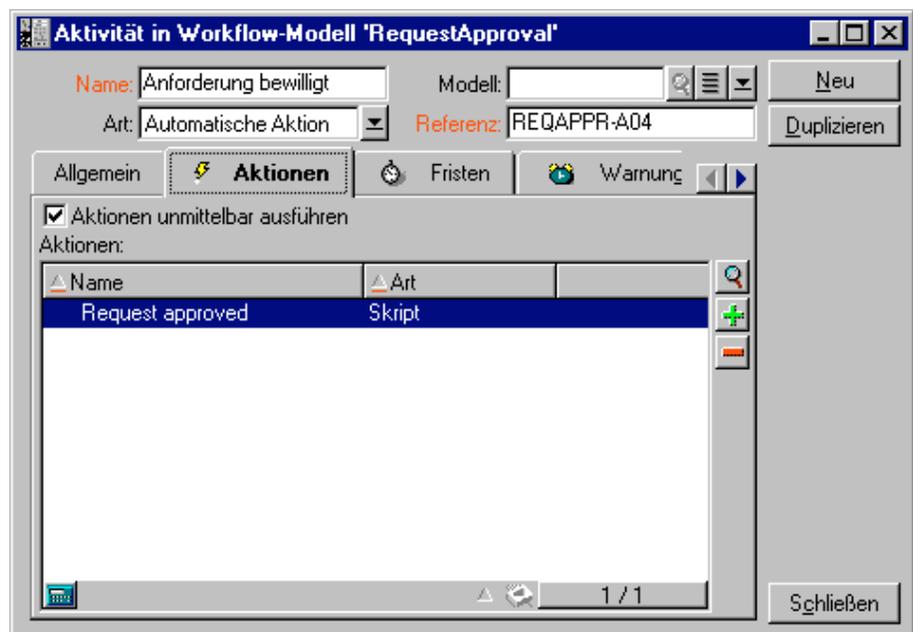
Bewilligte Anforderung

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 Aktion, die ausgeführt werden soll, ist die im vorigen Schritt erstellte Aktion **Anforderung bewilligt**.



Aktivität Anforderung bewilligt

Abgelehnte Anforderung

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 im vorigen Schritt erstellt wurde.

Ausgangsereignisse erstellen

Die Ereignisse, die eine Workflow-Instanz auslösen, sind der Aktivität **Beginn** zugeordnet.

Klicken Sie mit der rechten Maustaste auf die leere Ereigniszone in der Aktivität **Beginn** und wählen Sie in dem daraufhin erscheinenden Kontextmenü die Option **Ereignisdetail**, um das Ausgangsereignis zu definieren:

- In unserem Beispiel wird die Workflow-Instanz ausgelöst, sobald im Feld **Bewilligungsstatus** (SQL-Name: seApprStatus) der Registerkarte **Bewilligung** im Detailfenster einer Anforderung 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:

Parameter zum Auslösen eines Ausgangsereignisses

- Auf der Registerkarte **Allgemein** des Ereignisses wählen Sie im Feld **Bearbeitung** (SQL-Name: seProcessingMode) den Wert **Ereignis speichern und vom Server bearbeiten**.
- Auf der Registerkarte **Parameter** des Ereignisses:
 - ❖ Markieren Sie das Kontrollkästchen **Aktualisierung** (SQL-Name: bUpdate).

- ❖ Geben Sie im Feld **Kontrollierte Felder** (SQL-Name: MonitFields) die Informationen aus dem Feld mit dem SQL-Namen **seApprStatus** ein.

Übergänge erstellen

Sobald Sie die Aktivitäten definiert haben, müssen Sie sie nur noch durch Erstellen von Übergängen miteinander verknüpfen.

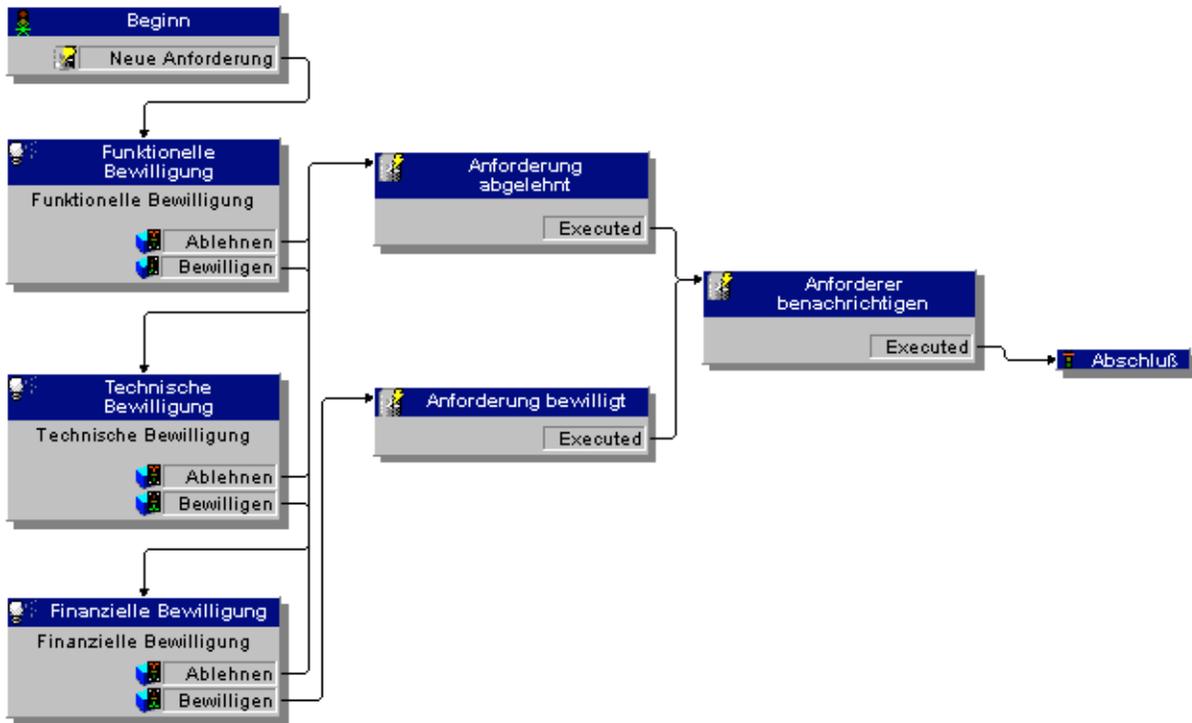
Gehen Sie zum Erstellen eines Übergangs wie folgt vor:

- Klicken Sie mit der linken Maustaste auf das Ausgangsereignis des Übergangs.
- 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:



Beispiele zum Auslösen des Workflows

Jetzt müssen Sie nur noch das einwandfreie Funktionieren des erstellten Workflow-Modells prüfen.

Befolgen Sie dabei die nachstehenden Anweisungen:

- ↪ Starten und konfigurieren Sie AssetCenter Server.
- ↪ Erstellen Sie die Techniker in der Tabelle der Abteilungen und Personen.
- ↪ Erstellen Sie eine Beschaffungsanforderung, die bewilligt werden muss.
- ↪ Lösen Sie eine Workflow-Instanz aus.
- ↪ Kontrollieren Sie den einwandfreien Ablauf der Instanz.

Start von AssetCenter Server

- ↪ Starten Sie AssetCenter Server.

- ↗ Stellen Sie eine Verbindung zur AssetCenter-Datenbank her.
- ↗ Wählen Sie das Menü **Aktionen/ Starten**. Wählen Sie in der Liste der auszuwählenden Module das Modul **Neue Workflow-Ausführungsgruppen suchen**.
- ↗ Klicken Sie auf OK: AssetCenter Server sucht nach neu erstellten Workflow-Gruppen und erstellt das für jede Gruppe entsprechende Überwachungsmodul.

Hinweis: Die Workflow-Modelle werden mit Hilfe der Workflow-Gruppen eingeteilt. Die Gruppe, zu der ein Workflow-Modell gehört, erscheint im Feld **Gruppe** (SQL-Name: GroupName) der Registerkarte **Allgemein** des Workflow-Details.

In unserem Beispiel haben wir keine Gruppe für das Workflow-Modell definiert. AssetCenter Server erstellt also das Überwachungsmodul **VWorkflow-Regeln für Modelle ohne Ausführungsgruppe ausführen...**

- ↗ Lassen Sie AssetCenter Server aktiviert.

Erstellung von Datensätzen in der Tabelle der Abteilungen und Personen

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:

- ↗ Erstellen Sie den Anforderer **Wolfgang Heinemann**, der in der Abteilung **EDV-Leitung** arbeitet.
- ↗ Ordnen Sie ihm ein **Login**, ein Kennwort und ein Benutzerprofil zu, das ihm die Eingabe einer Beschaffungsanforderung ermöglicht (Registerkarte **Profil** im Detailfenster des entsprechenden Datensatzes).
- ↗ Der Leiter der Abteilung **EDV-Leitung** heißt **Phillip Schumann**.
- ↗ Zum Vereinfachen des weiteren Verlaufs räumen Sie Phillip Schumann administrative Rechte an der Datenbank ein: Zeigen Sie hierzu die Registerkarte **Profil** im Detailfenster 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 Phillip Schumann ein.

Erstellung einer zu bewilligenden Beschaffungsanforderung

Im nun folgenden Schritt geht es um die Bewilligung einer Beschaffungsanforderung:

- ↪ Stellen Sie unter dem Login Wolfgang Heinemann eine Verbindung zur Beispieldatenbank her.
- ↪ Wählen Sie das Menü **Finanzen/ Anforderungen**.
- ↪ Klicken Sie auf .
- ↪ Wählen Sie im Feld **Anforderer** (SQL-Name: Requester) auf der Registerkarte **Allgemein** des Anforderungsdetails den Datensatz **Wolfgang Heinemann**.
- ↪ Wählen Sie den Wert **Zu bewilligen** für das Feld **Bewilligungsstatus** (SQL-Name: seApprStatus) auf der Registerkarte **Bewilligung** im Detailfenster der Beschaffungsanforderung.
- ↪ Bestätigen Sie die Erstellung der Beschaffungsanforderung: Das Ausgangsereignis für den Workflow ist eingetreten, und ein AssetCenter-Agent speichert das Ereignis in der Tabelle der grundlegenden Ereignisse der Workflow-Instanzen (SQL-Name **amWfOccurEvent**).

Auslösen des Workflows

- ↪ Verwenden Sie das Menü **Aktionen/ Starten** in AssetCenter Server. Wählen Sie in der Liste der auszulösenden Module das Modul **Workflow-Regeln für Modelle ohne Ausführungsgruppe ausführen...**
- ↪ Klicken Sie auf OK: AssetCenter Server erkennt, dass ein Ereignis eingetreten ist und aktiviert den Übergang dieses Ereignisses zur Aktivität **Funktionelle Bewilligung**. Daraufhin wird eine auszuführende Aufgabe erstellt, die dem für die Abteilung verantwortlichen Leiter des Anforderers zugeordnet wird.

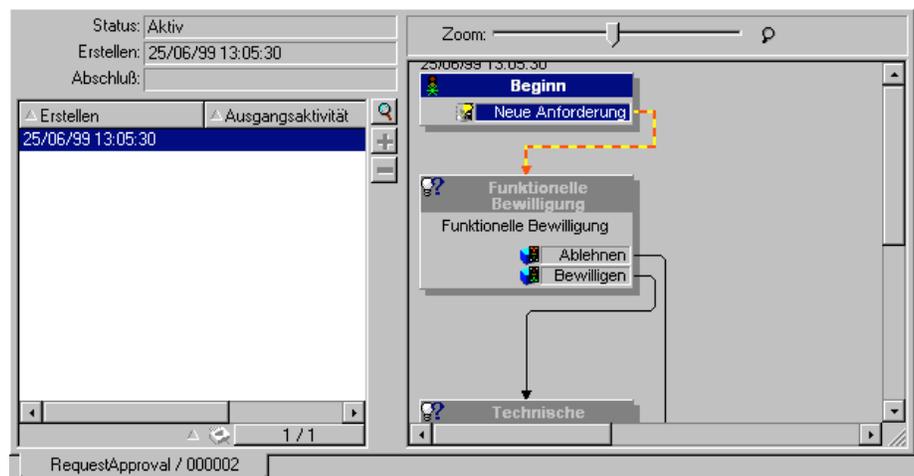
Kontrolle des einwandfreien Ablaufs des Workflows

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 Beispieldatenbank herstellen.

Anzeige der Workflow-Instanz

Zeigen Sie das Detailfenster der Beschaffungsanforderung an, die Sie zuvor erstellt haben: Auf der Registerkarte **Workflow** werden die aktuellen 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 der Unterregisterkarte erscheint der Status der Instanz. In unserem Beispiel blinkt die auszuführende Aufgabe **Funktionelle Bewilligung**.



Aktuelle Workflow-Instanz für die Beschaffungsanforderung

Anzeige der auszuführenden Aufgabe

- ❖ Wählen Sie das Menü **Extras/ Laufende Aufgaben**: Der Abteilungsleiter Phillip Schumann kann auf diese Weise das Detailfenster der Aufgabe anzeigen, die er ausführen soll:



Detail einer Aufgabe, die aus der Aktivität der funktionellen Bewilligung heraus entstanden ist

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 derer die Aufgabe ausgeführt werden muss. Dieses Datum wird anhand der Informationen berechnet, die auf der Registerkarte **Fristen** des Aktivitätendetails der funktionellen Bewilligung enthalten ist, und des Datums, an dem die Aufgabe erstellt wurde (Datum der Aktivierung des Übergangs durch AssetCenter Server).

Sie können auch auf die Schaltfläche  klicken, um das Detailfenster mit der Anforderung zu öffnen, die Gegenstand der Aufgabe ist.

- ❖ Klicken Sie auf die Schaltflächen  oder , um die Aufgabe auszuführen. Zusätzlich besteht die Möglichkeit, einen Kommentar zu der auf der Registerkarte **Allgemein** getroffenen Entscheidung hinzuzufügen.

Kontext eines Workflow-Modells

Jede Workflow-Instanz läuft in einem spezifischen Kontext ab.

Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

- Kontext eines Workflows definieren
- Referenzobjekt eines Workflow-Modells
- Aktuelle Workflow-Instanzen für ein Objekt einschränken

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 auf der Ebene des Ausgangsereignisses dar.

Beim Ablufen einer Workflow-Instanz macht das ursprüngliche Referenzobjekt eine Entwicklung im Vergleich zum ursprünglichen Referenzobjekt durch, und berücksichtigt dabei den Kontext, der auf der Ebene der 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 Ausgabeereignis 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 <i>Allgemein</i> des Workflows.	
		Bestätigt	Nicht bestätigt
Option Workflow neu starten, wenn bereits ein anderer Workflow läuft auf der Registerkarte <i>Allgemein</i> des Ausgabeereignisses der Aktivität Start.	Bestätigt	Wenn für das Objekt bereits ein aktueller Workflow vorliegt, wird er angehalten und ein neuer Workflow gestartet.	
	Nicht bestätigt	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** auf der Ebene des Ereignisses 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.

Hinweis: 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 die Tabelle mit den Workflow-Rollen 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 window titled "Detail - Rolle 'Finanzleiter'". It contains the following fields and controls:

- Bezeichnung:** Finanzleiter
- Referenz:** D000002
- Kontext:** Anforderungen (amRequest)
- Art:** Benannte Person
- Bearbeiter:** Colombo, Gerald

Buttons: "Neu", "Duplizieren", "Schließen".

Berechnete Person

Der Bearbeiter erscheint in der Tabelle der Abteilungen und Personen, wird jedoch mit Hilfe eines Skripts ermittelt.

Beispiel:

The screenshot shows a window titled "Detail - Rolle 'Abteilungsleiter des Anforderers'". It contains the following fields and controls:

- Bezeichnung:** Abteilungsleiter des Anforderers
- Referenz:** D000001
- Kontext:** Anforderungen (amRequest)
- Art:** Berechnete Person
- Skript:** RetVal=[Requester.Supervisor]

Buttons: "Neu", "Duplizieren", "Schließen".

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** (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 wie folgt definiert:

- 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:

- 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.

Hinweis: 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**.

- Es erscheint der Text mit der Frage bzw. den Anweisungen.
- 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 in den Unterregisterkarten mit der rechten Maustaste auf den Bereich mit den Namen, und wählen in dem daraufhin erscheinenden Kontextmenü die Option **Verknüpfung hinzufügen, Verknüpfung duplizieren** oder **Verknüpfung löschen**.

Hinweis: 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.

Hinweis: 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.

Hinweis: 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.

Hinweis: Als Ausgabeereignis einer Aktivität wird automatisch ein ausgeführtes Ereignis erstellt.

Ausführung

- Wenn Sie das Kontrollkästchen **Aktionen unmittelbar ausführen** markieren (SQL-Name: bExecImmediately), werden die Aktionen der Aktivität unmittelbar von der Workflow-Engine ausgeführt. Gleichzeitig aktiviert die Workflow-Engine den Übergang, der die Aktivität auslöst. Je nach dem Bearbeitungsmodus, den Sie für das Ereignis gewählt haben, das den Übergang auslöst, werden die Aktionen entweder von AssetCenter Server oder von einem AssetCenter -ausgeführt.
- Ansonsten werden die Aufgaben bei der nächsten Überwachung von AssetCenter Server durchgeführt.

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 Referenzprodukt 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 Produkt 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 in den Unterregisterkarten mit der rechten Maustaste auf den Bereich mit den Namen, und wählen in dem daraufhin erscheinenden Kontextmenü die Option **Verknüpfung hinzufügen**, **Verknüpfung duplizieren** oder **Verknüpfung löschen**.

Warnung: Der SQL-Name eines jeden Vorgangs muss mit dem Rückgabewert des Textskripts übereinstimmen.

Hinweis: Jeder Vorgang führt automatisch zur Erstellung eines Ausgabeereignisses der Aktivität.

Ausführung

- Wenn Sie das Kontrollkästchen **Aktionen unmittelbar ausführen** markieren (SQL-Name: bExecImmediately), werden die Aktionen der Aktivität unmittelbar von der Workflow-Engine ausgeführt. Gleichzeitig aktiviert die Workflow-Engine den Übergang, der die Aktivität auslöst. Je nach dem Bearbeitungsmodus, den Sie für das Ereignis gewählt haben, das den Übergang auslöst, werden die Aktionen entweder von AssetCenter Server oder von einem AssetCenter -ausgeführt.
- Ansonsten werden die Aufgaben bei der nächsten Überwachung von AssetCenter Server durchgeführt.

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.

Warnung: 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 Operator **AND** handelt, müssen alle Übergänge aktiviert werden, um die Aktivität auszulösen.
 - ❖ Wenn es sich bei der Eingabebedingung für die Aktivität um den Operator **OR** handelt, muss nur einer der Übergänge aktiviert werden, um die Aktivität auszulösen.

Hinweis: 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
- Aktivität vom Typ Automatische Aktion oder Test / Skript

- Aktuelle Aufgabenliste anzeigen
- Benutzeraufgaben ausführen
- Benutzeraufgaben zuordnen
- Workflow-Aufgaben verwalten

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 markiert 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 entweder um den AssetCenter Server oder um 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 auf der Ebene der 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:

- Positionieren Sie den Cursor auf einer Aktivität.
- Drücken Sie die rechte Maustaste.
- Wählen Sie in dem daraufhin erscheinenden Kontextmenü die Option **Laufende Aufgaben**.

Hinweis: 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 , um auf das Detail des Objekts zuzugreifen, auf das sich die Aufgabe bezieht.

Zum Ausführen einer Aufgabe öffnen Sie im Aufgabendetail die Registerkarte **Allgemein**:

- 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**

Im folgenden Abschnitt werden die Workflow-Ereignisse sowie die Art ihrer Bearbeitung beschrieben:

- Ereignisse vom Typ System
- Ereignisse vom Typ Warnung
- Ereignisse vom Typ Benutzer
- Allgemeine Bedingungen zur Aktivierung von Ereignissen
- Bearbeitung der Ereignisse
- Anwendung – Ausführung eines synchronen Workflows
- Abschließendes Ereignis

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 **Ausgeführt**, 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**.

Hinweis: 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 Datensätze 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:

The screenshot shows a software interface with four tabs: 'Allgemein', 'Parameter', 'Historie', and 'Dokumente'. The 'Parameter' tab is selected. Under the heading 'Zu kontrollierende Datensätze', there is a dropdown menu for 'Verknüpfung / Kontext:' and a text field for 'Kontrollierte Tabelle:' containing 'amRequest'. Below this, under 'Aktivierungsbedingungen für kontrollierte Datensätze', there are three checkboxes: 'Einflügen' (unchecked), 'Aktualisierung' (checked), and 'Löschen' (unchecked). Underneath, there is a text field for 'Kontrollierte Felder:' containing 'mTotalCost' and a search icon. At the bottom, there is a 'Skript:' field which is currently empty.

Registerkarte **Parameter** für ein Ereignis vom Typ **Bei Datenbankänderung**

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 der Überschrift "Bearbeitung der Ereignisse", auf der Seite 288 in diesem Handbuch.

Hinweis: Bei Ereignissen vom Typ **Bei Datenbankänderung** sind die Informationen, die auf der Registerkarte **Parameter** erscheinen, normalerweise zur Definition der zu überwachenden Datensätze völlig ausreichend. Die Eingabe von Daten im Feld **AQL-Bedingung** (SQL-Name: AQLCond) der Registerkarte **Allgemein** des Ereignisdetails ist nur selten erforderlich.

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 Kategorie **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 vom Planer im Detailfenster des Ereignisses und dort auf der Registerkarte **Parameter** definiert.

Eine Beschreibung der anschließenden Verarbeitung des Ereignisses finden Sie im Abschnitt "Bearbeitung der Ereignisse" auf Seite 288 in diesem Handbuch.

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)

Hinweis: 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:

		Option Ein laufender Workflow pro Objekt auf der Registerkarte <i>Allgemein</i> des Workflows.	
		Bestätigt	Nicht bestätigt
Option Workflow neu starten, wenn bereits ein anderer Workflow läuft auf der Registerkarte <i>Allgemein</i> des Ausgabeereignisses der Aktivität Start.	Bestätigt	Wenn für das Objekt bereits ein aktueller Workflow vorliegt, wird er angehalten und ein neuer Workflow gestartet.	
	Nicht bestätigt	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:

	<i>Ereignis speichern und vom Server bearbeiten</i>	Ereignis speichern und unmittelbar bearbeiten	<i>Ereignis unmittelbar bearbeiten, nicht speichern</i>
Ereignis vom Typ In regelmäßigen Abständen	<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 Parameter des Ereignisdetails.</p> <p>Sobald das Ereignis eintritt, speichert AssetCenter Server das Ereignis in der Tabelle mit dem SQL-Namen amWfOccurEvent.</p> <p>Der Übergang wird anschließend von AssetCenter Server aktiviert. Die Definition der Häufigkeit, mit der AssetCenter Server die zu aktivierenden Übergänge überwacht, 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 Parameter des Ereignisdetails.</p> <p>Sobald das Ereignis eintritt, speichert AssetCenter Server das Ereignis in der Tabelle mit dem SQL-Namen amWfOccurEvent.</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 Parameter des Ereignisdetails.</p> <p>Wenn das Ereignis eintritt, wird es nicht in der Tabelle mit dem SQL-Namen amWfOccurEvent gespeichert. Der Übergang wird jedoch unmittelbar von AssetCenter Server ausgeführt.</p>
Ereignis vom Typ Bei Datenbankänderung oder Systemereignis, das von AssetCenter ausgelöst wird (Ergebnis einer	<p>Sobald ein Ereignis eintritt, wird es von der AssetCenter-Client-Station in der Tabelle mit dem SQL-Namen amWfOccurEvent gespeichert.</p> <p>Der Übergang wird später von AssetCenter Server aktiviert. Die</p>	<p>Sobald ein Ereignis eintritt, wird es von der AssetCenter-Client-Station in der Tabelle mit dem SQL-Namen amWfOccurEvent gespeichert.</p> <p>Der Übergang wird unmittelbar von der AssetCenter-Client-</p>	<p>Wenn ein Ereignis eintritt, wird es nicht in der Tabelle mit dem SQL-Namen amWfOccurEvent gespeichert. Der Übergang wird jedoch unmittelbar von der AssetCenter-Client-Station aktiviert.</p>

Aktivität vom Typ Frage oder Benutzeraktion, Ergebnis einer Aktivität vom Typ Automatische Aktion oder Test / Skript, die von AssetCenter ausgeführt wird	Definition der Häufigkeit, mit der AssetCenter Server die zu aktivierenden Übergänge überwacht, erfolgt auf der Ebene der Optionen von AssetCenter Server.	Station aktiviert.	
Von AssetCenter Server ausgelöstes Systemereignis (Ergebnis einer Aktivität vom Typ Test / Skript oder Automatische Aktion ausgeführt von AssetCenter Server) oder Ereignis mit einer Warnung	<p>Sobald ein Ereignis eintritt, wird es von AssetCenter Server in der Tabelle mit dem SQL-Namen amWfOccurEvent gespeichert.</p> <p>Der Übergang wird später von AssetCenter Server aktiviert. Die Definition der Häufigkeit, mit der AssetCenter Server die zu aktivierenden Übergänge überwacht, erfolgt auf der Ebene der Optionen von AssetCenter Server).</p>	<p>Sobald ein Ereignis eintritt, wird es von AssetCenter Server in der Tabelle mit dem SQL-Namen amWfOccurEvent 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 amWfOccurEvent gespeichert. Der Übergang wird jedoch unmittelbar von AssetCenter Server aktiviert.</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

Sie können synchrone, asynchrone und gemischte Workflow-Modelle ausführen.

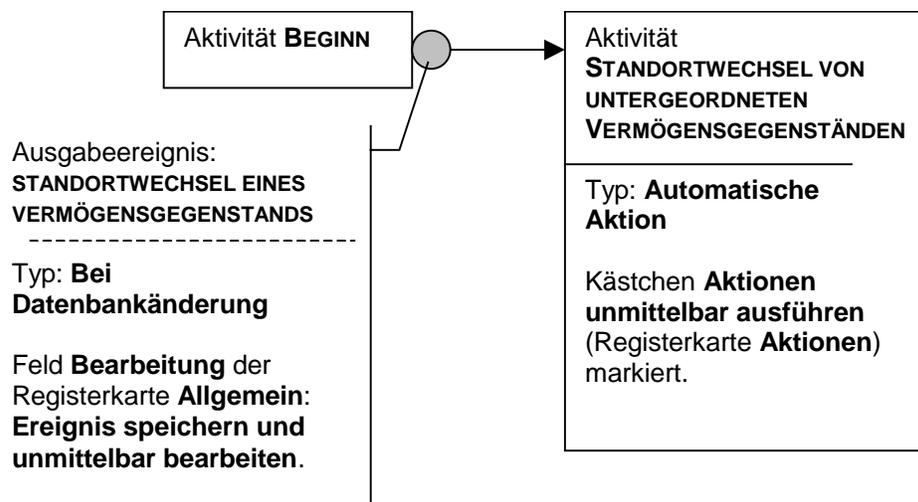
Anwendung – Ausführung eines synchronen Workflows

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:



Beispiel eines synchronen Workflows

Wenn der Standort eines Vermögensgegenstands geändert wird und Sie auf die Schaltfläche klicken, geschieht Folgendes:

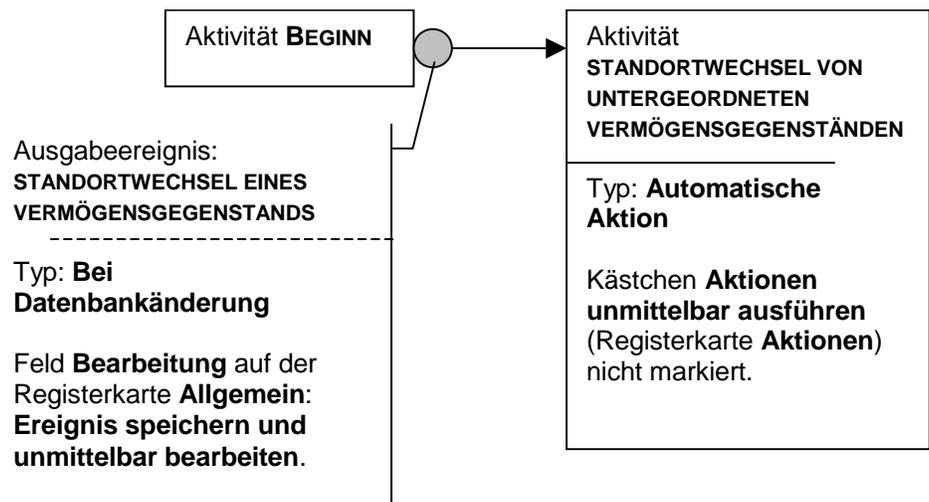
- ↪ Ein Ereignis vom Typ **Bei Datenbankänderung** tritt ein.
- ↪ Der Standort des Vermögensgegenstands wird geändert.
- ↪ Der Workflow wird gestartet.
- ↪ Der Übergang des Workflows wird aktiviert.

- ↳ Der Standort der untergeordneten Vermögensgegenstände wird geändert.
- ↳ 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.



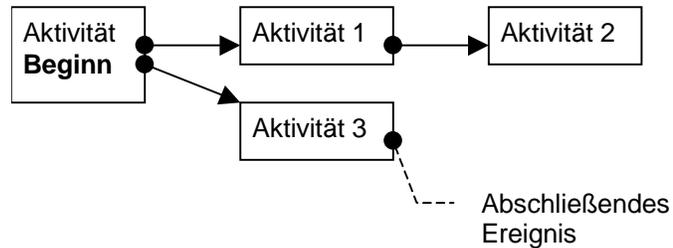
Beispiel eines asynchronen Workflows

Abschließendes Ereignis

Definition

Ein abschließendes Ereignis beendet eine Workflow-Instanz, auch wenn noch weitere Aufgaben im Rahmen der Instanz ausgeführt werden müssen.

Beispiel:



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:

- Wählen Sie das Ereignis.
- Drücken Sie die rechte Maustaste, um das Kontextmenü anzuzeigen.
- 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.

Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

- Fristen
- Warnungen

Fristen

Die Definition der Frist, innerhalb der eine Workflow-Aktivität ausgeführt worden sein muss, erfolgt auf der Registerkarte **Fristen** des Aktivitätendetails.

Hinweis: 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.

Hinweis: 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 vor Aufgabenbeginn**)
- Über eine Frist, die vor einem in der Datenbank gespeicherten Datum liegt (Art **Zeitspanne vor Aufgabenende**)
- Über den Prozentsatz einer Frist, vor deren Ablauf die Aktivität ausgeführt worden sein muss (Feld **Frist** (SQL-Name: tsResolDelay) der Registerkarte **Fristen**)

Hinweis: 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 Definition der Häufigkeit, mit der die Überwachung durchgeführt wird, erfolgt in den Optionen von AssetCenter Server.

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 kontrolliert die Erstellung der neuen Workflow-Ausführungsgruppen.

Sobald AssetCenter Server eine neue Workflow-Ausführungsgruppe G erkennt, wird ein neues Kontrollmodul **Workflow-Regeln für die Ausführungsgruppe G ausführen** erstellt.

Dieser Mechanismus bringt nachstehende Vorteile mit sich:

- Er ermöglicht die Definition von Überwachungszeiten für einzelne Workflow-Ausführungsgruppen.
- Die einzelnen Workflow-Ausführungsgruppen können von verschiedenen AssetCenter Server-Programmen überwacht werden.

Sobald eine Workflow-Ausführungsgruppe erkannt wird, überwacht 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.

Kapitel 15 - Import von Daten

In diesem Kapitel wird der Import von Daten mit AssetCenter erläutert.

Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

- Funktionsprinzip des Datenimports
- Empfehlungen
- Auflagen, die beim Import von Daten in ein Feld zu beachten sind
- Textdateien oder ODBC-Datenbank importieren
- AssetCenter-Datenbank importieren
- Importskripte speichern und ausführen
- Importskript über eine 32-Bit-Eingabeaufforderung ausführen

Der Import einer einzelnen Textdatei erfolgt über das Menü Datei/ Import, Option Textdatei importieren.

Der Import mehrerer Textdateien oder einer Datenbank erfolgt über das Menü Datei/ Import, Option Datenbank importieren.



Auswahl eines Importtyps

Funktionsprinzip des Datenimports

AssetCenter ermöglicht einem Administrator, Daten auf verschiedene Weisen in die AssetCenter-Datenbank zu importieren:

Mit Hilfe einzelner Textdateien

Die Textdatei ist einer Haupttabelle in der AssetCenter-Datenbank zugeordnet.

Jedes Feld der Textdatei ist einem Feld in der Datenbank zugeordnet. Dieses Feld gehört zu einer Haupttabelle oder einer verknüpften Tabelle.

Mit Hilfe mehrerer Textdateien

Jede Textdatei ist einer Tabelle der AssetCenter-Datenbank zugeordnet.

In jeder Textdatei sind die einzelnen Felder einem Datenbankfeld zugeordnet. Die Felder in der Datenbank gehören zu einer Haupttabelle oder einer verknüpften Tabelle.

Mit Hilfe einer vollständigen Datenbank

Sie haben die Möglichkeit, eine vollständige Datenbank zu importieren:

- AssetCenter
- ODBC für andere Datenbanken

Jede Tabelle der Ausgangsdatenbank ist einer Tabelle der Zieldatenbank zugeordnet.

Jedes Feld der Ausgangstabellen ist einem Feld der Zieldatenbank zugeordnet. Jedes Feld gehört zu einer zugeordneten Tabelle oder einem verknüpften Feld.

Sie können Datensätze in der AssetCenter-Datenbank hinzufügen oder ändern. Das Löschen eines Datensatzes aus der AssetCenter-Datenbank ist jedoch nicht möglich.

Sämtliche, die Datenübertragung betreffenden Informationen können in Form eines Skripts gespeichert werden. Damit ist der erneute Import von Daten zu einem späteren Zeitpunkt möglich, ohne dass Sie die Importparameter dafür ein zweites Mal definieren müssten.

Weiterhin lässt Sie das Importmodul die Verwaltung von Fehlermeldungen einrichten und bietet Ihnen die Möglichkeit, das Ergebnis der ausgeführten Vorgänge in einer Protokolldatei festzuhalten.

Hinweis zum Schutz der Datenbank: Der Zugriff auf das Menü **Datei/Import** ist einem AssetCenter-Administrator vorbehalten (Login **Admin** oder Benutzer mit administrativen Rechten). Für alle anderen Benutzer erscheint das Menü grau unterlegt. Der Zugriff auf die Datenbank ist damit geschützt.

Hinweis zum Import von numerischen Daten: Jedes numerische Feld muss unter Berücksichtigung eines anerkannten Standards formatiert werden. Die in der Windows-Systemsteuerung definierten Optionen, wie z. B. Ziffern, fehlende Leerstellen oder als Trennzeichen für Dezimalstellen definierte Punkte spielen dabei keine Rolle. Die numerischen Felder müssen auch als numerische Felder importiert werden. Das hat den Vorteil, dass Zahlen unabhängig von den in der Windows-Systemsteuerung gewählten Optionen und unabhängig von der Arbeitsstation importiert werden können, die den Import ausführt.

Hinweis zum Import von Bildern, Formularen, Abfragen, Berichten, Zugriffsberechtigungen, Ansichten, Kennwörtern und Logins: Der Import dieser Elemente ist nur beim Import einer AssetCenter-Datenbank möglich.

Empfehlungen

Im Folgenden finden Sie einige Empfehlungen, die Sie beim Import von Daten in die AssetCenter-Datenbank beachten sollten:

Standardwerte für obligatorische Felder

Das Importmodul trägt dem obligatorischen Charakter, den Sie gegebenenfalls für bestimmte Felder definiert haben, keine Rechnung. Aus diesem Grund ist es empfehlenswert, den obligatorischen Feldern einen Standardwert zuzuordnen, damit sie beim Import nicht leer

bleiben. Der in den importierten Dateien enthaltene Wert hat Vorrang vor dem für das Feld geltenden Standardwert.

Verwendung der Identifizierungsschlüssel als Abstimmungsschlüssel vermeiden

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.

AssetCenter-Datenbank speichern

Beim Import von Daten kann es in den Daten, die bereits in der AssetCenter-Datenbank gespeichert sind, zu tiefgreifenden Änderungen kommen. Sie sollten daher vor dem Start eines Imports eine Sicherheitskopie anlegen.

Gleichzeitigen Zugriff mehrerer Benutzer auf die AssetCenter-Datenbank vermeiden

Wir raten Ihnen dringend davon ab, mehrere Importe gleichzeitig auf verschiedenen Rechnern durchzuführen oder AssetCenter während des Imports auf einem anderen Rechner zu verwenden.

Auflagen, die beim Import von Daten in ein Feld zu beachten sind

Beim Import von Daten in die AssetCenter-Datenbank prüft das Importmodul, ob die importierten Daten mit der Struktur der Datenbank kompatibel sind. Sollten bestimmte Felder oder Verknüpfungen der importierten Daten nicht mit der Struktur der Datenbank kompatibel sein, kann es zu folgenden Ergebnissen kommen:

- Der Datensatz wird vollständig abgelehnt
- Der Wert ist unvollständig
- Das Feld bleibt unverändert erhalten

Wir empfehlen Ihnen daher, auf die Kompatibilität der Daten mit der Struktur der Datenbank zu achten, um Zurückweisungen zu vermeiden.

Dabei sind zwei Situationen zu unterscheiden:

- Sie importieren die Daten einer strukturierten Datenbank:
 - ❖ Wenn das Ausgangs- und das Zielfeld dem gleichen Typ entsprechen (z. B. Ausgangsdatum und Zieldatum), und wenn die Felder kompatibel sind (z. B. Ausgangsdatum + -uhrzeit und Zieldatum), müssen Sie keine Auflagen beachten.
 - ❖ Wenn das Ausgangsfeld dem Typ **Text** entspricht und der Typ des Zielfelds präzise definiert ist, müssen Sie die Auflagen des Zielfelds beachten.
- Sie importieren die Daten einer Textdatei:
 - ❖ Sie müssen die Auflagen des Zielfelds berücksichtigen.

Wert der Eigenschaft *UserType* im Zielfeld

Der Wert entspricht:	Auflagen:
Default	Sie müssen das von der Eigenschaft definierte Format Type respektieren.
Number oder Money	<p>Im Zielfeld muss eine Zahl erscheinen.</p> <p>Wenn das Zielfeld aus einer strukturierten Datenbank stammt und das Feld dem Typ Number oder Money entspricht, gibt es keine Auflagen.</p> <p>Wenn das Zielfeld dem Typ Text entspricht, müssen Sie eine Zahl eingeben. Verwenden Sie einen . zum Trennen von Dezimalstellen und geben Sie keine Zeichen zum Trennen von Tausendern ein.</p>
Yes/No	Im Zielfeld muss der Wert 1 (für Ja), bzw. 0 (für Nein) erscheinen.
Date	<p>Wenn das Zielfeld aus einer strukturierten Datenbank stammt und das Feld dem Typ Datum oder Datum+Uhrzeit entspricht, gibt es keine Auflage.</p> <p>Wenn das Zielfeld dem Typ Text entspricht, müssen Sie folgende Auflagen beachten:</p> <ul style="list-style-type: none"> ❖ Die Reihenfolge bei der Eingabe von Monat, Tag und Jahr muss für alle Datensätze identisch sein. Die Reihenfolge wird zum Zeitpunkt des Imports definiert. ❖ Sie müssen grundsätzlich einen Tag, einen Monat und ein Jahr eingeben. ❖ Zum Trennen von Tagen, Monaten und Jahren müssen Sie in allen Datensätzen dasselbe Trennzeichen verwenden. Die Definition des Trennzeichens erfolgt zum Zeitpunkt des Imports.
System itemized list	<p>Sie müssen einen der Werte eingeben, und zwar unter Ausschluß aller anderen Werte. Ansonsten wird die Zeile nicht importiert. Sie können das Element der Aufzählung anhand des angezeigten Werts oder der in der Datenbank gespeicherten Zahl identifizieren.</p> <p>Ein Beispiel: Die Aufzählung lautet Ja 1 Nein 0. Sie können den Wert Ja oder den Wert 1 importieren und erhalten das gleiche Ergebnis.</p> <p>Wenn Sie das Feld leer lassen, ordnet das Importmodul dem Feld den Wert 0 zu.</p> <p>Wir empfehlen Ihnen die Eingabe der gespeicherten numerischen Werte, da sie von einer AssetCenter-Version zur nächsten eine größere Stabilität aufweisen</p>

Der Wert entspricht:	Auflagen:
	als Felder vom Typ Text und außerdem von den AssetCenter-Sprachen unabhängig sind.
Custom itemized list	Der Wert für das Feld wird in einer vom Benutzer anpassbaren Aufzählung ausgewählt. Sie können dann im Importmodul einen der Werte der Aufzählung eingeben. Wenn es sich um eine offene Aufzählung handelt, können Sie einen beliebigen Wert eingeben, der dann in der Aufzählung hinzugefügt wird.
Percentage	Bei den importierten Werten muss es sich um den Prozentsatz mit oder ohne das Symbol % handeln (beispielsweise: 10 oder 10%).
Time span	Wenn es sich bei der Eigenschaft UserType um den Typ Time span handelt, müssen Sie die für die Eigenschaften UserType und UserTypeFormat definierten Auflagen berücksichtigen.
Table or field SQL name	Sie haben die Möglichkeit, jeden beliebigen alphanumerischen Wert zu importieren. Wenn dieser Wert jedoch nicht dem SQL-Namen eines Felds oder einer gültigen Tabelle entspricht, laufen Sie Gefahr, die Datenbank zu beschädigen.

Wert der Eigenschaft **Type** im Zielfeld

Wenn die Eigenschaft **UserType** den Wert **Default** aufweist, müssen Sie die Auflagen respektieren, die mit dieser Eigenschaft verknüpft sind.

Sonderfall:

Der Wert entspricht:	Auflagen:
Date+Time	Wenn das Zielfeld aus einer strukturierten Datenbank stammt, und der Typ des Felds Datum+Uhrzeit oder Datum entspricht, gibt es keine Auflagen. Wenn das Zielfeld dem Typ Text entspricht, müssen Sie die Auflagen respektieren, die über die Eigenschaft UserType definiert werden, wenn die Eigenschaft UserType dem Typ Datum oder Date+Time entspricht.

Wert der anderen Eigenschaften des Zielfelds

Für die Eigenschaft:	Erscheint der Wert:	Auflagen:
MandatoryType	Yes	Wenn das Zielfeld leer ist, werden die Datensätze, die vom Importmodul normalerweise hinzugefügt bzw. geändert werden, weder hinzugefügt noch geändert.
Size	Ausgefüllt	Die Werte überlanger Zielfelder werden beim Import abgeschnitten.
ReadOnly	Yes	Der Import eines Werts in ein Feld mit dieser Eigenschaft ist nicht möglich.

Feld *dtLastModif* (SQL-Name)

Dieses Feld wird aktualisiert, sobald ein Datensatz in der AssetCenter-Datenbank erstellt oder geändert wird. Dabei spielt es keine Rolle, ob die Erstellung bzw. Änderung des Datensatzes über die Benutzeroberfläche oder den Datenimport durchgeführt wurde. Der Wert verweist auf das Datum, an dem die Erstellung bzw. Änderung des Datensatzes stattgefunden hat.

Der in dieses Feld importierte Wert hat Vorrang vor dem tatsächlichen Importdatum.

Textdateien oder ODBC-Datenbank importieren

In diesem Abschnitt wird das Importieren einer oder mehrerer Textdateien bzw. einer ODBC-Datenbank in die AssetCenter-Datenbank beschrieben:

Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

- Vorbereitung des Imports von Textdateien
- Schritt 1 - Auswahl der zu importierenden Textdatei bzw. der ODBC-Datenbank
- Schritt 2 - Beschreibung des Parsings der zu importierenden Dateien bzw. Tabellen
- Schritt 3 - Beschreibung der zu importierenden Felder
- Schritt 4 - Zuordnung der zu importierenden Felder zu den Feldern der AssetCenter-Datenbank
- Schritt 4 - Zuordnung der einzelnen Textdateien oder Ausgangstabellen zu einer Zieltabelle
- Schritt 4 - Zuordnung der zu importierenden Felder zu den Feldern der AssetCenter-Datenbank
- Schritt 4 - Zuordnung zusätzlicher berechneter Felder zur Ausgangsdatei
- Schritt 4 - Sonderfälle des Imports
- Schritt 4 - Anwendungsbeispiele für Identifizierungsschlüssel
- Schritt 5 - Parameter für die Übertragung der Daten
- Schritt 6 - Übertragung der Daten

Der Import einer einzelnen Textdatei erfolgt über das Menü Datei/Import, Option Textdatei importieren.

Der Import mehrerer Textdateien erfolgt über das Menü Datei/Import, Option Datenbank importieren, Registerkarte Text.

Der Import einer ODBC-Datenbank erfolgt über das Menü Datei/Import, Option Datenbank importieren, Registerkarte ODBC.

Vorbereitung des Imports von Textdateien

Beginnen Sie mit der Vorbereitung der Dateien, in denen die zu importierenden Daten gespeichert sind. Diese Dateien müssen den nachstehenden Bedingungen entsprechen:

- Jede Spalte muss einem Feld entsprechen.
- Jede Zeile muss einem Datensatz entsprechen.
- Zu Anfang der Datei können Sie gegebenenfalls die Namen der Felder anzeigen. Wenn Sie die Namen der Felder nicht direkt in der Datei eingeben, können Sie dies zum Zeitpunkt des Imports nachholen.

- Am Anfang der Datei und noch vor der Zeile, die gegebenenfalls die Feldnamen enthält, können Sie bis zu 100 Zeilen Kommentar einfügen, die beim Import nicht berücksichtigt werden. Diesen Kommentarzeilen muss kein besonderes Zeichen vorangestellt werden, da Sie die Anzahl dieser Zeilen beim Import in die AssetCenter-Datenbank angeben.
- Bei der Erstellung der Datei können Sie entweder den Zeichensatz OEM DOS (DOS) oder ANSI (Windows) verwenden.
- Die Spalten können entweder eine feste oder eine variable Breite aufweisen und mit einem beliebigen Trennzeichen voneinander abgegrenzt werden.
- Die Abgrenzung des Inhalts der Felder kann mit einem Zeichen Ihrer Wahl erfolgen.
- Die Felder müssen dem Typ Numerisch, Zeichen oder Datum entsprechen.
- Die importierten Werte müssen die für die AssetCenter-Datenbank geltenden Bedingungen respektieren. Dazu gehören beispielsweise das Eingabeformat, der Feldtyp, der Verknüpfungstyp und der Indextyp.
- Sie können alle Zeichen mit Ausnahme der Zeichen verwenden, die zum Abgrenzen von Text dienen. Das Einfügen eines Trennzeichens in einer Textkette ist nicht möglich.
- Für die Informationen vom Typ **Datum**, **Datum + Uhrzeit** und **Dauer** gelten dieselben Bedingungen wie bei der direkten Eingabe in die Software.
- Wir empfehlen Ihnen, für jede Zielhaupttabelle eine Textdatei zu erstellen.

Wenn die ursprünglichen Daten in einer Datenbank enthalten sind, deren Engine vom Importmodul nicht verwaltet werden kann, müssen Sie die Daten zunächst als Textdatei extrahieren, um Sie anschließend importieren zu können. Wenn die Datenbank jedoch von AssetCenter erkannt wird, kann der Import der Datenbank direkt über das Menü **Datei/ Import**, Option **Datenbank importieren** erfolgen.

Schritt 1 - Auswahl der zu importierenden Textdatei bzw. der ODBC-Datenbank

Import einer einzelnen Textdatei

↪ Verwenden Sie das Menü **Datei/ Import**.

- ↪ Wählen Sie die Option **Textdatei importieren**.
- ↪ AssetCenter fordert Sie auf, die Position der strukturierten Textdatei anzugeben, die die zu importierenden Daten enthält.

Import von mehreren Textdateien

Gehen Sie zum Auswählen der Textdateien, die Sie zuvor unter Berücksichtigung der vorgegebenen Bedingungen erstellt haben, wie folgt vor:

- ↪ Verwenden Sie das Menü **Datei/ Import**, Option **Datenbank importieren**, Registerkarte **Text**.
- ↪ Klicken Sie auf die Schaltfläche .
- ↪ Wenn Sie sich im Importmodul befinden, verwenden Sie das Menü **Datei/ Datei hinzufügen**. Geben Sie nacheinander die Namen der Textdateien an, die Sie hinzufügen möchten.

Import einer ODBC-Datenbank

Die Auswahl der Ausgangs-Datenbank erfolgt über das Menü **Datei/ Import**, Option **Datenbank importieren**.

AssetCenter ermöglicht den Import einer ODBC-Datenbank über die Registerkarte **ODBC**.

- ↪ Identifizieren Sie die Datenquelle, den Benutzer und das Kennwort.
Das Symbol  ermöglicht die direkte Erstellung einer ODBC-Quelle, ohne dass Sie dazu die Windows Systemsteuerung und dann die ODBC-Konfiguration aufrufen müssten.
- ↪ Klicken Sie auf die Schaltfläche .

Schritt 2 - Beschreibung des Parsings der zu importierenden Dateien bzw. Tabellen

Warnung: Wenn die Datei aus Feldern gleicher Breite besteht, wird bei diesem Schritt nur ein einziger Bildschirm angezeigt. Besteht die Datei jedoch aus Feldern, die durch ein Trennzeichen voneinander abgegrenzt sind, besteht dieser Schritt aus zwei Bildschirmen.

Import einer einzelnen Textdatei

Sobald Sie, wie unter Schritt 1 beschrieben, eine Textdatei ausgewählt haben, werden Sie von AssetCenter automatisch zur Beschreibung des Parsings aufgefordert.

Import von mehreren Textdateien

Sobald Sie eine Datei, wie unter Schritt 1 beschrieben, hinzugefügt haben, werden Sie von AssetCenter automatisch zur Beschreibung des Parsings aufgefordert. Sie haben die Wahl, diese Beschreibung sofort einzugeben oder durch Klicken auf die Schaltfläche , auf einen späteren Zeitpunkt zu verschieben.

Wenn Sie sich bereits auf dem Hauptbildschirm des Importmoduls befinden und den Bildschirm zur Beschreibung der Importdatei erneut aufrufen möchten, gehen Sie wie folgt vor:

- ❖ Doppelklicken Sie auf die Ausgangsdatei, oder
- ❖ Verwenden Sie nach der Auswahl der Ausgangsdatei das Menü **Bearbeiten/ Eigenschaften**.

Die Liste der Textdateien befindet sich in der Spalte **Ausgangstabellen** auf dem Hauptbildschirm des Importmoduls.

Import einer ODBC-Datenbank

Sobald Sie die Datenbank wie unter Schritt 1 beschrieben geöffnet haben, können Sie die Beschreibung der Tabellen über den Hauptbildschirm des Importmoduls einsehen:

- ❖ Doppelklicken Sie dazu auf die Ausgangstabelle, oder
- ❖ Verwenden Sie nach der Auswahl der Ausgangstabelle das Menü **Bearbeiten/ Eigenschaften**.

Die Liste der Textdateien befindet sich in der Spalte **Ausgangstabellen** auf dem Hauptbildschirm des Importmoduls.

Erster Bildschirm

Zeichen codieren

Geben Sie in diesem Feld an, ob der Text im Format ANSI (Windows) oder ASCII (DOS) vorliegt.

Erste Importzeile

Geben Sie hier die Nummer der Zeile an, die den Anfang der zu importierenden Daten kennzeichnet. Die Zeilen vor dieser Nummer werden vom Importmodul nicht berücksichtigt.

Wenn Ihr Dokument eine Zeile mit dem Namen der Felder enthält und sich diese Zeile direkt vor der ersten, zu importierenden Zeile befindet, geben Sie in diesem Feld die Nummer der Zeile mit den Feldnamen an.

Mit AssetCenter haben Sie die Möglichkeit, bis zu 100 Zeilen am Anfang der Datei unberücksichtigt zu lassen.

Erste Importzeile mit Spaltennamen

Wenn die erste zu importierende Zeile den Namen der Felder und damit der Spalten enthält, sollten Sie dieses Kontrollkästchen markieren. Sie ersparen sich damit die Zuordnung von Spaltennamen zu einem späteren Zeitpunkt.

Enthält Ihre Datei jedoch keine Feldnamen, ist ein Markieren dieses Kästchens nicht erforderlich.

Trennzeichen

Markieren Sie dieses Kontrollkästchen, wenn die Werte der Felder durch Trennzeichen voneinander abgegrenzt sind.

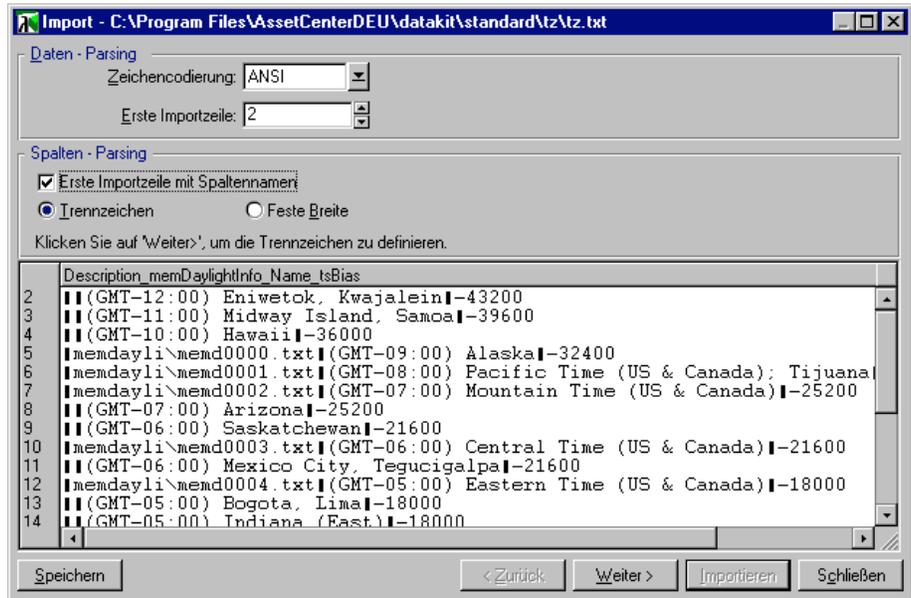
Die Definition des Trennzeichens erfolgt auf dem nachstehend beschriebenen Importbildschirm.

Feste Breite

Markieren Sie dieses Kontrollkästchen, wenn alle Werte eines bestimmten Felds die gleiche Breite aufweisen und dies in sämtlichen Feldern.

AssetCenter bietet automatisch senkrechte Trennstriche an.

- ❖ Wenn Sie einen Trennstrich verschieben möchten, wählen Sie den Strich mit Hilfe der Maus im Bereich der Datensätze und ziehen ihn auf die gewünschte Position. Das Markieren des Trennstrichs im Titelbereich ist jedoch nicht möglich.
- ❖ Um einen Trennstrich zu löschen, wählen Sie den Strich mit der Maus und ziehen ihn außerhalb der Tabelle.
- ❖ Wenn Sie einen Trennstrich hinzufügen wollen, klicken Sie mit der Maus im Bereich mit den Datensätzen auf die Position, an der Sie den Strich einfügen wollen.



Beschreibung des Parsings von Textdateien (Bildschirm 1)

Hinweis: In der unteren Hälfte des Bildschirms erscheint eine Vorschau der zu importierenden Daten mit maximal 25 Zeilen.

Eventuell erscheinender zweiter Bildschirm

Der zweite Bildschirm erscheint unter folgenden Bedingungen:

- ↪ Sie legen fest, dass die Werte der Felder durch ein Trennzeichen voneinander abgegrenzt sind.
- ↪ Sie klicken auf .

Spaltentrennzeichen

Geben Sie das Zeichen ein, mit dem die Werte zweier aufeinander folgender Felder getrennt werden sollen. Wir empfehlen, das Zeichen ; als Trennzeichen zu verwenden.

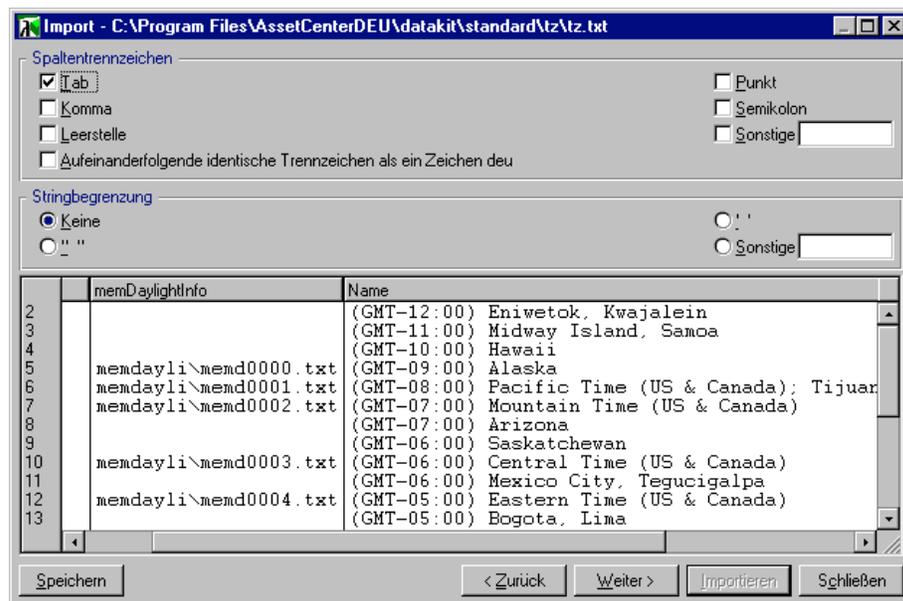
Wenn Sie das Kontrollkästchen **Aufeinanderfolgende identische Trennzeichen als ein Zeichen deuten** markieren, trägt AssetCenter zwei aufeinander folgenden Trennzeichen keine Rechnung und erstellt keine leere Spalte. Wenn Sie eine leere Spalte erstellen möchten, müssen Sie Markierung dieses Kontrollkästchens aufheben und in Ihrer Textdatei zwei hintereinanderliegende Trennzeichen verwenden.

Stringbegrenzung

Geben Sie hier das Zeichen ein, mit dem Sie Text eingrenzen wollen. AssetCenter löscht daraufhin vor der Übertragung des jeweiligen Felds in die Datenbank alle eventuell überflüssigen Zeichen.

Eine Spaltenbegrenzung zwischen zwei Zeichenfolgebegrenzungen wird von AssetCenter als Text angesehen. Die Definition einer Zeichenfolgebegrenzung verpflichtet Sie keineswegs, alle Werte grundsätzlich mit diesem Zeichen zu trennen. Sobald Sie jedoch den Anfang einer Zeichenfolge mit einer Stringbegrenzung gekennzeichnet haben, müssen Sie dieses Zeichen auch an das Ende der Zeichenfolge setzen.

Der Import einer Stringbegrenzung als Wert ist nicht möglich.



Beschreibung des Parsings von Textdateien (Bildschirm 2)

Hinweis: In der unteren Hälfte des Bildschirms erscheint eine Vorschau der zu importierenden Daten mit maximal 25 Zeilen.

Schritt 3 - Beschreibung der zu importierenden Felder

Import einer einzelnen Textdatei

Sobald Sie das unter Schritt 2 beschriebene Parsing vorgenommen haben, klicken Sie auf die Schaltfläche , um den Bildschirm zur Beschreibung der Felder aufzurufen, aus denen sich die Datei zusammensetzt.

Import von mehreren Textdateien

Sobald Sie das unter Schritt 2 beschriebene Parsing vorgenommen haben, klicken Sie auf die Schaltfläche , um den Bildschirm zur Beschreibung der Felder aufzurufen, aus denen sich die Datei zusammensetzt.

Wenn Sie sich im Hauptfenster des Importmoduls befinden, erfolgt der Zugriff auf diesen Bildschirm entweder durch einen Doppelklick auf die Ausgangstabelle oder über das Menü **Bearbeiten/ Eigenschaften**, nachdem Sie die Ausgangstabelle markiert und je nach Parsing ein oder zweimal auf die Schaltfläche  geklickt haben.

Import einer ODBC-Datenbank

Sobald Sie die Datenbank, wie unter Schritt 1 beschrieben, ausgewählt haben, können Sie die Beschreibung der einzelnen Tabellen über den Hauptbildschirm des Importmoduls anzeigen: Der Zugriff auf diese Beschreibung erfolgt entweder durch einen Doppelklick auf die Ausgangstabelle oder über das Menü **Bearbeiten/ Eigenschaften**, nachdem Sie die Ausgangstabelle markiert haben. Die Liste der Ausgangstabellen befindet sich in der Spalte der Ausgangstabellen auf dem Hauptbildschirm des Importmoduls.

Wählen Sie in der Tabelle die Spalte, deren Parameter Sie definieren möchten, mit einem Mausklick auf eine beliebige Stelle der Spalte.

Nummer

In diesem Feld erscheint die Nummer der ausgewählten Spalte.

Wenn Sie beim Ausführen von Schritt 2 das Kontrollkästchen **Erste Importzeile mit Spaltennamen** nicht markiert haben, können Sie die Spaltennummer direkt hier eingeben und müssen nicht mehr auf eine Spalte in der Tabelle klicken.

Name

In diesem Feld erscheint der Name der Spalte und damit der Feldname.

- Wenn Sie beim Ausführen von Schritt 2 das Kontrollkästchen **Erste Importzeile mit Spaltennamen**, markiert haben, ist eine Änderung des Spaltennamens nicht möglich.
- Wenn das Kästchen nicht markiert wurde, verwenden Sie entweder den Standardnamen oder geben einen Namen Ihrer Wahl ein. Dieser Name erleichtert die Identifizierung der Spalten bei den nachfolgenden Schritten.

Typ

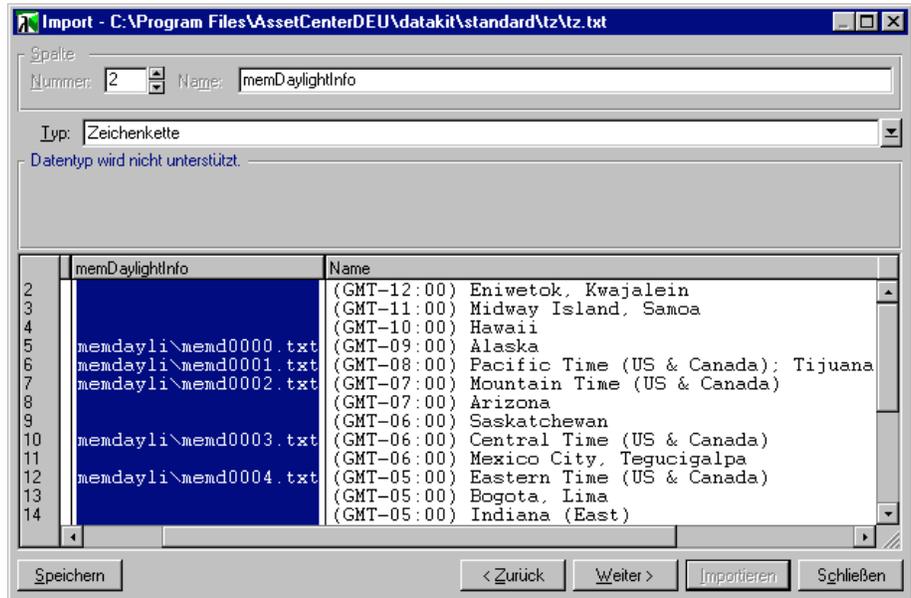
Geben Sie hier den Typ des zu importierenden Felds an. AssetCenter bietet Ihnen die Wahl zwischen den nachstehenden Formaten:

- **Numerisch**: Alle Zeichen müssen numerisch sein. Trifft AssetCenter auf einen nicht numerischen Wert, erhält das Feld den Wert Null.
- **Zeichenkette**: Mit Ausnahme der als Textbegrenzung definierten Zeichen können Sie alle übrigen Zeichen verwenden.
- **Datum**: Beim Import werden nur die Zeichen der Rubrik Datum mit dem entsprechenden Format berücksichtigt. Trifft AssetCenter im Verlauf des Imports auf andere Formate, erhält das Feld den Wert Null.

Format des Datums

Wenn Sie für das Feld den Typ **Datum** festgelegt haben, fordert Sie AssetCenter zur Definition eines Trennzeichens zwischen dem Tag, dem Monat und dem Jahr auf. Zusätzlich müssen Sie die Reihenfolge definieren, in der die drei Angaben erscheinen sollen.

Zusätzlich zu diesen beiden Parametern können Sie die Daten mit denselben Möglichkeiten anzeigen, wie dies bei ihrer Eingabe in AssetCenter der Fall war.



Beschreibung der Ausgangsfelder von Textdateien

Hinweis: In der Tabelle erscheinen maximal 25 Zeilen der zu importierenden Datei.

Schritt 4 - Zuordnung der zu importierenden Felder zu den Feldern der AssetCenter-Datenbank

Import einer einzelnen Textdatei

Sobald Sie die Ausgangsfelder, wie unter Schritt 3 erläutert, beschrieben haben, klicken Sie auf die Schaltfläche **Weiter >**. Es erscheint der Bildschirm, auf dem Sie die Zuordnung der Ausgangsfelder zu den Feldern der AssetCenter-Datenbank durchführen.

- ☞ Verwenden Sie das Feld **Zieltabelle**, um die Textdatei einer Tabelle der AssetCenter-Datenbank zuzuordnen.
- ☞ Ordnen Sie anschließend die Felder der zu importierenden Textdatei (in der Tabelle der Ausgangsfelder) einem Feld der AssetCenter-Datenbank zu (die Felder der Zieltabelle und der verknüpften Tabellen unter dem Feld **Zieltabelle**).

Import von mehreren Textdateien

- ↵ Sobald Sie die Ausgangsfelder, wie unter Schritt 3 erläutert, beschrieben haben, klicken Sie auf , um zum Hauptbildschirm des Importmoduls zurückzukehren.
- ↵ Ordnen Sie die einzelnen Textdateien einer Zieltabelle zu.
- ↵ Ordnen Sie anschließend bei jedem Paar (Textdatei und Zieltabelle) in der rechten Tabelle die Felder der Textdatei einem Feld der AssetCenter-Datenbank zu: Doppelklicken Sie auf das Paar oder markieren Sie das Paar und verwenden Sie das Menü **Bearbeiten/ Eigenschaften**.

Import einer ODBC-Datenbank

- ↵ Sobald Sie die Ausgangsfelder, wie unter Schritt 3 erläutert, beschrieben haben, klicken Sie auf , um zum Hauptbildschirm des Importmoduls zurückzukehren.
- ↵ Ordnen Sie jeder Ausgangstabelle eine Zieltabelle zu.
- ↵ Ordnen Sie anschließend bei jedem Paar (Textdatei und Zieltabelle) in der rechten Tabelle die Felder der Textdatei einem Feld der AssetCenter-Datenbank zu: Doppelklicken Sie auf das Paar oder markieren Sie das Paar und verwenden Sie das Menü **Bearbeiten/ Eigenschaften**.

Schritt 4 - Zuordnung der einzelnen Textdateien oder Ausgangstabellen zu einer Zieltabelle

Hinweis: Dieser Abschnitt ist nur dann von Nutzen, wenn Sie mehrere Textdateien oder ODBC-Datenbanken importieren.

Zuordnung der einzelnen Textdateien oder Ausgangstabellen zu einer Zieltabelle

Klicken Sie auf die Datei bzw. die Ausgangstabelle (Spalte **Ausgangstabellen**) und auf die entsprechende Zieltabelle (Spalte **Zieltabellen**) und dann:

- Verwenden Sie das Menü **Bearbeiten/ Zuordnen**, oder
- Klicken Sie auf das Symbol , oder

- Verwenden Sie das Menü **Bearbeiten/ Nach Namen zuordnen**: AssetCenter führt automatisch eine Zuordnung von Dateien bzw. Tabellen mit demselben Namen durch. Bei der Zuordnung wird der technische Name des Felds verwendet.

Verwenden Sie das Menü **Bearbeiten/ Dissoziieren** oder das Symbol , um die Zuordnung einer Datei bzw. einer Ausgangstabelle zu einer Zieltabelle aufzuheben.

Anzeige der Struktur der AssetCenter-Datenbank

Doppelklicken Sie auf die Zieltabelle in der Spalte **Zieltabellen** oder wählen Sie die Zieltabelle mit der Maus und verwenden Sie das Menü **Bearbeiten/ Eigenschaften**. AssetCenter zeigt die Liste der Felder, ihren Typ und ihre Länge an.

Schritt 4 - Zuordnung der zu importierenden Felder zu den Feldern der AssetCenter-Datenbank

Ausgangsfelder

In dieser Bildschirmhälfte erscheinen die Namen, die Sie den Spalten, wie unter Schritt 3 (Textdateien) beschrieben, zugeordnet haben, oder die Kurzbeschreibung der Felder (AssetCenter-Datenbank).

Zieltabelle

Import einer einzelnen Textdatei

Wählen Sie die Zielhaupttabelle, in die Sie die Informationen übertragen wollen. AssetCenter zeigt daraufhin die Struktur dieser Tabelle an, die sich aus direkten und aus verknüpften Feldern zusammensetzt.

Import von mehreren Textdateien oder einer ODBC-Datenbank

AssetCenter zeigt die Struktur der Zieltabelle an, die der Datei bzw. der Ausgangstabelle (direkte und verknüpfte Felder) zugeordnet ist.

Ausgangsfelder und Zielfelder zuordnen

Sie verfügen über die nachstehenden Möglichkeiten:

- Ziehen Sie das Ausgangsfeld mit der Maus auf das Zielfeld, um die Zuordnung vorzunehmen.
- Zusätzlich besteht die Möglichkeit, zunächst das Ausgangsfeld und dann das Zielfeld zu wählen, um die beiden Felder anschließend durch Klicken auf das Symbol  einander zuzuordnen.
- Zum Aufheben einer Zuordnung klicken Sie zunächst auf das Paar Ausgangsfeld und Zielfeld und dann auf das Symbol .
- Das Symbol  ermöglicht eine automatische Zuordnung von Ausgangsfeldern zu Zielfeldern, sofern Sie denselben Namen tragen. Bei der Zuordnung wird der technische Name des Felds verwendet. Dieser Name ist in der Datenbankstruktur sichtbar.

Zusätzlich berechnete Felder in der Ausgangsdatei hinzufügen

AssetCenter bietet Ihnen die Möglichkeit, zusätzliche Felder zu Ihrer Ausgangsdatei hinzuzufügen. Diese Felder sind jedoch nicht direkt auf dem Bildschirm integriert, sondern befinden sich im Speicher.

Verwenden Sie die Symbole ,  und , um diese Felder hinzuzufügen, zu löschen oder anzuzeigen.

Auf Kriterien passende Datensätze suchen

Sie haben die Möglichkeit, ein oder mehrere Paare vom Typ Ausgangsfeld-Zielfeld zu wählen, um anhand dieser Felder die Kriterien zur Identifikation von Datensätzen festzulegen. Mit diesen Kriterien können Sie dann Datensätze in Tabellen identifizieren. Wenn Sie mehrere Kriterien definieren, kommen bei der Identifizierung von Datensätzen sämtliche Kriterien zur Anwendung.

Wählen Sie die als Kriterium zu definierenden Paare vom Typ Ausgangsfeld-Zielfeld und klicken Sie auf die Schaltfläche . Ist die Schaltfläche aktiviert, erscheint sie hell und wie eine niedergedrückte Taste. Das kleine Symbol links neben dem Zielfeld oder der Zielverknüpfung erscheint wie folgt: .

AssetCenter importiert nacheinander die einzelnen Zeilen der Ausgangsdatei und geht dabei wie folgt vor:

- ↪ AssetCenter sucht in der AssetCenter-Datenbank nach einem Datensatz mit identischen Kriterien. Ist ein solcher Datensatz vorhanden, ändert AssetCenter den Datensatz unter Berücksichtigung der in der Textdatei enthaltenen Informationen.
- ↪ Liegen mehrere Datensätze mit denselben Kriterien vor, hält die Software bei dem ersten Datensatz an und lässt die übrigen Datensätze unberücksichtigt. Dementsprechend ist unbedingt auf die richtige Auswahl der Kriterien zu achten.
- ↪ Liegt kein dem Kriterium entsprechender Datensatz vor, erstellt AssetCenter einen neuen Datensatz in der Datenbank.

Hinweis: 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

Parameter für die Erstellung verknüpfter Datensätze definieren

Sie möchten eine Datei importieren, deren Daten in eine Reihe unterschiedlicher Tabellen importiert werden sollen. Das ist beispielsweise der Fall bei einer Datei, die Personen und die entsprechenden Standorte enthält. Wählen Sie hierzu die Hauptzieltabelle, die in unserem Beispiel die Tabelle mit den Personen ist, und geben Sie über die Verknüpfungen die übrigen Tabellen an, in unserem Beispiel die Tabelle der Standorte, in die die Daten importiert werden sollen.

AssetCenter lässt Sie die Parameter definieren, die bei der Erstellung von Datensätzen in der verknüpften Tabelle gelten sollen, sofern der Datensatz zum Zeitpunkt des Imports nicht existiert. Die Definition dieser Parameter erfolgt mit Hilfe des Symbols , wobei das Symbol nur zusammen mit den Verknüpfungen, nicht jedoch zusammen mit den Feldern der verknüpften Tabellen verwendet werden kann. Verknüpfungen werden mit den Symbolen  und  gekennzeichnet.

Gehen Sie zum Anzeigen des Bildschirms und Definieren der Parameter wie folgt vor:

- ↪ Ordnen Sie das zu importierende Feld dem Feld in der verknüpften Tabelle zu.

↪ Klicken Sie auf die entsprechende Verknüpfung.

↪ Klicken Sie auf das Symbol .

Wenn ein Datensatz mit Hilfe der eingegebenen Kriterien gefunden wird, ändert die Software gegebenenfalls die Informationen dieses Datensatzes.

Datensatz erstellen

Der Datensatz wird erstellt, sofern mit Hilfe der Identifizierungsschlüssel kein entsprechender Datensatz in der Datenbank gefunden wurde.

Datensatz nicht erstellen

Der Datensatz wird nicht erstellt, auch wenn mit Hilfe der Identifizierungsschlüssel kein entsprechender Datensatz in der Datenbank gefunden wurde.

Fehler anzeigen

AssetCenter zeigt einen Fehler an, wenn der Datensatz in der Datenbank mit Hilfe der Identifizierungsschlüssel nicht gefunden wurde.

Nur bereits verknüpfte Datensätze berücksichtigen (🔗)

Wenn Sie das Symbol 🔗 (Heftzwecke) neben eine Verknüpfung setzen, berücksichtigt das Importmodul nur die Datensätze, die bereits mit dem Hauptdatensatz verknüpft sind.

Ein Beispiel: Sie importieren eine Liste mit Personen und den Gegenständen, die diese Personen benutzen. Die Zieltabelle ist die Tabelle der Abteilungen und Personen. Sie ordnen der Verknüpfung mit der Vermögensstabelle eine Heftzwecke zu. Für jeden Vermögensgegenstand, der nur einer Person in der Ausgangsdatei zugeordnet ist, trägt das Importmodul der Gesamtheit der von der Person benutzten Gegenstände Rechnung (Registerkarte **Vermögen** im Personendetail).

Im Fall einer Verknüpfung vom Typ **Own** wird die Heftzwecke automatisch platziert und lässt sich nicht mehr entfernen. Bei Verknüpfungen vom Typ **Own** werden die verknüpften Datensätze automatisch gelöscht, sobald der entsprechende Hauptdatensatz gelöscht wird. Die Verknüpfung Person/ Schulungen ist ein Beispiel für eine solche Verknüpfung: Wenn Sie eine Person löschen, entfernen Sie damit gleichzeitig alle Schulungen, die dieser Person zugeordnet sind.

Je nach Verknüpfung hat die Heftzwecke unterschiedliche Auswirkungen:

- Wenn es sich bei der Zieltabelle um die Vermögenstabelle handelt, und Sie die Heftzwecke der Verknüpfung **Benutzer** zuordnen, sucht das Importmodul nur die mit den Vermögensgegenständen verknüpften Benutzer. Da es für einen Vermögensgegenstand nur jeweils einen Benutzer gibt, besteht die Möglichkeit, einen Benutzer zu ändern oder zu erstellen, ohne dass dazu die Identifizierung des Benutzers mit einem Identifizierungsschlüssel erforderlich wäre. Das ist auch dann von Nutzen, wenn Sie den Wert eines Merkmals für einen bestimmten Vermögensgegenstand ändern möchten.
- Wenn es sich bei der Zieltabelle um die Tabelle der Abteilungen und Personen handelt, und Sie die Heftzwecke der Verknüpfung **Vermögen** zuordnen, sucht das Importmodul nur die mit der Person verknüpften Vermögensgegenstände. Wenn Sie in diesem Fall die Vermögensgegenstände eines Benutzers ändern oder erstellen möchten, benötigen Sie unbedingt die Identifizierungsschlüssel der Gegenstände. Sie können eventuell weniger präzise sein, als beim Vorhandensein einer Heftzwecke.

Information: Bei den Verknüpfungen zwischen Datensätzen sind drei verschiedene Typen zu unterscheiden.
Die Verknüpfungen vom Typ n. Beispiel: Ein Gegenstand kann nur mit einem Standort verknüpft sein; ein Standort kann mit mehreren Gegenständen verknüpft sein.
Die Verknüpfungen vom Typ 1. Beispiel: Ein Gegenstand kann nur mit einem einzigen Kommentar verknüpft sein; ein Kommentar kann nur mit einem einzigen Gegenstand verknüpft sein.
Die Verknüpfungen vom Typ n-n. Beispiel: Ein Verbrauchsgut kann mit mehreren Produkten verknüpft sein; ein Produkt kann mit mehreren Verbrauchsgütern verknüpft sein.

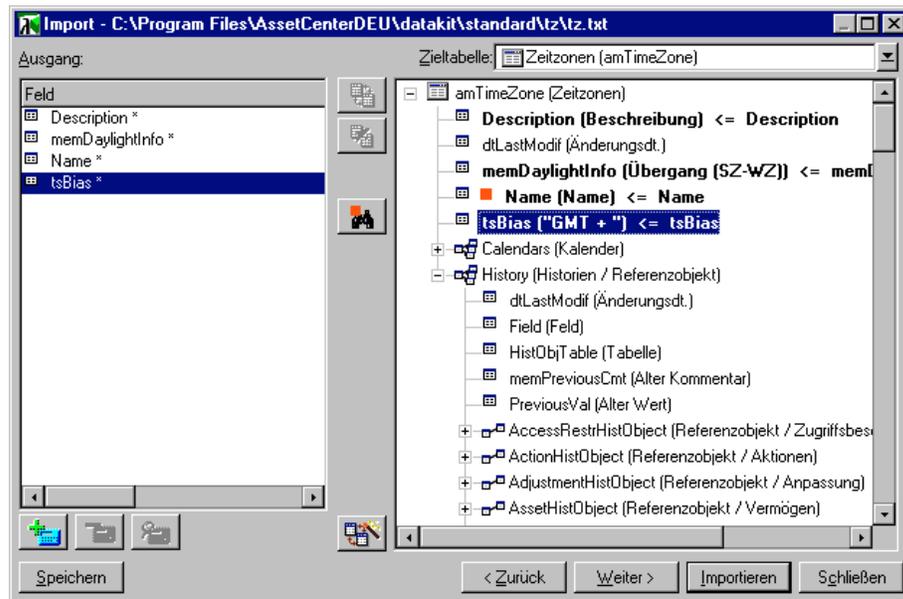
In der Baumstruktur der Zieltabellen verwendete Symbole

■ Der auf das Symbol folgende Name entspricht einer Tabelle (großes Symbol) oder einem Feld (kleines Symbol).

▣ Der auf das Symbol folgende Name entspricht einer Tabelle, die mit einer übergeordneten Tabelle verknüpft ist. Über das Detail eines Datensatzes aus der übergeordneten Tabelle kann nur jeweils ein einziger Datensatz der verknüpften Tabelle gewählt werden. Diese Art von Verknüpfung entspricht einem Feld, dessen Wert mit Hilfe eines Auswahlfensters oder einer Dropdown-Liste definiert werden kann.

☞ Der auf das Symbol folgende Name entspricht einer Tabelle, die mit einer übergeordneten Tabelle verknüpft ist. Über das Detail eines Datensatzes aus der übergeordneten Tabelle können mehrere Datensätze der verknüpften Tabelle gewählt werden. Diese Art von Verknüpfung entspricht einer Liste von Datensätzen, die sich in einer Registerkarte des Datensatzdetails der übergeordneten Tabelle befindet.

☞ Bei diesem Symbol handelt es sich um eine Heftzwecke. Eine Heftzwecke kann nur einer Verknüpfung zugeordnet werden, der auf eine Tabelle verweist, in der ein Feld einem der zu importierenden Felder zugeordnet ist. Wenn eine Heftzwecke gesetzt wurde, sucht das Importmodul nur in den Datensätzen, die mit dem Datensatz verknüpft sind, und dementsprechend mit einer Heftzwecke versehen sind. Der Einsatz von Heftzwecken wird über die Optionen gesteuert, die Sie durch Klicken auf das Symbol  abrufen.



Zuordnung von Ausgangsfeldern zu den Feldern der Datenbank

Schritt 4 - Zuordnung zusätzlicher berechneter Felder zur Ausgangsdatei

AssetCenter bietet Ihnen die Möglichkeit, zusätzliche Felder zu Ihrer Ausgangsdatei hinzuzufügen. Diese Felder sind jedoch nicht direkt auf dem Bildschirm integriert, sondern befinden sich im Speicher.

Verwenden Sie die Symbole ,  und , um diese Felder hinzuzufügen, zu löschen oder anzuzeigen.

Name

Geben Sie dem neuen Feld einen Namen.

Feldtyp

Geben Sie die Struktur an, die für das neue Feld gelten soll. Der Bildschirm passt sich dem von Ihnen gewählten Typ.

Verkettung

Wenn Sie diesen Modus wählen, können Sie mehrere Felder der Ausgangsdatei kombinieren. Wählen Sie nacheinander die entsprechenden Felder. Definieren Sie für die Abgrenzung der einzelnen Felder ein beliebiges Zeichen. Dabei ist die Verwendung des Zeichens **völlig ausreichend**.

Beispiel: Feld1 und Feld2

Feste Extraktion

Dieser Modus lässt Sie einen Teil eines Felds aus einer Textdatei extrahieren:

- ❖ Wählen Sie das Ausgangsfeld, auch Hauptfeld genannt.
- ❖ Nehmen Sie die erforderlichen Eingaben im Feld **Anzahl nicht verwendeter Zeichen** vor. Die entsprechenden Zeichen werden daraufhin von AssetCenter nicht berücksichtigt.
- ❖ Geben Sie die erforderliche Zahl im Feld **Anzahl verwendeter Zeichen** an. Nach Ausschluß der nicht verwendeten Zeichen berücksichtigt AssetCenter nur diese Anzahl von Zeichen.
- ❖ Markieren Sie das Kontrollkästchen **Extraktion am Feldende beginnen**, wenn AssetCenter die **Anzahl nicht verwendeter Zeichen** ab dem Feldende unberücksichtigt lassen soll, und die **Anzahl verwendeter Zeichen** nach Ausschluß der nicht verwendeten Zeichen vom ersten Zeichen an ab dem Feldende berücksichtigen soll.

Beispiel:

- ❖ **Anzahl nicht verwendeter Zeichen: 3**
- ❖ **Anzahl verwendeter Zeichen: 5**

- ❖ Wert des Felds in der Ausgangsdatei: **REFIMP05A18500**
- ❖ In die Datenbank importierter Wert: **IMP05**, wenn das Kontrollkästchen **Extraktion am Feldende beginnen** nicht markiert wurde, und **05A18**, wenn das Kästchen aktiviert wurde.

Begrenzte Extraktion

Dieser Modus lässt Sie einen Teil eines Felds aus einer Ausgangsdatei extrahieren:

- ❖ Wählen Sie das Ausgangsfeld, auch **Hauptfeld** genannt.
- ❖ Geben Sie die **Trennzeichen** an, die innerhalb der Werte des Hauptfelds verwendet werden.
- ❖ Geben Sie die Anzahl der **Nicht verwendeten Trennzeichen** an. AssetCenter berücksichtigt alle Informationen, die im Anschluss an diese Zeichen erscheinen.
- ❖ Geben Sie die Anzahl der **Verwendeten Trennzeichen** an: AssetCenter berücksichtigt die Informationen, die zwischen dem Anfang des zu berücksichtigenden Texts und dem Trennzeichen erscheinen, das auf das letzte zu beachtende Trennzeichen folgt.
- ❖ Markieren Sie das Kontrollkästchen **Extraktion am Feldende beginnen**, wenn AssetCenter die Anzahl der **Nicht verwendeten Trennzeichen** und die Anzahl der **Verwendeten Trennzeichen** ab dem Ende des Felds berücksichtigen soll.

Beispiel:

- ❖ **Trennzeichen:** /
- ❖ **Anzahl nicht verwendeter Trennzeichen:** 2
- ❖ **Anzahl verwendeter Trennzeichen:** 3
- ❖ Wert des Felds in der Ausgangsdatei: **1/2/3/4/5/6/7/8/9**
- ❖ In die Datenbank importierter Wert: **3/4/5/6**, wenn das Kontrollkästchen **Extraktion am Feldende beginnen** nicht markiert wurde, und **4/5/6/7**, wenn das Kästchen aktiviert wurde.

Fester Wert

Dieser Modus ermöglicht die Anzeige der nachstehenden Kombinationen:

- ❖ Mit einem " voneinander abgegrenzte Zeichenfolgen.
- ❖ Variablen. Es handelt sich hierbei um Variablen, die zusammen mit den Standardwerten der Felder AmLoginName(), AmDate(), AmCounter() verwendet werden.

Baumstruktur

Dieser Modus lässt Sie eine Baumstruktur anhand eines einzigen Felds der Ausgangsdatei wiederherstellen.

- ↪ Wählen Sie das Ausgangsfeld, auch Hauptfeld genannt.
- ↪ Geben Sie die Trennzeichen an, die zwischen den Nebenfeldern des Felds verwendet werden sollen.

AssetCenter teilt das Ausgangsfeld in ebenso viele Nebenwerte auf, wie durch ein Trennzeichen voneinander abgegrenzte Zeichenfolgen vorhanden sind. Anschließend wird für jeden der hierarchisch geordneten Nebenwerte ein Datensatz erstellt.

Beispiel:

- ↪ Sie erstellen eine Textdatei, die die Spalte **Name** enthält. In einer der Zeilen der Datei erscheint der Wert **Niederlassung Deutschland/Vertriebsleitung/Marketingleitung**.
- ↪ Sie konfigurieren das Importmodul durch Erstellen eines Formelfeld vom Typ **Baumstruktur** (das Trennzeichen ist /). Der Name des Felds lautet **FormelFeld**. Zusätzlich erstellen Sie ein Formelfeld **Fester Wert** (Wert = 1), das Sie dem Feld **Abteilung** (SQL-Name: bDepartment) zuordnen, um festzulegen, dass es Abteilungen und nicht Personen erstellen muss.
- ↪ Ordnen Sie dem Feld **Name** (SQL-Name: Name) in der Tabelle der Personen den Wert **FormelFeld** zu.
- ↪ Starten Sie die Importdatei.
- ↪ Ergebnis: Es werden 3 hierarchisch geordnete Abteilungen erstellt: **Niederlassung Deutschland, Vertriebsleitung** und **Marketingabteilung**.

Datei

Dieser Modus ermöglicht das Importieren einer Datei in die Datenbank. Der Modus ist besonders nützlich beim Importieren von Bildern bzw. langen Texten.

Sie können die Dateien nur in Felder importieren, die einem der nachstehenden Typen entsprechen:

- ❖ Memo
- ❖ Blob

Die importierten Dateien können in einem der nachstehende Formate erscheinen:

- ❖ ANSI-Text

- ❖ Bilder (Sie können alle von AssetCenter unterstützten Bildformate importieren.)

Geben Sie in der Formel zur Berechnung des Felds das Ausgangsfeld an, das den vollständigen Pfad (Pfad, Name und Erweiterung) der zu importierenden Datei enthält. Standardmäßig entspricht der Pfad dem aktuellen Verzeichnis.

Skript

In diesem Modus können Sie einen Wert mit Hilfe eines BASIC-Skripts berechnen. Das Skript kann sich auf importierte Ausgangsfelder beziehen.

Zum Bearbeiten eines Berechnungsskripts geben Sie das Skript direkt ein oder verwenden einen Assistenten durch Klicken auf die Schaltfläche .

Das Skript kann sich nicht auf Felder der Datenbank beziehen.

Test

Hinweis: Dieses Feld wird nur angezeigt, wenn es sich bei dem Typ der Feldformel um **Feste Extraktion** oder **Begrenzte Extraktion** handelt.

Geben Sie einen Wert in ein Feld Ihrer Wahl ein.

Ergebnis

Hinweis: Dieses Feld wird nur angezeigt, wenn es sich bei dem Typ der Feldformel um **Feste Extraktion** oder **Begrenzte Extraktion** handelt.

In diesem Feld erscheint der Testwert des simulierten Importergebnisses.

Schritt 4 - Sonderfälle des Imports

Import von Abteilungen und Personen

Beim Import von Datensätzen aus der Tabelle der Abteilungen und Personen müssen Sie unter Umständen festlegen, ob es sich bei dem importierten Datensatz um eine Person oder eine Abteilung handelt.

Zu diesem Zweck steht Ihnen das Feld **Abteilung** (SQL-Name: bDepartment) zur Verfügung. In diesem Feld erscheint eine **1** wenn es sich um eine Abteilung handelt und eine **0**, wenn es sich um eine Person handelt. Standardmäßig geht das Importmodul davon aus, dass in diesem Feld eine **0** steht.

Wir empfehlen Ihnen, beim Import eine Formel vom Typ **1** zu erstellen, und diesen Wert dem Feld **Abteilung** zuzuordnen, wenn das importierte Element eine Abteilung ist.

Hinweis: Das Importmodul geht davon aus, dass der importierte Datensatz eine Abteilung ist, wenn dieser Datensatz über einen untergeordneten Datensatz in der Tabelle der Abteilungen und Personen verfügt. Personen weisen keine untergeordneten Datensätze auf.

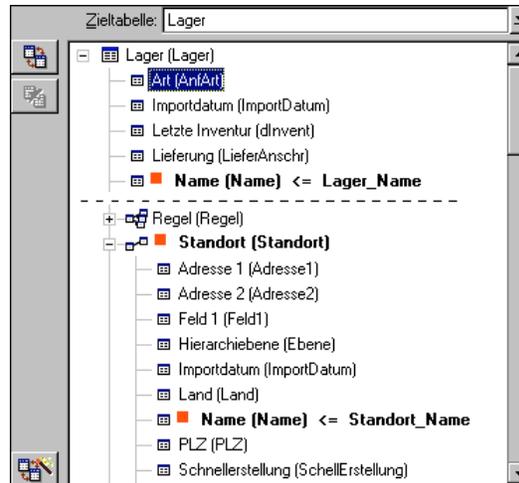
Import von Dokumenten

Zum Importieren von Dokumenten in die Dokumententabellen von AssetCenter müssen Sie dem Feld **Tabelle** (SQL-Name: DocObjTable) das zu importierende Feld in der Dokumententabelle zuordnen, wobei dieses Feld auf den SQL-Namen der Tabelle verweist, mit der das Dokument verknüpft ist.

Schritt 4 - Anwendungsbeispiele für Identifizierungsschlüssel

Im Folgenden erklären wir Ihnen, wie AssetCenter die Wahl der Schlüssel interpretiert:

Beispiel 1: Ein verknüpftes Feld als Schlüssel für die Haupttabelle verwenden



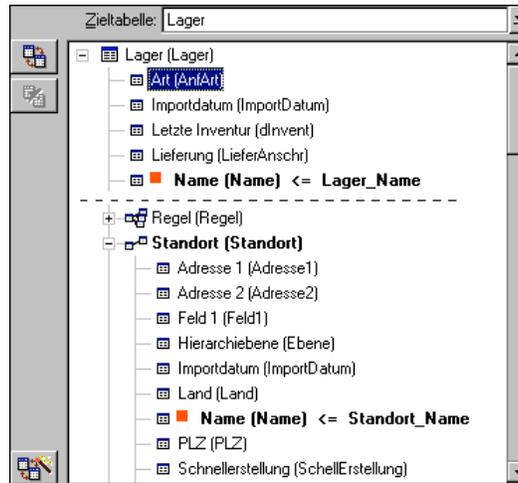
Im nachstehenden Beispiel wird das Lager anhand von zwei Hauptschlüsseln identifiziert:

- Lager.Name: Das Feld **Name (Name) <= Lager_Name** wird als Schlüssel definiert und ist in der Haupttabelle enthalten.
- Standort.Name: Das verknüpfte Feld **Name (Name) <= Standort_Name** wird als Schlüssel für die Tabelle mit den Standorten definiert und die Verknüpfung **Standort (Standort)** wird als Identifizierungsschlüssel definiert.

In demselben Beispiel wird der Standort als Hauptschlüssel angegeben:

- Standort.Name: Das Feld **Name (Name) <= Standort_Name** wird als Schlüssel der Tabelle mit den Standorten definiert.

Beispiel 2: Ein Feld als Schlüssel einer verknüpften Tabelle definieren, ohne dass das Feld auf der Ebene der Haupttabelle als Schlüssel dient



In diesem Beispiel wird das Lager anhand eines Schlüssels identifiziert:

- Lager.Name: Das Feld **Name (Name) <= Lager_Name** wird als Schlüssel definiert und ist in der Haupttabelle enthalten.
- Die Verknüpfung **Standort (Standort)** dient nicht als Schlüssel.

Im obigen Beispiel wird der Standort anhand eines Schlüssels identifiziert:

- Standort.Name: Das Feld **Name (Name) <= Standort_Name** wird als Hauptschlüssel definiert.

Schlußfolgerung

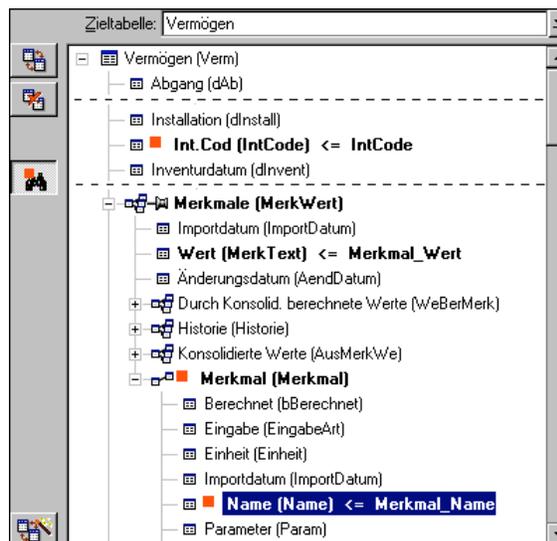
- Es ist durchaus möglich, bestimmte Schlüssel für die Haupttabelle festzulegen, und davon unabhängig andere Schlüssel für die mit der Haupttabelle verknüpften Tabellen zu definieren. Das bedeutet, dass Sie ohne weiteres Daten mit Hilfe einer einzigen Textdatei in mehrere Tabellen importieren können.
- Wenn Sie ein Feld einer verknüpften Tabelle als Schlüssel der Haupttabelle definieren wollen, müssen Sie sowohl das Feld der verknüpften Tabelle als auch die Verknüpfung wählen. Wählen Sie keine Verknüpfung, kommt der Schlüssel nur für die verknüpfte Tabelle zur Anwendung.

Beispiel 3: Kriterien definieren, mit denen die Merkmalwerte der in der Datenbank enthaltenen Datensätze aktualisiert werden können

Wenn Sie den Wert des Merkmals eines Datensatzes mit einem bestimmten Wert aktualisieren möchten, müssen Sie in der Datenbank ein Paar vom Typ Datensatz-Merkmal suchen und diesem Paar einen neuen Wert zuordnen. Sollte ein solches Paar nicht vorhanden sein, greift AssetCenter auf die Optionen zurück, die gegebenenfalls über das Symbol  auf der Ebene der Verknüpfung zum Erstellen eines verknüpften Datensatzes definiert wurden.

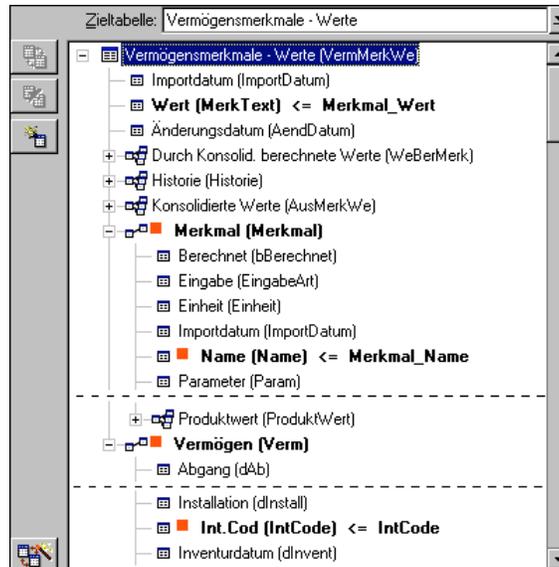
Dabei stehen zwei Methoden zur Auswahl:

Erste Methode anhand eines Beispiels aus der Tabelle der Vermögensgegenstände



- ❖ Mit dem Schlüssel  **Int.Cod (IntCode)** wird der zu ändernde Vermögensgegenstand identifiziert.
- ❖ Das Symbol  neben der Verknüpfung  **Merkmale (MerkWert)** weist darauf hin, dass nur die Merkmale des Gegenstands gesucht werden.
- ❖ Der Schlüssel  **Merkmal (Merkmal)** besagt, dass das Paar Merkmal-Wert anhand des Merkmals identifiziert wird.
- ❖ Der Schlüssel  **Name (Name) <= Merkmal_Name** weist darauf hin, dass das Merkmal anhand des Namens identifiziert wird.
- ❖ Der neue Wert erscheint im Feld **Wert (MerkText) <= Merkmal_Wert**.

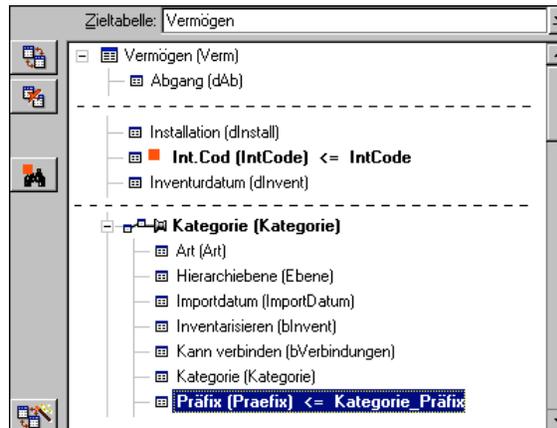
Zweite Methode



- ❖ Das Paar (**Merkmal (Merkmal)** , **Vermögen (Verm)**) verweist auf das dem Vermögensgegenstand zugeordnete Merkmal.
- ❖ Der Schlüssel **Name (Name)** verweist auf den Feldschlüssel der Verknüpfung **Merkmal (Merkmal)** .
- ❖ Der Schlüssel **Int.Cod (IntCode)** verweist auf das Feldschlüssel der Verknüpfung **Vermögen (Verm)** .
- ❖ Der neue Wert erscheint im Feld **Wert (MerkText) <= Merkmal_Wert** .

Beispiel 4: Den Inhalt eines verknüpften Datensatzes ändern, für den kein Identifizierungsschlüssel vorliegt

Beispiel: Sie möchten das Präfix der Kategorie ändern, die mit einem bestimmten Vermögensgegenstand verknüpft ist. In Ihrer Importdatei befindet sich kein Schlüssel, mit dem die Kategorie identifiziert werden kann. Sie verfügen lediglich über die Information, dass es sich um eine mit einem bestimmten Gegenstand verknüpfte Kategorie handelt.



- Der Schlüssel **Int.Cod (IntCode) <= IntCode** verweist auf den Vermögensgegenstand.
- Das Symbol  neben  **Kategorie (Kategorie)** besagt, dass nur nach den Kategorien gesucht werden soll, die bereits mit dem Gegenstand verknüpft sind.
- Das Feld  **Präfix (Präfix) <= Kategorie_Präfix** lässt Sie dem Feld **Präfix** (SQL-Name: Prefix) der Kategorie einen neuen Wert zuordnen.

Leistungsfähigkeit: Um die Leistungsfähigkeit des Imports zu optimieren, empfehlen wir Ihnen, die Schlüssel unter den Feldern auszuwählen, die den Index der Tabelle darstellen. Warnung: Ein Reihe von Indizes setzen sich aus mehreren Feldern zusammen.

Schritt 5 - Parameter für die Übertragung der Daten

Import einer einzelnen Textdatei

Sobald Sie die Ausgangsfelder, wie unter Schritt 4 beschrieben, den Feldern der Datenbank zugeordnet haben, klicken Sie auf die Schaltfläche , um das Fenster zur Konfiguration der Datenübertragung anzuzeigen.

Import mehrerer Textdateien oder einer ODBC-Datenbank

Verwenden Sie das Menü **Bearbeiten/ Optionen**, das Ihnen auf dem Hauptbildschirm des Importmoduls zur Verfügung steht.

Rahmen *Fehler verwalten*

Wählen Sie die zur Verwaltung von Fehlern geeignete Option:

Import bei Fehler unterbrechen

Der Import wird unmittelbar nach dem Auftreten eines Fehlers unterbrochen.

Jede importierte Zeile bestätigen

Das Importmodul fordert Sie auf, beim Import jede Zeile einzeln zu bestätigen. Sobald ein Fehler in der Zeile erkannt wird, sei es in der Haupttabelle oder in einer der verknüpften Tabellen, wird der Import der gesamten Zeile annulliert. Die Software stellt den Zustand der Datenbank vor dem Import dieser Zeile wieder her.

Zeilengruppen bestätigen

Das Importmodul bearbeitet beim Import ganze Zeilengruppen. Die Definition der Anzahl der Zeilen in diesen Gruppen bleibt Ihnen überlassen. Sobald ein Fehler in der Gruppe erkannt wird, sei es in der Haupttabelle oder in einer der verknüpften Tabellen, wird der Import sämtlicher in der Gruppe enthaltenen Zeilen abgebrochen. Die Software stellt den Zustand der Datenbank vor dem Import dieser Zeilengruppe wieder her. Die Integrität der Datenbank ist dadurch gewährleistet.

Rahmen *Protokoll*

Wählen Sie die Vorgänge, die in der Protokolldatei aufgeführt werden sollen:

- Fehler
- Hinzufügen und Aktualisieren

Geben Sie den Namen und den Pfad der Protokolldatei an. Sollte diese Datei nicht vorhanden sein, kann sie ohne weiteres von AssetCenter erstellt werden. Geben Sie eine beliebige Erweiterung an. Wir empfehlen Ihnen die Verwendung der Erweiterung **.log**.

Warnung: Das Importmodul kann keine Verzeichnisse erstellen.

Die Protokolldatei enthält unter anderem die nachstehenden Informationen:

- Uhrzeit des protokollierten Vorgangs
- Beschreibung des protokollierten Vorgangs

- Fehler

Die Protokolldatei wird beim Import überschrieben.

Schritt 6 - Übertragung der Daten

Sie können den Import nach jedem der auf den vorhergehenden Seiten beschriebenen Schritten starten, sofern ausreichend Informationen vorliegen. Klicken Sie hierfür auf die Schaltfläche .

Verhalten auf der Ebene der Datensätze

- AssetCenter führt einen zeilenweisen Import der Daten unter Berücksichtigung der in der Datei geltenden Reihenfolge durch.
- AssetCenter kann die Daten aus einer einzelnen Zeile in unterschiedliche Tabellen importieren.
- Wenn der Import eines Teils der Zeile nicht möglich ist, wird der fehlende Teil gegebenenfalls von AssetCenter erstellt.
- AssetCenter sucht in der AssetCenter-Datenbank nach Datensätzen, deren Identifizierungsschlüssel identische Werte aufweisen. Liegt ein solcher Datensatz vor, ändert AssetCenter diesen Datensatz unter Berücksichtigung der in der Textdatei enthaltenen Informationen.
- Liegt kein Datensatz vor, dessen Identifizierungsschlüssel identische Werte aufweisen, erstellt AssetCenter einen neuen Datensatz in der Datenbank.
- Geben Sie keinen Schlüssel an, fügt das Importmodul die importierten Datensätze hinzu, sofern die vorgegebenen Werte die eindeutige Kennzeichnung der Datenbankfelder nicht beeinträchtigen. Ohne Identifizierungsschlüssel ist die Aktualisierung der Datensätze durch das Importmodul nicht möglich.
- Das Importmodul macht keinen Unterschied zwischen dem Import von Hauptelementen und dem Import von verknüpften Elementen.

Verhalten auf der Ebene der Felder

- Das Steuern der obligatorischen Eingabe ist über das Importmodul nicht möglich. Das bedeutet, dass Sie die Felder, deren Eingabe obligatorisch ist, beim Import der Daten manuell prüfen müssen.

- Trifft das Importmodul bei einem Feld vom Typ **Systemaufzählung** auf einen Fehler, wird der Import der entsprechenden Zeile abgelehnt.
- Trifft das Importmodul auf einen Wert, der noch in keiner Standardaufzählung geführt wird, importiert das Modul die Zeile und fügt den neuen Wert in der Aufzählung hinzu, sofern sie offen ist. Ist die Aufzählung jedoch geschlossen, wird der Import der Zeile abgebrochen.
- Überschreitet der Wert eines Felds die maximale Länge, wird das Ende des Werts abgeschnitten.
- Die Felder, deren Werte in der Textdatei nicht vorhanden sind, löschen den in der Datenbank für das entsprechende Feld vorhandenen Wert.
- Führt der Import einer Zeile zur Erstellung eines neuen Datensatzes, fügt AssetCenter den Standardwert der Felder ein, die in der Textdatei bzw. der Ausgangstabelle nicht in Form einer Spalte vorhanden sind.
Wenn eine Spalte vorhanden, jedoch kein Wert angegeben ist, fügt AssetCenter keinen Standardwert ein, sondern importiert den leeren Wert.

AssetCenter-Datenbank importieren

Der Import einer AssetCenter-Datenbank erfolgt über das Menü Datei/ Import, Option Datenbank importieren, Registerkarte AssetCenter.

Sie finden ausführliche Informationen zum Importieren einer AssetCenter-Datenbank im Handbuch "Installations- und Aktualisierungshandbuch", Kapitel "Aktualisierung von AssetCenter", Abschnitt "Etappe 1: Alte Datenbank konvertieren", Unterabschnitt "Alte Datenbank in neue AssetCenter 3.0-Datenbank importieren".

Importskripte speichern und ausführen

Ein Skript besteht aus einer Reihe von Importparametern, die unter einem bestimmten Namen gespeichert wurden. Importskripte

ermöglichen das wiederholte Durchführen eines Imports unter ähnlichen Bedingungen, ohne dass Sie deshalb die Parameter jedes Mal neu definieren müssten. Das bedeutet, dass Sie durch die Verwendung von Skripten viel Zeit sparen können.

Skripte werden in folgenden Situationen sinnvoll eingesetzt:

- ❖ Wiederholtes Ausführen eines Imports, bis er Ihren Wünschen entsprechend abgelaufen ist. Eine Änderung der ursprünglichen Datei zwischen den einzelnen Versuchen ist ohne weiteres möglich.
- ❖ Regelmäßiges Aktualisieren der Datenbank, beispielsweise der Tabelle der Personen anhand einer Datei der Personalabteilung.

AssetCenter lässt Sie Skripte speichern und anschließend ausführen.

Skripte speichern

Import einer einzelnen Textdatei

- ❖ Verwenden Sie das Menü **Datei/ Import**, um den Import einer Textdatei zu wählen.
- ❖ Definieren Sie die Importparameter, also Namen und Standort der Datei, Struktur der Datei usw.
- ❖ Dabei können Sie jederzeit auf die Schaltfläche  klicken. AssetCenter gibt Ihnen daraufhin die Möglichkeit, die entsprechenden Bedingungen in Form einer Skriptdatei zu speichern.

Import einer Datenbank bzw. mehrerer Textdateien

- ❖ Verwenden Sie das Menü **Datei/ Import**, um den Import einer Datenbank zu wählen.
- ❖ Definieren Sie die Importparameter, also Standort der Datenbank bzw. Textdateien, die Zuordnung von Feldern usw.
- ❖ Dabei haben Sie jederzeit die Möglichkeit, die entsprechenden Bedingungen in Form einer Skriptdatei über das Menü **Datei/ Speichern** oder **Datei/ Speichern unter** zu speichern.

Skripte ändern

Import einer einzelnen Textdatei

- ❖ Verwenden Sie das Menü **Datei/ Import**.

- ❖ Wählen Sie den Import einer Textdatei.
- ❖ Zeigen Sie im Fenster **Datendatei öffnen** in der Dropdown-Liste im Feld **Typ** den Wert **Importskript (*.scr)** an.
- ❖ Öffnen Sie das Skript.
- ❖ Ändern Sie die Importparameter (Standort der Datenbank, oder der Textdateien, Feldzuordnungen, usw.).
- ❖ Dabei können Sie jederzeit auf die Schaltfläche **Speichern** klicken. AssetCenter gibt Ihnen damit die Möglichkeit, die neuen Bedingungen zu speichern.

Import einer Datenbank oder mehrerer Textdateien

- ❖ Verwenden Sie das Menü **Datei/ Import**, um den Import einer Datenbank zu wählen.
- ❖ Geben Sie die erforderlichen Informationen auf der Registerkarte **ODBC** oder **Text** ein. Klicken Sie auf **Öffnen**.
- ❖ Öffnen Sie das Skript über das Menü **Datei/ Skript öffnen**.
- ❖ Ändern Sie die Importparameter, z. B. den Speicherort der Datenbank bzw. der Textdateien, die Zuordnung zwischen Feldern usw.).
- ❖ Dabei können Sie jederzeit die neuen Bedingungen über das Menü **Datei/ Speichern** oder **Datei/ Speichern unter** speichern.

Ein neues Skript verfassen, während Sie die Importbedingungen definieren

Import einer einzelnen Textdatei

Klicken Sie auf die Schaltfläche **Schließen** und fahren Sie fort, wie beim Speichern eines neuen Skripts.

Import einer Datenbank bzw. mehrerer Textdateien

Verwenden Sie das Menü **Datei/ Neues Skript**. AssetCenter lässt Sie die aktuellen Parameter vor dem Löschen in Form eines Skripts speichern.

Skript ausführen

Gehen Sie zum Ausführen eines bereits bestehenden Skripts wie folgt vor:

- Verwenden Sie das Menü **Datei/ Import**.
- Klicken Sie auf das Symbol .
- Geben Sie den Namen der Skriptdatei an.
- Starten Sie den Import.

Sie können jedoch auch wie beim Ändern eines Skripts vorgehen und den Import über die Bildschirme zur Definition der Importparameter starten.

Importskript über eine 32-Bit-Eingabeaufforderung ausführen

Warnung: Dieser Vorgang ist nur in der Befehlszeile (DOS-Fenster) unter einer 32-Bit-Version von Windows, z. B. Windows NT oder 95, möglich.

Prinzip

Um ein DOS-Programm online ausführen zu können, müssen Sie zuvor mit Hilfe des Windows-Importmoduls ein Importskript erstellt haben.

Anschließend können Sie den Importbefehl mit Hilfe des Programms **AmImpl32.exe** entweder manuell oder automatisch, beispielsweise über eine Batch-Datei, ausführen. Das Programm **Bin32** befindet sich im AssetCenter-Programmverzeichnis und kann nur in einer 32-Bit-Umgebung ausgeführt werden.

Syntax

```
amimpl32 [-verbose] [-?|h|H] -src:<cnx> [-srcpass:<password>] -dst:<cnx> [-dstlogin:<login>] [-dstpass:<password>] [-log:<file>]
```

-verbose: Anzeigen von Meldungen im Verlauf des Imports, standardmäßig aktiv

-, -h oder -H: Anzeigen der Hilfe über das Programm

- src: Dieser Parameter zeigt unterschiedliche Informationen an:
- ❖ Pfad und Name des auszuführenden Importskripts
 - ❖ Name der Verbindung der insgesamt zu importierenden AssetCenter-Datenbank, wie er im Menü **Datei/ Verbindungen bearbeiten**) erscheint
 - ❖ Name der AssetCenter-Datenbank ohne Verbindung:
[<NameDerEngine>;<StandortDerDatenbank>;<Benutzer>;<Kennwort>]
- In diesem Fall geben Sie die Informationen in den nachstehenden Feldern wie folgt ein <>:

	Oracle	MS SQL Server	Sybase SQL Anywhere	Sybase SQL Server
NameDerEngine	Oracle	ODBC	ODBC	Sybase
Standort DerDatenbank	Name des Servers	Name der Datenquelle	Name der Datenquelle	NameDesServers:NameDerDatenbank
Benutzer	Name des Kontos	Name des MS SQL Server-Benutzers	Name des Sybase SQL Anywhere-Benutzers	Name des Kontos
Kennwort	Kennwort des Kontos	Kennwort des MS SQL Server-Benutzers	Kennwort des Sybase SQL Anywhere-Benutzers	Kennwort des Kontos

- srcpass: Kennwort der zu importierenden Ausgangsdatenbank. Im Fall einer AssetCenter-Datenbank handelt es sich um das Kennwort des Kontos **Admin**.
- dst: Name der Verbindung zur AssetCenter-Datenbank, in die die Daten importiert werden sollen. Der Name steht im Feld **Name** des Menüs **Datei/ Verbindungen bearbeiten**).
- dstlogin: Login des Administrators der AssetCenter-Datenbank, in die die Daten importiert werden sollen (**Admin** oder AssetCenter-Benutzer mit den entsprechenden Zugriffsberechtigungen).
- dstpass: Kennwort, das dem in **dstlogin** angegebenen Login zugeordnet ist.
- log: Vollständiger Pfad zur Datei mit der Importhistorie.

Hinweis: Wenn die Zeichenfolgen zwischen den Klammern <> Leerstellen aufweisen, müssen Sie die Folgen in Anführungszeichen setzen (").

Beispiel:

```
amimpl32 -verbose -src:Person.scr -srcpass:Kennwort -dst:AllgDatenbank  
-dstlogin:Gerald -dstpass:Kennwort -log:"Meine Protokolldatei.txt"
```


Kapitel 16 - Export von Daten und Verwaltung von SQL-Ansichten

In diesem Kapitel wird der Export von Daten und die Verwaltung der SQL-Ansichten der Datenbank mit AssetCenter erklärt.

Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

- Definitionen eines Skripts und einer Exportabfrage
- Daten der AssetCenter-Datenbank exportieren
- SQL-Ansichten der AssetCenter-Datenbank verwalten
- Empfehlungen
- Exportskripte definieren
- Exportskripte ausführen

Definitionen eines Skripts und einer Exportabfrage

Exportskript

Mit Hilfe der Exportskripte können Sie Daten exportieren oder SQL-Ansichten über AssetCenter Export oder **amexpl32.exe** erstellen bzw. löschen. Die definierten Exportskripte können zur Wiederverwendung gespeichert werden.

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

Es 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.

Exportabfragen

Die Definition der Exportabfragen erfolgt mit Hilfe von AssetCenter Export.

Eine Exportabfrage wird mit Hilfe der nachstehenden Elemente definiert:

- Ein Name
- Gegebenenfalls eine Exportdatei (im Exportmodus)
- Ein Kommentar, der nicht exportiert wird
- Eine Ausgangstabelle
- Eine Liste mit den Spalten, die extrahiert werden sollen (Felder, Verknüpfungen, Merkmale und berechnete Felder in der Ausgangstabelle) sowie den zugeordneten Sortierkriterien
- Ein Filter mit der Klausel WHERE, der die Extraktionsbedingungen festlegt
- Ein Filter mit der Klausel HAVING, der die Extraktionsbedingungen festlegt
- Eine Klartextanzeige der Abfrage, die den Registerkarten **Filter (WHERE-Klausel)** und der Klausel HAVING entspricht
- Eine 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 Menü **Datei/ Liste exportieren**. Dieses Menü wird angezeigt, wenn wenigstens eine Liste oder eine Registerkartenliste angezeigt ist, und ermöglicht das Exportieren der aktiven Liste.

Daten über ein Exportskript exportieren

- ↪ Starten Sie AssetCenter Export. Dieses Programm befindet sich in den Programmen des Menüs **Start** bzw. in der AssetCenter-Programmgruppe.
- ↪ Definieren Sie ein Exportskript im Modus **Daten exportieren**:
 - ❖ Schreiben Sie auf der Registerkarte **Abfragen** die Abfrage, mit der Sie die Daten definieren, die exportiert werden sollen.
 - ❖ Geben Sie auf der Registerkarte **Optionen** das Format der Textdateien an, in die die Daten exportiert werden sollen.
 - ❖ Speichern Sie das Exportskript über das Menü **Datei/ Skript speichern** oder das Menü **Datei/ Skript speichern unter**.
- ↪ Führen Sie das Exportskript aus wie folgt:
 - ❖ Entweder direkt in AssetCenter Export, oder
 - ❖ Durch Starten des Programms **amexpl32.exe** unter DOS. Dieses Programm ist nur in einer 32-Bit-Umgebung verfügbar.

Hinweis: Um die Kohärenz der in AssetCenter definierten Zugriffsbeschränkungen aufrechtzuerhalten, ist das Starten von AssetCenter Export bzw. das Ausführen des Programms **amexpl32.exe** dem Administrator (Login **Admin** bzw. einem Benutzer mit den entsprechenden administrativen Rechten) vorbehalten.

Daten über das Menü **Datei/ Liste exportieren** exportieren

Das Menü **Datei/ Liste exportieren** steht jedem AssetCenter-Benutzer zur Verfügung. Der Benutzer kann damit alle Daten exportieren, die er auch auf den Bildschirm rufen darf.

- ↪ 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.
- ↪ Wählen Sie das Menü **Datei/ Liste exportieren**.
- ↪ Geben Sie die erforderlichen Daten im Fenster ein, und klicken Sie dann auf die Schaltfläche .

Hinweis: Weitere Informationen zum Menü **Datei/ Liste exportieren** finden Sie im Handbuch "Referenzhandbuch: Allgemeine Ergonomie und Referenztabelle", Kapitel "Arbeiten 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 in der AssetCenter-Datenbank. Diese Ansichten können anschließend von externen Programmen bearbeitet werden, die anstelle der Textdateien eher Ansichten verwenden.

Warnung: 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:

- ↗ Starten Sie AssetCenter Export.
- ↗ Definieren Sie ein Exportskript im Modus **SQL-Ansichten erstellen/ entfernen**:
 - ❖ Schreiben Sie in die Registerkarte **Abfragen** die Abfragen, mit denen die zu extrahierenden Daten definiert werden.
 - ❖ 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.
 - ❖ Speichern Sie das Exportskript.
- ↗ Führen Sie das Exportskript wie folgt aus:
 - ❖ Entweder direkt in AssetCenter Export, oder
 - ❖ Durch Starten des Programms **amexpl32.exe** unter DOS. Dieses Programm ist nur in einer 32-Bit-Umgebung verfügbar.

Empfehlungen

Wir empfehlen Ihnen, den Identifizierungsschlüssel der Tabelle nicht als Abstimmungsschlüssel zu verwenden, wenn Sie die zuvor exportierte 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 kann auf zwei Arten verwendet werden:

- Im Exportmodus zum Exportieren von Daten
- Im Ansichtenmodus zum Erstellen oder Löschen von SQL-Ansichten der Datenbank

Es 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.

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:

- ↳ Starten Sie AssetCenter Export.

- ↵ 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.
- ↵ 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.
- ↵ Legen Sie in der oberen Hälfte des Bildschirms von AssetCenter Export fest, ob Sie Daten exportieren (**Daten exportieren**) oder SQL-Ansichten der Datenbank verwalten (**SQL-Ansichten erstellen/entfernen**) möchten.
- ↵ Schreiben Sie die Abfragen für das Exportskript in die Registerkarte **Abfragen**.
- ↵ Wenn Sie Daten exportieren, müssen Sie das Ausgabeformat für die exportierten Daten auf der Registerkarte **Optionen** festlegen.
- ↵ Wenn Sie SQL-Ansichten verwalten, definieren Sie die Vorgänge, die Sie auf der Registerkarte **Ansichten** ausführen möchten.
- ↵ 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 , um eine Exportabfrage hinzuzufügen.
- Klicken Sie auf die Schaltfläche , um die ausgewählten Exportabfragen zu löschen.

Abfragen für ein Exportskript erstellen

- Klicken Sie auf der Registerkarte **Abfragen** auf die Schaltfläche .
- Definieren Sie den Namen der Abfrage. Dieser Name wird in dem Ausführungsbericht der Registerkarte **Meldungen** im Detailfenster des Exportskripts verwendet.
- Sie können einen Kommentar eingeben, der nicht exportiert wird.
- Die Definition der Daten, die extrahiert werden sollen, erfolgt im Feld **Abfrage**.

- 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 Ausgabertextdatei ein, in die die von der Abfrage ausgewählten Daten exportiert werden sollen. Ein Exportskript, das mehrere Exportabfragen enthält, generiert damit mehrere Textdateien.

Hinweis: Das Feld **Datei** ist nicht aktiv, 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 exportiert werden sollen, 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(lAstId) FROM amAsset 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:

- ❖ Wählen Sie einen Index im Feld **Nach Index**, um einen Sortiervorgang anhand des Indizes zu definieren.
- ❖ Ansonsten markieren Sie die Kontrollkästchen **Sortieren** in der gewünschten Sortierreihenfolge.

Hinweis: 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 Verwendung für fortgeschrittene Benutzer**", 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 amAsset
```

Beispiel mit markiertem Kästchen **Nur eindeutige Datensätze**:

```
SELECT DISTINCT Brand FROM amAsset
```

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, definieren Sie das Format der Ausgabertextdateien auf der Registerkarte **Optionen**. Dieses Format gilt für sämtliche Exportabfragen.

Hinweis: Wenn Sie sich für das Entfernen, 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 (TT), Monat (MM) und Jahr (JJ).

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 die exportierten Stunden mit Sekunden angezeigt werden sollen.

SQL-Ansichten bearbeiten

Wenn Sie bestimmte, den Exportabfragen entsprechende SQL-Ansichten entfernen oder (neu) erstellen möchten, müssen Sie die Aktionen, die dabei ausgeführt werden sollen, auf der Registerkarte **Ansichten** definieren.

Hinweis: 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 Ansichtsbearbeitung** 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 Entfernen von Ansichten (Befehl **DROP VIEW**) zu generieren. Gehen Sie dann wie folgt vor:
 - ❖ 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 .
 - ❖ Wählen Sie das Trennzeichen für die SQL-Befehle unter ; (Oracle-Datenbanken) und **GO** (für alle anderen Datenbanken).

Exportskripte ausführen

Zum Exportieren von Daten oder 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:

- ↳ Starten Sie AssetCenter Export.

- ↵ Definieren Sie Ihr Exportskript und speichern Sie es.
- ↵ 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 '%s' wurde erfolgreich abgeschlossen**. Sollte es zu einem Fehler kommen, erscheint die Meldung **Fehler beim Ausführen des Skripts (xx)**.

Jeder Meldung wird ein Symbol vorangestellt:

-  Allgemeine Information
-  Fehler
-  Erfolgreicher Export
-  Warnung

Durch Klicken auf die Schaltfläche  brechen Sie den Export ab.

Exportskripte unter DOS ausführen

Prinzip

Vor dem Ausführen des Online-DOS-Programms, müssen Sie zunächst ein Exportskript mit Hilfe von AssetCenter Export erstellen.

Anschließend können Sie einen Exportbefehl manuell eingeben oder, beispielsweise über eine Batch-Datei, mit Hilfe des Programms **AmExpl32.exe** automatisch ausführen. Dieses Programm befindet sich im Unterverzeichnis **Bin32** des AssetCenter-Installationsverzeichnisses.

Dieses Programm steht nur in 32-Bit-Umgebungen zur Verfügung.

Syntax

```
amexpl32 [-verbose] [-?|h|H] -script:<Skript> -
cnx:<Verbindung> [-login:<Login>] [-password:<Kennwort>]
-verbose: Anzeigen der Meldungen im Verlauf des Exports
-?, -h oder -H: Anzeigen der Hilfmeldungen zum Programm
-script: Pfad und Name des auszuführenden Exportskripts
```

-cnx: 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 -cnx:AllgDatenbank -login:Gerald -password:Kennwort`

Kapitel 17 - Arbeiten mit Skripten

In diesem Kapitel wird das Arbeiten mit Skripten erklärt. Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

Einführung

- Definition eines Skripts
- Anwendungsbereiche von Skripten
- Einführung in die Funktionen
- Klassifizierung der BASIC-Funktionen

Schreiben von Skripten

- Erste Schritte zum Schreiben eines Skripts
- Vorsichtsmaßnahmen und Tipps

Beispiele

- Erstes Beispiel für ein Skript
- Zweites Beispiel für ein Skript

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 Definieren der Eigenschaften von Objekten in 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 auf den folgenden Seiten 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 Verwendung für fortgeschrittene Benutzer", Kapitel "Import von Daten" sowie im Handbuch "Referenzhandbuch: Verwaltung und Verwendung für fortgeschrittene Benutzer", Kapitel "Export von Daten und Verwaltung von SQL-Ansichten".
- Gemischte Skripte (Prozedur und Beschreibung). Diese Art von Skripten kommt in den AssetCenter-Assistenten zur Anwendung.

Information zu BASIC

Das in AssetCenter verwendete BASIC-Programm besteht aus mehreren Komponenten, die mit Visual Basic for Applications™ von Cypress kompatibel sind. Wir empfehlen Ihnen, zusätzliche Informationen zur Sprache, Struktur und Syntax dieses BASIC-Programms im Handbuch des entsprechenden BASIC-Programms nachzulesen.

So werden beispielsweise nur bestimmte Funktionen von Visual Basic for Applications™ unterstützt. Dazu gehören:

- Funktionen zum Zugriff auf nicht unterstützte Dateien.

- Insbesondere unter UNIX werden die Funktionen für Datum und Uhrzeit nur begrenzt unterstützt.
- Die Steuerelemente von Visual Basic for Applications™ sind nicht verfügbar.

Schreibweise für den Zugriff auf Daten

Die Syntax des von AssetCenter verwendeten BASIC-Programms ähnelt der Standardsyntax, mit Ausnahme der Schreibweisen beim Zugriff auf die Daten über einen aktuellen Datensatz. In diesem Fall gilt:

[Verknüpfung.Verknüpfung.Feld]

Beispiel: [Category.FullName] über die Tabelle der Produkte.

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]

Anwendungsbereiche von Skripten

AssetCenter bietet Ihnen die Möglichkeit der Verwendung von BASIC-Skripten in nachstehenden Bereichen:

- Zum Konfigurieren folgender Felder (Kontextmenü **Objekt konfigurieren** oder AssetCenter Database Administrator):
 - ❖ **Standardwert.**
 - ❖ **Obligatorisch**
 - ❖ **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**.

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 bereits geschrieben wurden. Diese Programme führen die verschiedensten Aufgaben durch (Berechnungen, Konvertierung der vom Benutzer gelieferten Informationen) und geben ein entsprechendes Ergebnis zurück. Der Benutzer muss die Funktion nur anhand ihres Namens aufrufen, und die zur Rückgabe eines Ergebnisses eventuell 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, sind:
 - ❖ Die Währung A
 - ❖ Die Währung B
 - ❖ Der umzurechnende 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

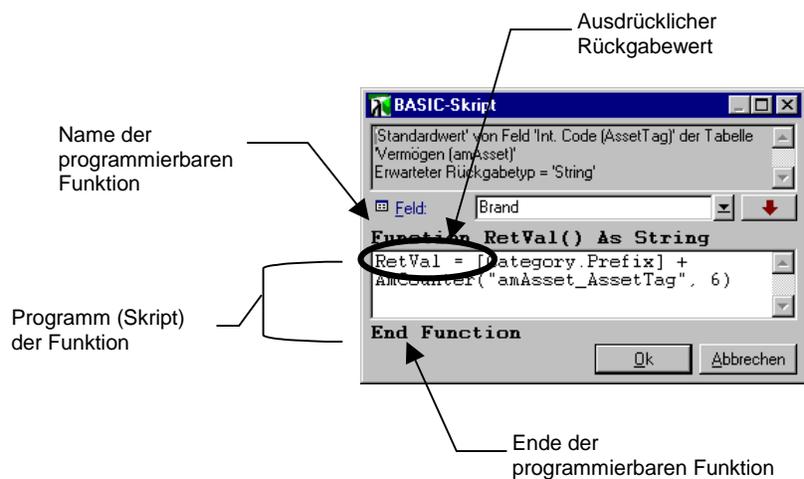
Die programmierbaren Funktionen sind Programme, die der Benutzer selbst schreiben muss. Dabei gibt der Benutzer ausdrücklich den Wert an, der in der Variablen **RetVal**, auch Rückgabewert genannt, von der

programmierbaren Funktion in der nachstehenden Form zurückgegeben werden muss:

```
RetVal=<Ausdruck>
```

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:



Der Skripteditor

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 Vermögensgegenstände 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. Wir empfehlen Ihnen, dieser Information besondere Beachtung zu schenken, da sie beim Kompilieren und Ausführen eines BASIC-Skripts zu Fehlern führen 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 **Datum** bzw. **Datum+Uhrzeit** 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 dementsprechend nicht kompatibel mit dem Format eines Felds vom Typ **Datum**. 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:

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**:

- Wenn es sich um eine Funktion vom Typ **Boolean** (Boolesch) handelt, kann sie nur die Werte **TRUE** oder **FALSE** annehmen. Jeder andere Wert führt beim Kompilieren 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.

Wenn Sie den Typ der Funktion nicht beachten, kann es beim Kompilieren 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-Programm, das zum Schreiben von Skripten verwendet wird, greift auf Funktionen unterschiedlicher Klassen zurück:

- Herkömmliche BASIC-Funktionen, die dem Standard **Visual BASIC for Applications**™ 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 praktischen Beispiels:

- 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 die Produkte verfügbar sein, deren **Kategorie** den Wert **Großrechner/ Systemplatine** aufweist.

Vorgehensweise

Ordnen Sie dem Feld **Verfügbar** (SQL-Name: seAvailable) in den Parametern des Merkmals **Lernprogramm** ein BASIC-Skript zu.

Schritt 1 – Erstellen des Merkmals *Lernprogramm*

Öffnen Sie den Bildschirm zur Erstellung eines Merkmals über das Menü **Extras/ Administration/ Merkmale**, und klicken Sie dann auf die Schaltfläche **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
Bez. (SQL-Name: TextLabel)	Lernprogramm
SQL-Name (SQL-Name: SQLName)	Lernprogramm (In diesem Beispiel nicht verwendet)
Eingabeart (SQL-Name: seDataType)	Numerisch

Klicken Sie auf **Erstellen**, um das Merkmal zu erstellen.

Positionieren Sie den Cursor auf der 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.

Parameter zu Merkmal 'Lernprogramm (Lernprogramm)' hinzufügen

Tabelle: Produkte (amProduct)

Merkmal: Lernprogramm (Lernprogramm)

Einstellungen Standardwerte Hilfe Historie

Eingabe

Verfügbar: (Skript)

Anzeige erzwingen: Nein

Obligatorisch: Nein

Historie anlegen: Nein

Historie auch bei Erstellen von Hauptdatensatz anlegen

Hinzufügen Hinzuf. Abbrechen

Parameter des Merkmals **Lernprogramm**

Schritt 2 – Öffnen des Bearbeitungsfensters

Wählen Sie auf der Registerkarte **Einstellungen** des Detailbildschirms 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:

- Ändern des Werts im Feld **Verfügbar** (SQL-Name: seAvailable) in **Ja**, wenn die Kategorie des Produkts **/Rechner/Systemplatine** entspricht.
- In allen anderen Fällen, Aufrechterhalten des Werts **Nein** im Feld **Verfügbar**.

Unser Algorithmus schreibt sich wie folgt:

```
Wenn die Kategorie des Produkts "/Rechner/Systemplatine" entspricht,  
dann ist  
Das Merkmal verfügbar  
Ansonsten ist  
Das Merkmal nicht verfügbar
```

Damit bestimmt der Wert des Felds **Kategorie** (SQL-Name: Category) der Produkttabelle den Wert des Felds **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 **Kategorie** (SQL-Name: Category) über die Dropdown-Liste im Bearbeitungsfenster.

Bei dem Feld **Kategorie** handelt es sich um ein Feld vom Typ **Verknüpfung**.

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. Das Programm entspricht dem in

Schritt 3 geschriebenen Algorithmus, nur das es hier in BASIC übersetzt wird.

```
If [Category]="/Rechner/ Systemplatine" Then
  RetVal=True
Else
  RetVal=False
End If
```

Die Groß- und Kleinschreibung hat keinerlei Auswirkung auf die Bearbeitung und Kompilierung des Skripts.

Klicken Sie auf , um das Skript zu bestätigen.

Schritt 5 – Testen des BASIC-Programms

Mit diesem Kontrollschritt vergewissern Sie sich der einwandfreien Funktionsweise des Skripts.

- Wählen Sie das Menü **Ausrüstung/ Produkte**, um die Tabelle der Produkte zu öffnen, und erstellen Sie ein neues Produkt durch Klicken auf die Schaltfläche .
- Füllen Sie nur die obligatorischen Felder aus:
 - ❖ **Kategorie** (SQL-Name: Category) mit dem Wert **Rechner/Systemplatine**
 - ❖ **Modell** (SQL-Name: Model) und **Strichcode** (SQL-Name: BarCode) mit einem beliebigen Wert.
- Erstellen Sie das neue Produkt durch Klicken auf .
- Wählen Sie jetzt die Registerkarte **Merkmale** und fügen Sie durch Klicken auf die Schaltfläche ein Merkmal hinzu. Im Auswahlbildschirm erscheint der Merkmalname, für den Sie das Skript bearbeitet haben.
- Ändern Sie jetzt den Wert der Kategorie des Produkts in **/Großrechner/** und bestätigen Sie diese Änderung durch Klicken auf .
- Wählen Sie die Registerkarte **Merkmale** im Detailfenster des Produkts und klicken Sie auf die Schaltfläche , um ein Produkt hinzuzufügen. Der Name des Merkmals, für das Sie das Skript bearbeitet haben, erscheint nicht mehr im Auswahlfenster.

Das Skript erfüllt seine Funktion.

Vorsichtsmaßnahmen und Tipps

Im den folgenden Abschnitt finden Sie einige Vorsichtsmaßnahmen und Tipps, die Sie zum schnellen und effizienten Schreiben Ihrer Berechnungskripte unbedingt beachten sollten.

- Vorsichtsmaßnahmen beim Arbeiten mit programmierbaren Funktionen
- Format der Konstanten vom Typ Datum+Uhrzeit in den Skripten
- Format der Konstanten vom Typ Dauer in den Skripten
- Lese- und Schreibzugriff auf den Wert einer Systemaufzählung
- Die virtuelle Verknüpfung CurrentUser
- Kommentar zu einem BASIC-Skript eingeben
- Fehlermeldungen auslösen

Vorsichtsmaßnahmen beim Arbeiten mit programmierbaren Funktionen

Im Folgenden finden Sie einige Vorsichtsmaßnahmen, 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="1998/07/12 13:05:00"
```

Der Bindestrich (-) kann ebenfalls als Trennzeichen für Datumsangaben verwendet werden.

BASIC-Datum und UNIX-Datum

Daten werden in BASIC und unter UNIX unterschiedlich ausgedrückt:

- In BASIC kann ein Datum entweder im internationalen Format oder in Form einer Gleitkommazahl vom Typ **Double** 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 Stunden entspricht.
- Unter UNIX werden die Daten in Form einer ganzen langen Zahl vom Type **Long** ausgedrückt, mit der die Anzahl der Sekunden dargestellt wird, die seit dem 01/01/1870 um Mitternacht vergangen sind. Den Zeitzonen (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 **Dauer** 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 **AnWorkTimeSpanBetween()**, geben ein Ergebnis in Sekunden aus.

Bei Konvertierungen geht AssetCenter davon aus, dass ein Jahr 12 Monaten und ein Monat 30 Tagen entspricht (Also gilt: 1 Jahr = 360 Tage).

Lese- 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 Rahmen **Standardzuordnung** der Registerkarte **Allgemein** im Detailfenster der Vermögensgegenstände.

In der nachstehenden Tabelle finden Sie die Werte der Aufzählung:

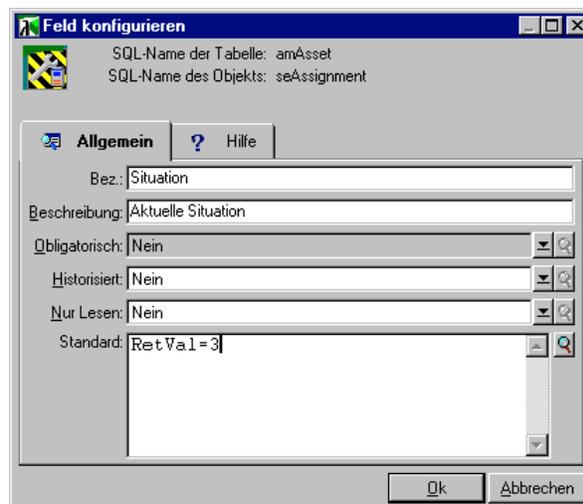
Werte der Aufzählung	Dem Wert zugeordnete ganze Zahl
In Betrieb	0
Auf Lager	1
Abgang	2
Ausstehende Lieferung	3

Zum Definieren des Standardwerts einer Aufzählung brauchen Sie dann nur noch Folgendes zu tun:

- Identifizieren Sie die ganze Zahl, die dem gewünschten Wert entspricht.
- 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:



Bearbeitung des Standardwerts einer Aufzählung

Eine Systemaufzählung ist nicht mit einer vom Benutzer definierten geschlossenen Aufzählung zu verwechseln.

Die vollständige Liste der Werte der Systemaufzählungen finden Sie in der Datei **Database.txt**, die im Unterverzeichnis **Infos** des AssetCenter-Installationsverzeichnisses gespeichert ist. Die Spalten **Data display and entry type** und **Additional information on data display and entry type** beschreiben den Typ der Aufzählung sowie die für die Aufzählung möglichen Werte.

Die virtuelle Verknüpfung *CurrentUser*

Definition

CurrentUser ist eine Verknüpfung, der von allen Tabellen ausgehend auf den Datensatz der Tabelle der Abteilungen und Personen zeigt, der dem aktuellen Benutzer entspricht.

- In der Form **CurrentUser** zeigt die Verknüpfung auf den Datensatz mit dem aktuellen Benutzer und gibt die Feld-Id zurück.
- In der Form **CurrentUser.<SQL-Name des Felds>** gibt die Verknüpfung den Wert des Felds für den aktuellen Benutzer zurück.

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]**

Kommentar zu einem BASIC-Skript eingeben

In bestimmten Fällen kann es sich durchaus als hilfreich erweisen, einen Kommentar zu einem BASIC-Skript einzugeben, um die vom Skript ausgeführten Operationen im Klartext zu beschreiben, oder um einem Benutzer das Verstehen und Ändern des Skripts zu erleichtern. AssetCenter kennzeichnet den Kommentar durch Verwendung eines Apostrophs ('). Alle Zeichen, die auf das Apostroph folgen und auf derselben Zeile stehen, werden vom Compiler als Kommentar erkannt und ignoriert. Dabei sind zwei Situationen zu unterscheiden:

- Der Kommentar füllt die gesamte Zeile des BASIC-Skripts aus, wie im nachstehenden Beispiel zu sehen ist. Sie brauchen keinerlei Vorsichtsmaßnahmen zu beachten.

```
' Wir testen hier den Wert des Felds BarCode der Vermögenstabelle
' 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 hinzugefügt, die vom BASIC-Compiler interpretiert werden muss. In diesem Fall müssen Sie das Zeichen : verwenden, um das Skript vom Kommentar zu trennen. 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>)
```

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 beschreiben wir Ihnen ein Problem, das durch die Erstellung eines BASIC-Skripts gelöst werden kann. Versuchen Sie, die richtige Antwort auf dieses Problem zu finden, bevor Sie sich die Lösung ansehen.

- Darstellung des Problems
- Schritt 1 – Analysieren und Schreiben des Algorithmus
- Schritt 2 – Schreiben des BASIC-Skripts
- Schritt 3 – Testen des BASIC-Programms

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 Merkmals 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:



Parameter des Merkmals **Beispiel1**

Schritt 1 – Analysieren und Schreiben des Algorithmus

Der Algorithmus muss nachstehende Aufgaben erfüllen:

- Ändern des Werts im Feld **Obligatorisch** (SQL-Name: seMandatory) (SQL-Name: seMandatory) auf **Ja**, wenn die Intervention abgeschlossen ist.
- In allen anderen Fällen, Aufrechterhalten 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) **Abgeschlossen** lautet.

Damit bestimmt der Wert im Feld **Status** (SQL-Name: seStatus) der Interventionstabelle den Wert im Feld **Obligatorisch** des Merkmals. In unserem Algorithmus kommt nur dieses Feld zur Anwendung.

Das Feld **Status** (SQL-Name: seStatus) in der Interventionstabelle finden Sie über die Dropdown-Liste im Bearbeitungsfenster.

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:

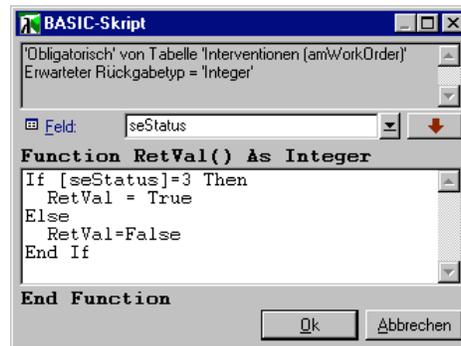
Wert der Aufzählung	Diesem Wert zugeordnete ganze Zahl
Benachrichtigt	0
Geplant	1
In Arbeit	2
Abgeschlossen	3

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.



Bearbeitung des Programms

Klicken Sie auf , um das Skript zu bestätigen.

Schritt 3 – Testen des BASIC-Programms

Mit diesem Kontrollschritt vergewissern Sie sich der einwandfreien Funktionsweise des Skripts.

- Wählen Sie das Menü **Support/ Interventionen**, um die Interventionstabelle zu öffnen und wählen Sie eine Intervention mit dem Status **Abgeschlossen**. Wenn keine abgeschlossene Intervention vorhanden sein sollte, können Sie sie auch erstellen.
- Wählen Sie die Registerkarte **Merkmale**. AssetCenter hat ein Merkmal zu der betroffenen Intervention hinzugefügt und seine Definition ist obligatorisch.
- Wählen Sie jetzt eine Intervention, für die der Status nicht den Wert **Abgeschlossen** aufweist, und gehen Sie in die Registerkarte **Merkmale** dieser Intervention. Das Merkmal **Beispiel** wird angezeigt, wobei die Definition jedoch nicht obligatorisch ist.

Das Skript erfüllt seine Funktion.

Zweites Beispiel für ein Skript

In diesem Abschnitt beschreiben wir Ihnen ein Problem, das durch die Erstellung eines BASIC-Skripts gelöst werden kann. Versuchen Sie, die richtige Antwort auf dieses Problem zu finden, bevor Sie sich die Lösung ansehen.

- Darstellung des Problems

- Schritt 1 - Analysieren und Schreiben des Algorithmus
- Schritt 2 – Schreiben des BASIC-Skripts
- Schritt 3 – Testen des BASIC-Programms

Darstellung des Problems

Wir möchten, dass im Feld **Feld 1** (SQL-Name: Field1) des Personendetails gegebenenfalls Name und Vorname der Person erscheinen. Wenn kein Vorname vorhanden ist, reicht auch der Nachname.

Schritt 1 - Analysieren und Schreiben des Algorithmus

Der Algorithmus muss nachstehende Aufgaben erfüllen:

- Standardmäßiges Anzeigen von Namen und Vornamen der Person im Feld **Feld 1** (SQL-Name: Field1) eines Personendetails, sofern diese Angaben existieren.
- Standardmäßiges Anzeigen des Nachnamens der Person im Feld **Feld 1** 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"
```

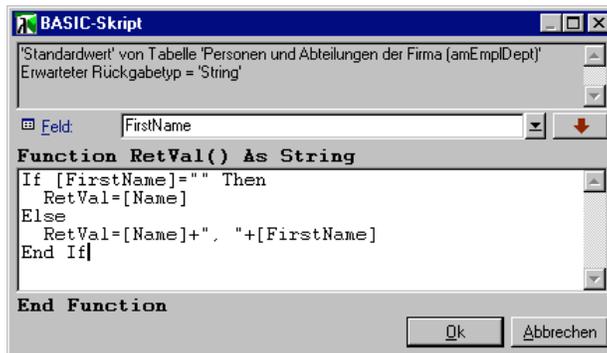
Damit bestimmt der Wert in den Feldern **Name** (SQL-Name: Name) und **Vorname** (SQL-Name: FirstName) der Personentabelle den Standardwert für das Feld **Feld 1** im Detailfenster einer Person. In unserem Algorithmus kommen nur diese beiden Felder zur Anwendung.

Jetzt brauchen Sie nur noch mit der rechten Maustaste im Personendetail auf das Feld **Feld 1** zu klicken 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 vorgesehenen Fenster übersetzt werden.



Bearbeitung des Programms

Klicken Sie auf , um das Skript zu bestätigen.

Schritt 3 – Testen des BASIC-Programms

Mit diesem Kontrollschritt vergewissern Sie sich der einwandfreien Funktionsweise des Skripts.

- Wählen Sie das Menü **Ausrüstung/ Abteilungen und Personen**, um die Tabelle mit den Personen zu öffnen, und erstellen Sie eine neue Person.
- Geben Sie die erforderlichen Informationen in den Feldern **Name** (SQL-Name: Name) und **Vorname** (SQL-Name: FirstName) ein, und bestätigen Sie ihre Erstellung durch Klicken auf . AssetCenter zeigt im Feld **Feld 1** (SQL-Name: Field1) Namen und Vornamen der Person an.

Das Skript erfüllt seine Funktion.

Kapitel 18 - Verwaltung von Kalendern

Der Zugriff auf die Kalender erfolgt über das Menü Extras/ Kalender.

In diesem Kapitel wird die Definition und Verwaltung von Kalendern beschrieben. Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

Einführung

- Allgemeine Übersicht und Funktionsweise eines Kalenders
- Auswirkung der Kalender auf bestimmte Funktionalitäten

Erstellung eines Kalenders

- Vorgehensweise beim Erstellen von Kalendern
- Beschreibung der Erstellung eines Kalenders

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 allgemein gültigen 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.

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

Bei der Verwendung von Kalendern definieren Sie die Perioden, die gearbeitet werden.

Gehen Sie zum Erstellen eines Kalenders wie folgt vor:

- ↪ Geben Sie dem Kalender einen eindeutigen **Name** (SQL-Namen: Name).
- ↪ Ordnen Sie dem Kalender gegebenenfalls eine Zeitzone zu, und geben Sie dazu die erforderlichen Informationen im Feld **Zeitzone** (SQL-Namen: TimeZone) ein.
- ↪ Definieren Sie die wöchentlichen Arbeitsstunden auf der Registerkarte **Arbeitszeiten** des Kalenderdetails.
- ↪ Definieren Sie die Ausnahmen auf der Registerkarte **Ausnahmen** des Kalenderdetails.
- ↪ 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:

- Allgemeine Informationen zur Eingabe des Kalenders
- Eingaben auf der Registerkarte Arbeitszeiten
- Eingaben auf der Registerkarte Ausnahmen eines Kalenders
- 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.

Zusätzlich 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.

Die Registerkarte **Arbeitszeiten** im Kalenderdetail

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.
 - ❖ Klicken Sie mit der linken Maustaste auf den Regler, und zwar dort, wo der Zeitbereich beginnen soll.
 - ❖ 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.
 - ❖ 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.

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:

- ↪ Geben Sie der Ausnahme einen Namen.
- ↪ 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.
- ↪ 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*



Registerkarte *Ausnahmen* im Kalenderdetail

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.

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
Taglich	Die Ausnahme gilt fur alle Tage des Jahres.
Tag des Jahres	Die Ausnahme gilt fur einen bestimmten Tag oder eine Auswahl von Tagen, die mit Hilfe der Kontrollkastchen Tag , Monat und Jahr festgelegt werden.
Am ersten	Die Ausnahme gilt fur einen Tag in der Woche, der mit Hilfe des Kontrollkastchens Tag festgelegt wird, und fur Monate und Jahre, die mit Hilfe der Kontrollkastchen Monat und Jahr festgelegt werden. Beispiel: Am ersten Freitag eines jeden Monats.
Am zweiten	Die Ausnahme gilt fur einen Tag in der Woche, der mit Hilfe des Kontrollkastchens Tag , festgelegt wird, und fur Monate und Jahre, die mit Hilfe der Kontrollkastchen Monat und Jahr festgelegt werden. Beispiel: Am zweiten Montag des Monats September.
Am vorletzten	Die Ausnahme gilt fur einen Tag in der Woche, der mit Hilfe des Kontrollkastchens Tag , und fur Monate und Jahre, die mit Hilfe der Kontrollkastchen Monat und Jahr festgelegt werden. Beispiel: Am vorletzten Mittwoch des Monats September.
Am letzten	Die Ausnahme gilt fur einen Tag in der Woche, der mit Hilfe des Kontrollkastchens Tag , und fur Monate und Jahre, die mit Hilfe der Kontrollkastchen Monat und Jahr festgelegt werden. Beispiel: Am letzten Dienstag eines jeden Monats im Jahr 1998.

Beispiel fur 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.

- ❖ Klicken Sie auf die Schaltfläche , um mit der Erstellung einer Ausnahme zu beginnen.
- ❖ 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.
- ❖ Sie beenden die Definition der Ausnahme mit der Zuordnung, in dem Sie dem Feld **Tag** den Wert **Am ersten** zuordnen.

Regel Nr. 2: Im August wird nur morgens von 8:30 Uhr bis 12:30 Uhr gearbeitet.

- ❖ Klicken Sie auf die Schaltfläche , um mit der Erstellung einer Ausnahme zu beginnen.
- ❖ 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**.
- ❖ Die Ausnahme gilt nur für eine Auswahl von Tagen (alle Tage im Monat August). Sie brauchen dem Feld **Tag** also nur noch den Wert **Tag des Jahres** zuzuordnen.
- ❖ 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**.

The screenshot shows a software window with three tabs: 'Arbeitszeiten', 'Ausnahmen', and 'Vorschau...'. The 'Vorschau...' tab is active. Below the tabs, there is a 'Test' section with three fields: 'Anfangsdatum' (29/06/99 15:23:53), 'Enddatum' (29/06/99 15:23:53), and 'Verstrichene Zeit' (0s). Below this is a section titled 'Dem Anfangsdatum zugeordneter Kalender'. It displays a calendar grid for the week of June 28 to July 4, 1999. The days are listed on the left, and the time slots (8:00-12:00 and 14:00-18:00) are shown on the right. The days from Monday to Friday are highlighted with blue bars, indicating working hours.

Registerkarte **Vorschau** des Kalenderdetails

- 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.

Sie können im Feld **Verstrichene Zeit** eine Dauer eingeben, um die erneute Berechnung des **Enddatum** unter Berücksichtigung des **Anfangsdatum** zu erzwingen.

Kapitel 19 - Verwaltung von Zeitzonen

In diesem Kapitel wird die Verwaltung von Zeitzonen erläutert.

Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

Einführung

- Vorteile der Verwaltung von Zeitzonen

Vorgehensweise

- Zeitzonen einrichten
- Zeitzonen erstellen
- Zeitzonen verwalten
- Verwaltung von Zeitzonen in AssetCenter Server

Auswirkungen

- Auswirkungen der Zeitzonen auf verschiedene Vorgänge

Vorteile der Verwaltung von Zeitzonen

Es kommt immer häufiger vor, dass sich Server- und Client-Stationen 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 **Datum+Uhrzeit** werden auf der Client-Station unter Berücksichtigung der für diese Station geltenden Zeitzone angezeigt.
- Alle Felder vom Typ **Datum+Uhrzeit** 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 beiden Client-Stationen stehen respektive in Paris und in London. Wir definieren die Zeitzone für die einzelnen Stationen und beziehen uns dabei auf den Meridian von Greenwich:

- Zeitzone der Server-Station = GMT-5
- Zeitzone der Client-Station in Paris = GMT+1
- Zeitzone der Client-Station in London = GMT
- Zeitzone der Daten = GMT+1

Alle Felder vom Typ **Datum+Uhrzeit** sind in der Datenbank des Servers im Format GMT+1 gespeichert und müssen auf der Client-Station in Paris im Format GMT+1 und auf der Client-Station in London im Format GMT erscheinen. Ein Beispiel: Sie nehmen einen Anruf auf der Client-Station in London an. Wenn für das Datum, bis zu dem die Support-Unterlage abgeschlossen sein muss, der 15.05.98 um 17:30 Uhr gilt, erscheint dieses Datum auf den anderen Stationen wie folgt:

- Auf dem Server: 15.05.98 um 12:30 Uhr
- Auf der Client-Station in Paris: 15.05.98 um 18:30 Uhr
- Auf der Client-Station in London: 15.05.98 um 17:30 Uhr

Zeitzone einrichten

Um eine optimale Verwaltung der Zeitzone durch AssetCenter zu gewährleisten, müssen Sie vorgehen wie folgt:

- Definieren Sie die Zeitzone bei der Erstellung der Datenbank über AssetCenter Database Administrator mit Hilfe der Option **Zeitzone verwenden**.
- Erstellen Sie die Zeitzone, beispielsweise durch Importieren von Informationen bezüglich der Zeitzone.
- Definieren Sie die für Ihren Rechner gültige Zeitzone über das Menü **Extras/ Optionen**.
- Definieren Sie die Kalender in Übereinstimmung mit den geltenden Zeitzone.

Zeitzone erstellen

Windows 95 und Windows NT sind in der Lage, eine Zeitzone und die damit zusammenhängenden Wechsel von der Winterzeit zur Sommerzeit, und umgekehrt, zu beschreiben. Diese Beschreibung wird von AssetCenter vervollständigt. So berücksichtigt AssetCenter beispielsweise die Regeln zur Änderung der Uhrzeiten über mehrere Jahre hinweg, und zeigt die lokale Uhrzeit dementsprechend besonders präzise an. 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
- An Standorten ohne Wechseln zwischen Sommer- und Winterzeit haben diese Informationen keinen weiteren Nutzen

Die 16-Bit-Versionen von Windows enthalten keine Beschreibung der Zeitzone.

Im Lieferumfang von AssetCenter ist eine Datei mit der Beschreibung der hauptsächlich verwendeten Zeitzonen enthalten, um Ihnen die manuelle Eingabe dieser Zonen zu ersparen. Gehen Sie zum Importieren dieser Datei wie folgt vor:

- Wählen Sie das Menü **Datei/ Import**. AssetCenter öffnet den Bildschirm zur Auswahl des Importtyps.
- Klicken Sie auf die Schaltfläche  (**Skript ausführen**). AssetCenter öffnet den Bildschirm zur Aktualisierung der Datenbank. Klicken Sie auf die Schaltfläche , um das Skript zu wählen, das Sie ausführen möchten. Dabei sollte es sich um das Skript **tz.scr** im Unterverzeichnis **datakit/standard** des AssetCenter-Installationsverzeichnisses handeln.
- Klicken Sie auf die Schaltfläche . AssetCenter führt den Import gemäß den im Skript enthaltenen Anweisungen durch.

Zeitzonen verwalten

In diesem Abschnitt gehen wir näher auf das Feld **SZ -> WZ** (SQL-Name: memDaylightInfo) ein:

- Format des Felds **SZ -> WZ**
- Werte für das Argument **<Year>**
- Werte für das Argument **<DaylightInfo>**
- Beispiel zur Verwaltung der Sommerzeit

Format des Felds **SZ -> WZ**

Die Struktur des Felds **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>.

	Das Argument <DaylightInfo> ist leer	Das Argument <DaylightInfo> weist einen Wert auf
Das Argument <Year> ist leer (<Year>= erscheint nicht)	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 <Year> definiert sind.
Das Argument <Year> verfügt über einen Wert	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 <Year> definiert wird, und das bis zum nächsten Argument <Year>

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 <StdShift> 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 <DDayPos> einen Wert zwischen 1 und 31 geben, um den Tag festzulegen, an der Wechsel von der Sommerzeit zur Winterzeit stattfinden soll.</p>
<SMonth>	<p>Monat des Wechsels von der</p>	<p>January</p>

Nebenargument	Beschreibung	Mögliche Werte
	Sommerzeit zur Winterzeit.	February March April November December
<SDayPos>	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 der der Wechsel von der Winterzeit zur Sommerzeit stattfinden soll.
<DMonth>	Monat des Wechsels von der Winterzeit zur Sommerzeit.	January February March April

Nebenargument	Beschreibung	Mögliche Werte
		November December
<DDayPos>	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

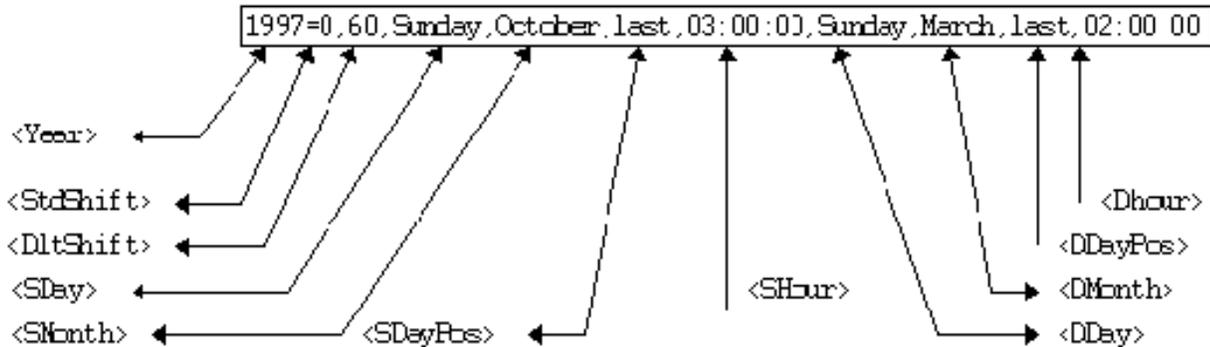
In diesem Beispiel nehmen wir die Informationen zur Sommerzeit der Zeitzone **(GMT+01:00) Paris, Madrid, Amsterdam**.

```
1997=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:

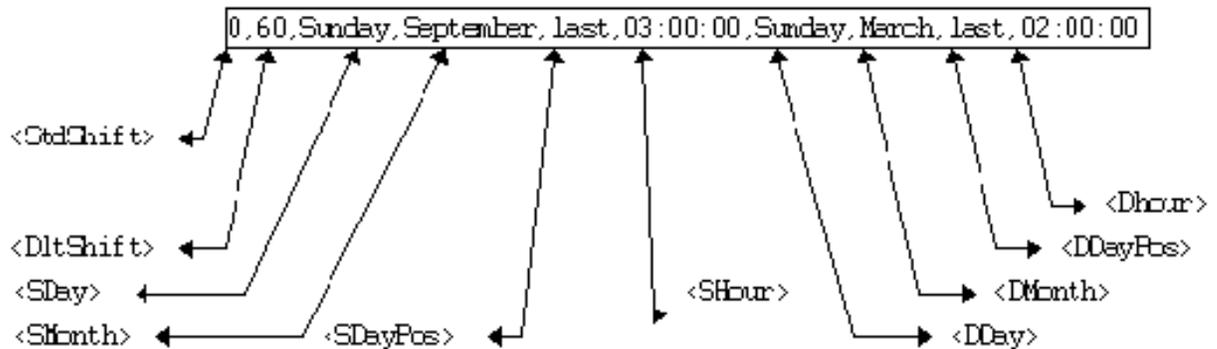
```
1997=0,60,Sunday,October,last,03:00:00,Sunday,March,last,02:00:00
0,60,Sunday,September,last,03:00:00,Sunday,March,last,02:00:00
```

Erster Parameter



- `<Year> = 1997` bedeutet, dass die folgenden Parameter vom Typ `<DaylightInfo>` ab dem Jahr 1997 gültig sind.
- `<StdShift> = 0` bedeutet, dass die Zeitverschiebung zwischen der Winterzeit und der Zonenzeit gleich Null ist. Die Winterzeit entspricht damit der 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:

Ab dem Jahr 1997 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 1997 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.

Verwaltung von Zeitzonen in AssetCenter Server

AssetCenter Server lässt Sie Tests bezüglich der Zeitzonen konfigurieren. Wählen Sie das Menü **Optionen/ 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*

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.

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.

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 in erster Linie bei der Herstellung der Verbindung zwischen AssetCenter Server und der Datenbank durchgeführt.
- Wenn Sie auf der Registerkarte **Module** des Konfigurationsbildschirms von AssetCenter Server einen Kalender definieren, können diese Tests auch häufiger durchgeführt werden.

Auswirkungen der Zeitzonen auf verschiedene Vorgänge

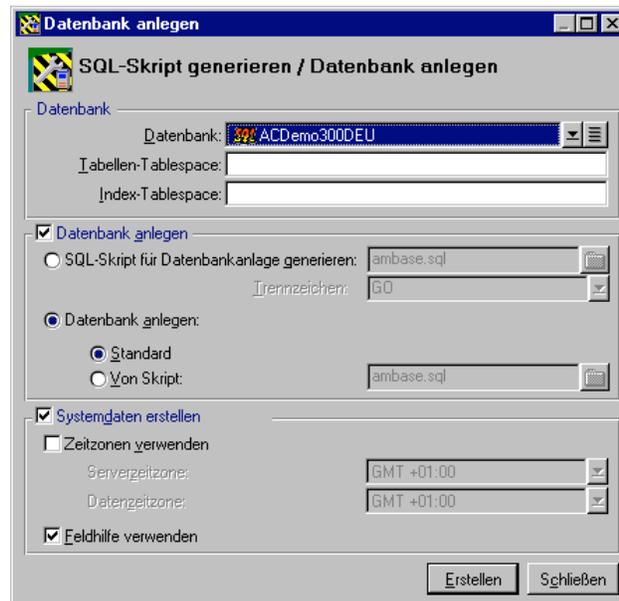
Die Verwaltung der Zeitzonen wirkt sich auf eine Reihe von Vorgängen in AssetCenter aus:

- Datenbanken erstellen
- Verbindung zu einer Datenbank und Zeitzonen
- Import und Export

Datenbanken erstellen

Bei der Erstellung einer Datenbank gibt Ihnen AssetCenter Database Administrator die Möglichkeit, die Optionen zur Verwaltung der Zeitzonen festzulegen.

Wählen Sie in AssetCenter Database Administrator das Menü **Aktion/ Datenbank anlegen**. Der Rahmen **Systemdaten erstellen** enthält die Optionen zur Definition der Zeitzonen.



Optionen bei der Erstellung einer Datenbank

Über das Kontrollkästchen **Zeitzonen 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.

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, machen die Angaben, die für die Felder **Datum+Uhrzeit** in der Datenbank gespeichert wurden, keinen Sinn mehr.

Verbindung zu einer Datenbank und Zeitzonen

Bei der Herstellung einer Verbindung sucht AssetCenter in der Datei **aam.ini** nach dem Eintrag **LocalTimeZone**, der die Zeitzone der Client-Station definiert. Die Datei **aam.ini** befindet sich im Root-Verzeichnis von Windows.

Sollten dort keine Informationen vorliegen, verwendet AssetCenter die für das System festgelegte Zeitzone.

Anschließend versucht AssetCenter die Zeitzone in der Zeitzonentabelle der Datenbank zu finden, die entweder dem Eintrag **LocalTimeZone** in der Datei **aam.ini** oder der unter Windows definierten Zeitzone entspricht.

Die nachstehenden Tabelle zeigt die verschiedenen Möglichkeiten:

Vorhandensein des Eintrags <i>LocalTimeZone</i> in der Datei <i>aam.ini</i>	Vorhandensein einer entsprechenden Zeitzone in der Zeitzonentabelle	In der Datei <i>aam.ini</i> für den Eintrag <i>LocalTimeZone</i> 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 der Client-Station und ermittelt die Zeitverschiebung zwischen den beiden Rechnern. Die Zeitverschiebung wird wie folgt berechnet:

Zeitverschiebung = Modulo((Unterschied zwischen den Uhrzeiten der beiden Rechner, ausgedrückt in Minuten)/30)

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).

```
Zeitverschiebung = Modulo (43/30)= 13 Minuten
```

Wenn die Zeitverschiebung den nicht veränderbaren Wert von fünf Minuten überschreitet, schlägt AssetCenter die Anpassung der Uhrzeit auf der Client-Station vor.

Sollte der Benutzer diese Anpassung ablehnen, kommt die Verbindung nicht zustande.

AssetCenter führt diese Kontrolle regelmäßig und vor allem bei jeder Änderung der Uhrzeit auf der Client-Station durch. Standardmäßig erfolgt die Kontrolle alle 60 Minuten. Dieser Wert kann jedoch über die Option **g_lTimeZoneCheckInMns** in der Datei **Aam.ini** (im Root-Verzeichnis von Windows), und dort im Abschnitt [option] geändert werden.

```
[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.

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 **Datum+Uhrzeit** auf die Zeitzone des Rechners beziehen, der den Import bzw. den Export durchführt.

Kapitel 20 - Arbeiten mit AssetCenter als DDE-Befehlsserver

In diesem Abschnitt werden die von AssetCenter erkannten DDE-Aufrufe beschrieben, und zwar bei der Verwendung von AssetCenter als DDE-Befehlsserver.

Auf den folgenden Seiten finden Sie zunächst eine theoretische Beschreibung, die dann anhand eines praktischen Beispiels der Funktionsweise von DDE-Aufrufen veranschaulicht wird:

Allgemeine Informationen über DDE-Dienste

- Definition eines DDE-Servers
- Dynamic Data Exchange aufrufen

DDE-Befehle

- Einführung in DDE-Befehle
- Globale Befehle
- Einer Tabelle zugeordnete Befehle
- Einer Tabelle und einem Feld bzw. einer Verknüpfung zugeordnete Befehle

Beispiele für DDE-Aufrufe

- Beispiele für DDE-Aufrufe
- Erstes Szenario – AssetCenter interne DDE-Aufrufe
- Zweites Szenario – DDE-Aufrufe über Excel 97

- Drittes Szenario – DDE-Aufrufe über Visual BASIC 5.0

Definition eines DDE-Servers

DDE steht für **Dynamic Data Exchange** und bezeichnet den dynamischen Datenaustausch zwischen verschiedenen Windows-Anwendungen. In unserem Fall könnten Sie beispielsweise AssetCenter mit Hilfe von DDE-Befehlen über andere Anwendungen steuern.

Dynamic Data Exchange aufrufen

DDE basiert auf Diensten, die von den einzelnen Anwendungen bereitgestellt werden. Um einen DDE-Befehl ausführen zu können, müssen Sie ein Thema definieren, das den Kontext angibt, in dem der Befehl ausgeführt werden soll. Jedes Mal, wenn ein Kontext neu definiert wird, müssen Sie den vorherigen Kontext schließen, um die Integrität der DDE-Aufrufe aufrechtzuerhalten.

Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

- DDE-Dienste (Service)
- DDE-Thema
- DDE-Befehl

DDE-Dienste (Service)

In den meisten Fällen handelt es sich bei dem Dienst (Service) um den Namen des geladenen ablauffähigen Programms. Wenn Sie AssetCenter wie in unserem Beispiel als DDE-Server verwenden, lautet der Name des Dienstes **aam**.

Das Programm übernimmt die Rolle des DDE-Befehlsservers, der unbedingt gestartet werden muss, um die einwandfreie Funktionsweise des DDEs zu gewährleisten. In unserem Beispiel müssen Sie dementsprechend AssetCenter laden.

DDE-Thema

Das Thema gibt den Kontext an, in dem die Aktion ausgeführt wird. Für AssetCenter lautet dieses Thema notwendigerweise **AssetCenter**.

DDE-Befehl

Hierbei handelt es sich um Befehle, die von AssetCenter ausgeführt werden sollen. DDE-Befehle lassen sich in verschiedene Gruppen aufgliedern:

- Globale Befehle, für deren Ausführung kein Tabellenname oder Feld als Parameter erforderlich ist.
- Einer Tabelle zugeordnete Befehle, für deren Ausführung der SQL-Name der Tabelle als Parameter erforderlich ist.
- Einer Tabelle und einem Feld bzw. einer Verknüpfung zugeordnete Befehle, für deren Ausführung der SQL-Name der Tabelle und der SQL-Name eines Felds bzw. Verknüpfungen als Parameter erforderlich ist.

Die Befehle in diesen beiden Gruppen entsprechen zwei Typen:

- **Execute** zum Ausführen einer Aufgabe in AssetCenter.
- **Request** zum Abfragen von Informationen aus AssetCenter.

SQL-Namen einer Tabelle, eines Feld und einer Verknüpfung voneinander unterscheiden

Wenn Sie mit der rechten Maustaste auf ein beliebiges Feld in einer Tabelle klicken, zeigt AssetCenter ein Kontextmenü an. Wenn Sie die Option **Objekt konfigurieren** wählen, erscheint in dem daraufhin von AssetCenter angezeigten Fenster der SQL-Name der Tabelle sowie der SQL-Name des Objekts (Verknüpfung oder Feld), auf das Sie geklickt haben.

Einführung in DDE-Befehle

Vorgehensweise

Beachten Sie die nachstehenden drei Schritte, um eine fehlerfreie Ausführung eines DDE-Befehls zu gewährleisten:

- Sie müssen zuerst den Kontext definieren, in dem der Befehl ausgeführt werden soll. Dazu müssen Sie den Dienst und das Thema angeben. Bei der Verwendung von AssetCenter als DDE-Befehlsserver, lautet das Thema grundsätzlich **AssetCenter**.

Nach der Definition gilt der Kontext standardmäßig für alle anschließend ausgeführten DDE-Befehle, solange Sie keinen neuen Kontext definieren.

- Anschließend müssen Sie den Befehl ausführen. Dabei sind zwei Typen von Befehlen zu unterscheiden.
 - ❖ **Execute:** <Befehl>(<Parameter>)
 - ❖ **Request:** <Befehl>(<Parameter>)
- Schließen Sie den zuvor definierten Kontext.

Besonderheiten

Jede Windows-Anwendung verwendet spezifische Methoden zum Weiterleiten von DDE-Aufrufen an andere Anwendungen und zum Empfangen von DDE-Befehlen von anderen Anwendungen. Auf den nächsten Seiten finden Sie folgende Informationen:

- Eine ausführliche Liste der DDE-Befehle, die von AssetCenter empfangen werden. Dazu gehört eine detaillierte Beschreibung der Syntax jedes einzelnen Befehls.
- Beispiele zur Steuerung von AssetCenter mit Hilfe von DDE-Befehlen. In jedem Beispiel wird dabei auf eine andere Programmiersprache zurückgegriffen.

Globale Befehle

Unter globalen Befehlen sind die Befehle zu verstehen, die weder von einer Tabelle noch von einem Feld abhängig sind. Zur Ausführung globaler Befehle ist der SQL-Name einer Tabelle oder eines Felds als Parameter nicht erforderlich.

Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

- Connect(<Cnx>, <User>, <Password>)
- Disconnect()
- ExecuteAction(<ActionName>)
- ListAllTables([Mask])
- ListOpenTables([Mask])
- OpenView(<ViewName>)

Connect(<Cnx>, <User>, <Password>)

Aktionstyp

Execute

Beschreibung

Herstellen der Verbindung zur Datenbank unter Verwendung der nachstehenden Parameter:

<Cnx>

Dieser Parameter enthält eines der nachstehenden Elemente:

- Den Namen der zuvor unter AssetCenter hergestellten Verbindung. Sie finden diesen Namen in der Datei **AmDb.ini**.
- Die vollständige Definition einer Verbindung zu einer Datenbank gemäß der nachfolgend beschriebenen Syntax:

```
[<Datenbank-Engine>;<Standort der Datenbank>;<Login für die  
Datenbank-Engine>;<Kennwort der Datenbank-Engine>]
```

<User>

Dieser Parameter enthält den Namen des AssetCenter-Benutzers, den Sie zur Herstellung einer Verbindung zur Datenbank verwenden möchten.

<Password>

Dieser Parameter enthält das dem Login zugeordnete Kennwort (Wert des Parameters <User>).

Beispiele

Mit dem nachstehenden Befehl können Sie eine Verbindung zu einer Oracle-Datenbank herstellen, für die zuvor eine Verbindung unter AssetCenter definiert wurde. Der Name dieser Verbindung lautet **TDemo**. Die Verbindung erfolgt unter dem Login **Admin** mit dem Kennwort **password**.

```
Execute:Connect(TDemo, Admin, password)
```

Dieser Befehl stellt die gleiche Verbindung her, auch ohne dass sie zuvor unter AssetCenter definiert worden wäre. Die Datenbank **TDemo** befindet sich auf dem Server **Joshua**. Das Kennwort für die Verbindung zur Oracle-Engine lautet **Root**.

```
Execute:Connect([Oracle;Joshua;TDemo;Root], Admin, password)
```

Disconnect()

Aktionstyp

Execute

Beschreibung

Trennen der Verbindung zwischen AssetCenter und der aktuellen Datenbank.

Beispiel

Der nachstehende Befehl trennt die Verbindung zur AssetCenter-Datenbank:

```
Execute:Disconnect()
```

ExecuteAction(<ActionName>)

Aktionstyp

Execute

Beschreibung

Auslösen der Aktion mit dem Namen <ActionName>.

<ActionName>

Dieser Parameter enthält den Namen der Aktion, wie sie unter AssetCenter im Feld **Name** (SQL-Name: Name) des Aktionsdetails definiert wurde.

Beispiel

Der folgende Befehl führt zum Auslösen der Aktion mit dem Namen **Erinnerung: Unterlage nicht zugeordnet**:

```
Execute:ExecuteAction(Erinnerung: Unterlage nicht zugeordnet)
```

ListAllTables([Mask])

Aktionstyp

Request

Beschreibung

Anzeigen einer Liste mit sämtlichen Tabellen der Datenbank. Die Liste besteht aus den SQL-Namen der Tabellen und kann mit Hilfe des Parameters **<Mask>** gefiltert werden.

<Mask>

Über diesen Parameter können Sie die Daten unter Verwendung der folgenden Zeichen filtern:

- Das Fragezeichen (?) kommt als Stellvertreterzeichen für jedes beliebige Zeichen zur Anwendung.
- Der Stern (*) dient zur Darstellung eines beliebigen Zeichens oder einer Zeichengruppe.

Beispiele

Mit dem nachstehenden Befehl zeigen Sie die Liste der SQL-Namen sämtlicher, in der aktuellen Datenbank vorhandenen Tabellen an:

```
Request:ListAllTables()
```

Mit dem folgenden Befehl zeigen Sie die Liste der SQL-Namen sämtlicher Tabellen an, deren SQL-Name mit **amA** beginnt:

```
Request:ListAllTables(amA*)
```

Mit dem folgenden Befehl zeigen Sie die Liste der SQL-Namen sämtlicher Tabellen an, deren SQL-Name den Buchstaben **v** enthält:

```
Request:ListAllTables(*v*)
```

Mit dem folgenden Befehl zeigen Sie die Liste der SQL-Namen sämtlicher Tabellen an, deren SQL-Name mit einem **am** beginnt, und für die der vierte Buchstabe ein **t** ist:

```
Request:ListAllTables(am?t*)
```

ListOpenTables([Mask])

Aktionstyp

Request

Beschreibung

Anzeigen einer Liste mit den SQL-Namen von allen, in der Datenbank geöffneten Tabellen. Diese Liste kann mit Hilfe des Parameters **<Mask>** gefiltert werden.

<Mask>

Über diesen Parameter können Sie die Daten unter Verwendung der folgenden Zeichen filtern:

- Das Fragezeichen (?) kommt als Stellvertreterzeichen für jedes beliebige Zeichen zur Anwendung.
- Der Stern (*) dient zur Darstellung eines beliebigen Zeichens oder einer Zeichengruppe.

Beispiele

Mit dem folgenden Befehl erhalten Sie eine Liste mit den SQL-Namen aller in der aktuellen Datenbank geöffneten Tabellen:

```
Request:ListOpenTables()
```

Mit dem nachstehenden Befehl erhalten Sie eine Liste mit den SQL-Namen aller in der aktuellen Datenbank geöffneten Tabellen, die mit **amA** beginnen:

```
Request:ListOpenTables(amA*)
```

Wir gehen davon aus, dass die drei Tabellen, **amAsset**, **amAction** und **amProduct**, unter AssetCenter geöffnet wurden. Der zuletzt aufgeführte Befehl gibt die SQL-Namen der beiden Tabellen zurück, die mit **amA** beginnen, also: **amAsset** und **amAction**.

OpenView(<ViewName>)

Aktionstyp

Execute

Beschreibung

Öffnen einer zuvor unter AssetCenter definierten Ansicht.

<ViewName>

Dieser Parameter enthält den Namen der Ansicht, wie sie unter AssetCenter definiert wurde.

Befehl

Der folgende Befehl öffnet eine Ansicht mit dem Namen **Geliehene Vermögensgegenstände**:

```
Execute:OpenView(Geliehene Vermögensgegenstände)
```

Einer Tabelle zugeordnete Befehle

Die nächsten Befehle sind von einer Tabelle abhängig. Dementsprechend ist zu ihrer Ausführung der SQL-Name einer Tabelle erforderlich.

Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

- `OpenTable(<Table>)`
- `CloseTable(<Table>)`
- `<Table>.GetRecordCount()`
- `<Table>.SetViewMode(<Mode>)`
- `<Table>.SetRecordMode(<Mode>)`
- `<Table>.ListAllFields([Mask])`
- `<Table>.ListAllLinks([Mask])`
- `<Table>.SetFilter(<Condition>)`
- `<Table>.SetSelection(<Condition>)`
- `<Table>.GetSelectionId()`

OpenTable(<Table>)

Aktionstyp

Execute

Beschreibung

Öffnen einer Tabelle mit dem SQL-Namen **<Table>**.

<Table>

Dieser Parameter enthält den SQL-Namen der Tabelle, die über den Befehl **OpenTable** geöffnet werden soll.

Beispiel

Der nachstehende Befehl öffnet die Tabelle der Vermögensgegenstände (SQL-Name: amAsset):

```
Execute:OpenTable(amAsset)
```

CloseTable(<Table>)

Aktionstyp

Execute

Beschreibung

Schließen einer Tabelle, die zuvor unter AssetCenter geöffnet wurde.

<Table>

Dieser Parameter enthält den SQL-Namen der Tabelle, die über den Befehl **CloseTable** geschlossen werden soll.

Beispiel

Der nachstehende Befehl schließt die Tabelle der Vermögensgegenstände (SQL-Name: amAsset):

```
Execute:CloseTable(amAsset)
```

<Table>.GetRecordCount()

Aktionstyp

Request

Beschreibung

Zurückgeben der Anzahl der Datensätze in der Tabelle mit dem SQL-Namen <Table>. Der Befehl kann nur ausgeführt werden, wenn die Tabelle, für die die Abfrage ausgeführt wird, geöffnet ist.

<Table>

Dieser Parameter enthält den SQL-Namen der Tabelle, für die die Anzahl der Datensätze festgestellt werden soll.

Beispiel

Die folgende Befehl gibt die Anzahl der Datensätze in der Tabelle der Vermögensgegenstände (SQL-Name: amAsset) zurück:

```
Request:amAsset.GetRecordCount ( )
```

<Table>.SetViewMode(<Mode>)

Aktionstyp

Execute

Beschreibung

Definieren der Ansicht einer zuvor geöffneten Tabelle.

<Table>

Dieser Parameter enthält den SQL-Namen der Tabelle, deren Ansicht definiert werden soll.

<Mode>

Dieser Parameter kann einen der nachstehenden Werte annehmen:

- **Arbo**: Die Datensätze der Tabelle <Table> werden als Baumstruktur angezeigt.
- **List**: Die Datensätze der Tabelle <Table> werden als Liste angezeigt.
- **ListOnly**: Die Ansicht beschränkt sich auf die Liste der Datensätze der Tabelle <Table>.
- **DetailOnly**: Die Ansicht beschränkt sich auf das Detail des in der Tabelle <Table> ausgewählten Datensatzes.
- **ListDetail**: Die Ansicht zeigt sowohl die Liste der Datensätze der Tabelle <Table> als auch das Detail der in dieser Liste ausgewählten Datensätze.

Beispiel

Mit dem nachstehenden Befehl wird die Tabelle der Vermögensgegenstände (SQL-Name: amAsset) als Baumstruktur gezeigt:

```
Execute:amAsset.SetViewMode(Arbo)
```

<Table>.SetRecordMode(<Mode>)

Aktionstyp

Execute

Beschreibung

Definieren des Zusammenspiels mit den Datensätzen einer geöffneten Tabelle.

<Table>

Dieser Parameter enthält den SQL-Namen der betroffenen Tabelle.

<Mode>

Dieser Parameter kann einen der nachstehenden Werte annehmen:

- **New:** In der Tabelle <Table> wird ein neuer Datensatz erstellt. Der Befehl entspricht dem Klicken auf die Schaltfläche .
- **Duplicate:** Der in der Tabelle <Table> gewählte Datensatz wird dupliziert. Der Befehl entspricht dem Klicken auf die Schaltfläche .
- **Delete:** Der in der Tabelle <Table> gewählte Datensatz wird gelöscht. Der Befehl entspricht dem Klicken auf die Schaltfläche .
- **Modify:** Bestätigen der Änderungen, die an dem in der Tabelle <Table> ausgewählten Datensatz vorgenommen wurden. Der Befehl entspricht dem Klicken auf die Schaltfläche .
- **Create:** Bestätigen der Erstellung eines neuen Datensatzes in der Tabelle <Table>. Der Befehl entspricht dem Klicken auf die Schaltfläche .
- **CreateContinue:** Kombinieren von Erstellen und Duplizieren. Der Befehl entspricht dem Klicken auf die Schaltfläche .
- **Cancel:** Abbrechen der Erstellung eines neuen Datensatzes oder der am ausgewählten Datensatz vorgenommenen Änderungen. Der Befehl entspricht dem Klicken auf die Schaltfläche .
- **Close:** Schließen der zuvor geöffneten Tabelle <Table>. Der Befehl entspricht dem Klicken auf die Schaltfläche .

Beispiel

Mit den nachstehenden Befehlen wird die Tabelle der Vermögensgegenstände (SQL-Name: amAsset) geöffnet, die Erstellung eines neuen Datensatzes begonnen und dann abgebrochen:

```
Execute: OpenTable ( amAsset )  
Execute: amAsset . SetRecordMode ( New )  
Execute: amAsset . SetRecordMode ( Cancel )
```

<Table>.ListAllFields([Mask])

Aktionstyp

Request

Beschreibung

Zurückgeben der Liste der SQL-Namen aller Felder in der zuvor geöffneten Tabelle **<Table>**.

<Table>

Dieser Parameter enthält den SQL-Namen der Tabelle, in der die Abfrage durchgeführt wird.

<Mask>

Über diesen Parameter können Sie die Daten unter Verwendung der folgenden Zeichen filtern:

- Das Fragezeichen (?) kommt als Stellvertreterzeichen für jedes beliebige Zeichen zur Anwendung.
- Der Stern (*) dient zur Darstellung eines beliebigen Zeichens oder einer Zeichengruppe.

Beispiel

Der nachstehende Befehl gibt die SQL-Namen sämtlicher Felder der Tabelle der Vermögensgegenstände wieder:

```
Request:amAsset.ListAllFields
```

Der folgende Befehl gibt die SQL-Namen sämtlicher Felder der Vermögenstabelle (SQL-Name: amAsset) wieder, deren SQL-Name mit **se** beginnt:

```
Request:amAsset.ListAllFields(se*)
```

<Table>.ListAllLinks([Mask])

Aktionstyp

Request

Beschreibung

Zurückgeben der Liste mit den SQL-Namen für alle Verknüpfungen in der zuvor geöffneten Tabelle **<Table>**.

<Table>

Dieser Parameter enthält den SQL-Namen der Tabelle, in der die Abfrage durchgeführt wird.

<Mask>

Über diesen Parameter können Sie die Daten unter Verwendung der folgenden Zeichen filtern:

- Das Fragezeichen (?) kommt als Stellvertreterzeichen für jedes beliebige Zeichen zur Anwendung.
- Der Stern (*) dient zur Darstellung eines beliebigen Zeichens oder einer Zeichengruppe.

Beispiel

Der nachstehende Befehl gibt die SQL-Namen sämtlicher Felder der Tabelle der Vermögensgegenstände (SQL-Name: amAsset) wieder:

```
Request:amAsset.ListAllLinks
```

Der folgende Befehl gibt die SQL-Namen sämtlicher Felder der Vermögenstabelle (SQL-Name: amAsset) wieder, deren SQL-Name mit **se** beginnt:

```
Request:amAsset.ListAllLinks(se*)
```

<Table>.SetFilter(<Condition>)

Aktionstyp

Execute

Beschreibung

Anwenden eines Filters auf die Tabelle **<Table>** unter Berücksichtigung des Parameters **<Condition>**.

<Table>

Dieser Parameter enthält den SQL-Namen der Tabelle, deren Datensätze gefiltert werden sollen.

<Condition>

Dieser Parameter enthält die Bedingung, die bei der Ausführung des Befehls respektiert werden soll. Im Allgemeinen wird der Parameter wie folgt dargestellt:

```
(<Objekt><Relationaler Operator><Wert>)<Logischer Operator>(<Objekt><Relationaler Operator><Wert>)...
```

Beispiel

Mit dem folgenden Befehl wird ein Filter auf die Tabelle der Vermögensgegenstände (SQL-Name: amAsset) angewendet. Bei Anwendung des Filters werden nur die Datensätze angezeigt, die vor dem 28.08.98 um 15:00:00 geändert wurden:

```
Execute:amAsset.SetFilter(dtLastModif<[98/08/28 15:00:00])
```

<Table>.SetSelection(<Condition>)

Aktionstyp

Execute

Beschreibung

Auswählen eines oder mehrerer Datensätze in der zuvor geöffneten Tabelle <Table> unter Berücksichtigung der Bedingung <Condition>.

<Table>

Dieser Parameter enthält den SQL-Namen der Tabelle, für die ein oder mehrere Datensätze ausgewählt werden sollen.

<Condition>

Dieser Parameter enthält die Bedingung, die bei der Ausführung des Befehls respektiert werden soll. Im Allgemeinen wird der Parameter wie folgt dargestellt:

```
(<Objekt><Relationaler Operator><Wert>)<Logischer Operator>(<Objekt><Relationaler Operator><Wert>)...
```

Beispiel

Mit dem nachstehenden Befehl werden alle Vermögensgegenstände gewählt, deren interner Code größer oder gleich 7 ist:

```
Execute:amAsset.SetSelection(AssetTag>=7)
```

<Table>.GetSelectionId()

Aktionstyp

Request

Beschreibung

Zurückgeben der Liste der Kennungen der in der Tabelle **<Table>** ausgewählten Datensätze.

<Table>

Dieser Parameter enthält den SQL-Namen der Tabelle, in der die Abfrage durchgeführt werden soll.

Beispiel

Im nachstehenden Beispiel werden die Datensätze der Tabelle der Vermögensgegenstände gewählt (SQL-Name: amAsset), deren interner Code größer oder gleich 7 ist. Anschließend wird die Liste der Kennungen der ausgewählten Datensätze zurückgegeben:

```
Execute:amAsset.SetSelection(AssetTag>=7)  
Request:amAsset.GetSelectionId()
```

Einer Tabelle und einem Feld bzw. einer Verknüpfung zugeordnete Befehle

Die nächsten Befehle sind von dem Feld einer Tabelle abhängig. Zum Ausführen dieser Befehle sind der SQL-Name einer Tabelle und der SQL-

Name eines Felds bzw. einer Verknüpfung dieser Tabelle als Parameter erforderlich.

Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

- `<Table>:<Item>.GetValue()`
- `<Tabelle>:<Item>.SetFocus()`
- `<Table>:<Item>.SetValue(<Value>)`
- `<Table>:<Link>.SetValueWhere(<Condition>)`

<Table>:<Objet>.AddLink()

Aktionstyp

Execute

Beschreibung

Simuliert einen Mausklick auf die Schaltfläche  in einem Bereich einer Liste. Dieser Befehl ermöglicht das Hinzufügen einer Verknüpfung zu einem Datensatz einer Tabelle.

<Table>

Dieser Parameter enthält den SQL-Namen der Tabelle, die von der Operation betroffen ist.

<Objet>

Dieser Parameter enthält den SQL-Namen des von der Operation betroffenen Felds.

Beispiel

Der folgende Befehl fügt einer Aufzählung einen Wert hinzu.

```
Execute:amItemizedList:ItemListVals.AddLink()
```

<Table>:<Item>.GetValue()

Aktionstyp

Request

Beschreibung

Zurückgeben des Werts eines <Objekts> (Feld oder Verknüpfung) der Tabelle <Tabelle> für einen ausgewählten Datensatz.

<Tabelle>

Dieser Parameter enthält den SQL-Namen der Tabelle, in der die Abfrage durchgeführt werden soll.

<Objekt>

Dieser Parameter enthält den SQL-Namen des Felds oder der Verknüpfung der Tabelle <Tabelle>, aus der Sie einen Wert abrufen möchten.

Beispiel

Der nachstehende Befehl gibt den Wert des Felds **Feld 1** (SQL-Name: Field1) aus der Tabelle der Vermögensgegenstände (SQL-Name: amAsset) zurück:

```
Request:amAsset:Field1.GetValue()
```

Der nachstehende Befehl gibt den Wert der Verknüpfung **Kategorie** (SQL-Name: Category) der Tabelle der Vermögensgegenstände (SQL-Name: amAsset) zurück:

```
Request:amAsset:Category.GetValue()
```

Der nachstehende Befehl gibt den Wert der Verknüpfung **Kommentar** (SQL-Name: Comment) der Tabelle der Vermögensgegenstände (SQL-Name: amAsset) zurück:

```
Request:amAsset:Comment.GetValue()
```

<Table>:<Objet>.Highlight()

Aktionstyp

Execute

Beschreibung

Setzt den Fokus auf ein Feld und lässt es blinken.

<Table>

Dieser Parameter enthält den SQL-Namen der Tabelle, die von der Operation betroffen ist.

<Objet>

Dieser Parameter enthält den SQL-Namen des von der Operation betroffenen Felds. Dieser Befehl funktioniert nicht in Verbindung mit einer Verknüpfung.

Beispiel

Der folgende Befehl lässt das Feld **Strichcode** (SQL-Name: Barcode) in der Vermögenstabelle (SQL-Name: amAsset) blinken:

```
Execute:amAsset:Barcode.Highlight()
```

<Table>:<Objet>.RemoveLink()

Aktionstyp

Execute

Beschreibung

Simuliert einen Mausklick auf die Schaltfläche  in einem Listenbereich. Dieser Befehl ermöglicht das Löschen einer Verknüpfung mit dem Datensatz einer Tabelle.

<Table>

Dieser Parameter enthält den SQL-Namen der Tabelle, die von der Operation betroffen ist.

<Objet>

Dieser Parameter enthält den SQL-Namen des von der Operation betroffenen Felds.

Beispiel

Der folgende Befehl löscht den markierten Wert in einer Aufzählung.

```
Execute:amItemizedList:ItemListVals.RemoveLink()
```

<Tabelle>:<Item>.SetFocus()

Aktionstyp

Execute

Beschreibung

Verschieben Sie den Fokus auf das Feld oder die Verknüpfung **<Objekt>** der Tabelle **<Tabelle>** für den ausgewählten Datensatz.

<Tabelle>

Dieser Parameter enthält den SQL-Namen der Tabelle mit dem Feld bzw. der Verknüpfung, auf die Sie den Fokus verschieben möchten.

<Objekt>

Dieser Parameter enthält den SQL-Namen des Felds bzw. der Verknüpfung der Tabelle **<Tabelle>**, auf die Sie den Fokus verschieben wollen.

Beispiel

Der nachstehende Befehl verschiebt den Fokus auf die Verknüpfung **Kategorie** (SQL-Name: Category) der Tabelle der Vermögensgegenstände (SQL-Name: amAsset):

```
Request:amAsset:Category.SetFocus( )
```

<Table>:<Item>.SetValue(<Value>)

Aktionstyp

Execute

Beschreibung

Füllt das Feld <Field> der Tabelle <Table> für den ausgewählten Datensatz mit dem Wert <Value>.

<Table>

Dieser Parameter enthält den SQL-Namen der Tabelle mit dem Feld, das Sie ausfüllen möchten.

<Field>

Dieser Parameter enthält den SQL-Namen des Felds der Tabelle <Table>, die Sie ausfüllen möchten.

<Value>

Dieser Parameter enthält den Wert, den Sie dem Feld <Field> der Tabelle <Table> für einen ausgewählten Datensatz zuordnen möchten. Wenn das Feld dem Typ **Datum** oder **Datum+Uhrzeit** entspricht, muss dieser Parameter im international gültigen Datumsformat (yy/mm/dd hh:mm:ss) angegeben werden.

Beispiele

Der nachstehende Befehl ruft den Wert **Test** ins Feld **Feld 1** (SQL-Name: Field1) der Tabelle der Vermögensgegenstände (SQL-Name: amAsset) für den ausgewählten Datensatz ab:

```
Execute:amAsset:Field1.SetValue(Test)
```

Der nächste Befehl ruft den Wert **28.08.98** in das Feld **Install.-Dt.** (SQL-Name: dInstall) der Tabelle der Vermögensgegenstände (SQL-Name: amAsset) für den ausgewählten Datensatz ab:

```
Execute:amAsset:dInstall.SetValue(98/08/28)
```

<Table>:<Link>.SetValueWhere(<Condition>)

Aktionstyp

Execute

Beschreibung

Eingeben der Verknüpfung <Link> der Tabelle <Table> für den ausgewählten Datensatz unter Berücksichtigung der Bedingung <Condition>.

<Table>

Dieser Parameter enthält den SQL-Namen mit dem Feld, das Sie ausfüllen möchten.

<Link>

Dieser Parameter enthält den SQL-Namen der Verknüpfung der Tabelle <Table>, die Sie ausfüllen möchten.

<Condition>

Mit diesem Datensatz wird der Zieldatensatz der Verknüpfung identifiziert. Im Allgemeinen weist der Parameter die nachstehende Struktur auf:

```
<Objekt><Relationeller Operator><Wert><Logischer  
Operator>( <Objekt><Relationeller Operator><Wert> )...
```

Beispiele

Im folgenden Beispiel wird der Wert **Test** in die Verknüpfung **Kategorie** (SQL-Name: Category) der Tabelle der Vermögensgegenstände (SQL-Name: amAsset) für den ausgewählten Datensatz gerufen. Die Kategorie **Test** muss vorhanden sein, damit der DDE-Befehl ordnungsgemäß ausgeführt werden kann.

```
Execute:amCategory.SetSelection(Name='Test')
```

<Table>:<Objet>.Show()

Aktionstyp

Execute

Beschreibung

Verschiebt den Fokus auf ein Feld oder eine Verknüpfung, der bzw. die auf dem Bildschirm nicht sichtbar ist. Die Tabelle mit dem Feld oder der Verknüpfung muss geöffnet sein.

<Table>

Dieser Parameter enthält den SQL-Namen der von der Operation betroffenen Tabelle.

<Objet>

Dieser Parameter enthält den SQL-Namen des betroffenen Objekts.

Beispiel

Mit dem folgenden Befehl wird der Fokus auf das Feld verschoben. **Code** (SQL-Name: AcctCode) der Tabelle der Vermögensgegenstände (SQL-Name: amAsset):

```
Execute:amAsset:AcctCode.Show( )
```

Beispiele für DDE-Aufrufe

Zur Veranschaulichung dieser AssetCenter-Funktionalität und der Verwendung von DDE-Diensten möchten wir Ihnen auf den nächsten Seiten einige Szenarien vorstellen:

- Im ersten Szenario beschreiben wir den theoretischen Ablauf eines DDE-Aufrufs.
- Im zweiten Szenario stellen wir Ihnen die Verwendung von DDE-Aufrufen unter Excel anhand eines praktischen Beispiels vor. Das Beispiel wurde mit Hilfe von Visual BASIC for Applications programmiert.
- Das dritte Szenario zeigt eine in Visual BASIC 5.0 geschriebene Anwendung, die Ihnen das Ausführen von DDE-Aufrufen auf eine besonders ergonomische Weise vorführt.

Sie müssen AssetCenter starten, um die Beispiele nachvollziehen zu können.

Erstes Szenario – AssetCenter interne DDE-Aufrufe

In diesem Szenario haben wir uns zum Ziel gesetzt, den geeigneten DDE-Befehl und die entsprechenden Parameter für jede Aktion zu identifizieren, die Sie ausführen möchten. Das Beispiel zeigt die theoretischen Aspekte der Verwendung von DDE-Mechanismen. Der praktische Aspekt ist den nächsten Szenarien vorbehalten.

In unserem Beispiel erstellen wir zunächst ein Merkmal vom Typ **Numerisch**, das wir **Arbeitsspeicher** nennen. Nachstehend finden Sie die Aktionen, die dabei ausgeführt werden:

- Öffnen der Merkmaltabelle
- Eingeben der Bezeichnung des Merkmals
- Definieren des Eingabetyps für das Merkmal
- Eingeben der Einheit

- Erstellen des Merkmals

Einführung

Wir müssen zuerst den Dienst und das Thema für die Ausführung des DDE-Befehls identifizieren.

Wir befinden uns in der Anwendung AssetCenter, und daher gilt:

- Dienst: **aam**
- Thema: **AssetCenter**

Sie brauchen jetzt nur noch den Befehl auszuführen, mit dem die Merkmaltabelle geöffnet wird:

- Befehl: `OpenTable()`
- Parameter: Der SQL-Name der Tabelle, die Sie öffnen möchten, also **amFeature**.

Der Befehl schreibt sich wie folgt:

```
OpenTable(amFeature)
```

AssetCenter öffnet die Tabelle mit den Merkmalen. Starten Sie jetzt einfach die Erstellung eines neuen Datensatzes für diese Tabelle:

- Befehl: `SetRecordMode()`
- Präfix des Befehls: Der SQL-Name der Tabelle, hier **amFeature**
- Parameter: Der Eingabemodus, hier **New**

Der Befehl schreibt sich wie folgt:

```
amFeature.SetRecordMode(New)
```

Eingaben

Sie müssen AssetCenter jetzt nur noch angeben, welche Befehle uns bei der Eingabe der Felder interessieren:

- Feld **Bez.** (SQL-Name: `TextLabel`). Der zu verwendende Befehl und die entsprechenden Parameter lauten wie folgt:
 - ❖ Befehl : `<Table>:<Item>.SetValue(<Value>)`
 - * Parameter **<Table>**: Der SQL-Name der Tabelle, also **amFeature**
 - * Parameter **<Item>**: Der SQL-Name des Felds, also **TextLabel**
 - * Parameter **<Value>**: Der Wert des Felds, also **Arbeitsspeicher**

```
amFeature:TextLabel.SetValue(Arbeitsspeicher)
```

- Feld **Eingabeart** (SQL-Name: seDataType). Der zu verwendende Befehl und die entsprechenden Parameter lauten wie folgt:

- ❖ Befehl: `<Table>:<Item>.SetValue(<Value>)`
 - * Parameter **<Table>**: Der SQL-Name der Tabelle, also **amFeature**
 - * Parameter **<Item>**: Der SQL-Name des Felds, also **seDataType**
 - * Parameter **<Value>**: Der Wert des Felds, also **Numerisch**

```
amFeature:seDataType.SetValue(Numerisch)
```

- Feld **Einheit** (SQL-Name: Unit). Der zu verwendende Befehl und die entsprechenden Parameter lauten wie folgt:

- ❖ Befehl: `<Table>:<Item>.SetValue(<Value>)`
 - * Parameter **<Table>**: Der SQL-Name der Tabelle, also **amFeature**
 - * Parameter **<Item>**: Der SQL-Name des Felds, also **Unit**
 - * Parameter **<Value>**: Der Wert des Felds, also **MB**

```
amFeature:Unit.SetValue(MB)
```

Merkmal erstellen

Zum Erstellen des Merkmals brauchen Sie jetzt nur noch den Datensatz in der Merkmaltabelle zu erstellen:

- Befehl: `<Table>.SetRecordMode(<Mode>)`
 - ❖ Parameter **<Table>**: Der SQL-Name der Tabelle, also **amFeature**
 - ❖ Parameter **<Mode>**: Der Erstellungsmodus eines Datensatzes, also **Create**

```
amFeature.SetRecordMode(Create)
```

Zweites Szenario – DDE-Aufrufe über Excel 97

Zum Nachvollziehen dieses Beispiels müssen Sie zunächst die Datei **TestDDE.xls** öffnen. Diese Datei befindet sich im AssetCenter-Verzeichnis **AmDemo** und enthält alle Makro-Befehle, die für dieses Beispiel erforderlich sind.

Dieses Beispiel ist identisch mit dem vorherigen, mit Ausnahme der DDE-Aufrufe, für die der VBA-Standard (Visual BASIC for Applications) gilt.

AssetCenter muss gestartet werden, bevor Sie die von Excel 97 weitergeleiteten DDE-Befehle empfangen können.

In diesem Kapitel finden Sie Informationen zu den nachstehenden Punkten:

- Beschreibung des Makros
- Quelle des Makros

Beschreibung des Makros

Positionieren Sie den Cursor auf dem Excel-Blatt mit der Überschrift **Eingabe** in der Datei **TestDDE.xls**. Die Datei enthält eine Tabelle mit drei Spalten: **Bez.**, **Eingabeart**, und **Einheit**, sowie eine Schaltfläche (mit der Bezeichnung **Merkmal erstellen**) der das Makro **Create** zugeordnet wurde.

Geben Sie die erforderlichen Werte in den Feldern **Bez.** (in unserem Beispiel die Zelle B6), **Eingabeart** (Zelle C6), **Einheit** (Zelle D6) ein, und klicken Sie auf die Schaltfläche **Merkmal erstellen**, damit Excel die Software AssetCenter zum Ausführen der nachstehenden Aktionen auffordert:

- Öffnen der Merkmaltabelle
- Eingeben der Bezeichnung des Merkmals
- Definieren des Eingabetyps für das Merkmal
- Eingeben der Einheit
- Erstellen des Merkmals

Quelle des Makros

Nachstehend finden Sie die Liste des Makro-Programms, das die DDE-Aufrufe ausführt. Weitere Informationen zu den DDE-Aufrufen unter Excel oder Word und zur Syntax der Sprache VBA finden Sie in den Handbüchern, die im Lieferumfang dieser Software enthalten sind.

```
Sub CreateFeature()  
Set Label = Worksheets("Eingabe").Range("B6")  
Set Type = Worksheets("Eingabe").Range("C6")  
Set Unit = Worksheets("Eingabe").Range("D6")  
Contexte = Application.DDEInitiate(app:="aam", topic:="AssetCenter")  
Application.DDEExecute Contexte, "OpenTable(amFeature)"  
Application.DDEExecute Contexte, "amFeature.SetRecordMode(New)
```

```
Application.DDEExecute Contexte,  
"amFeature:TextLabel.SetValue("+Label+")"  
Application.DDEExecute Contexte,  
"amFeature:seDataType.SetValue("+Type+")"  
Application.DDEExecute Contexte, "amFeature:Unit.SetValue("+Unit+")"  
Application.DDEExecute Contexte, "amFeature.SetRecordMode(Create)"  
Application.DDETerminate Contexte  
End Sub
```

Drittes Szenario – DDE-Aufrufe über Visual BASIC 5.0

In diesem Szenario stellen wir Ihnen ein Dienstprogramm vor, das Ihnen eine benutzerfreundliche Schnittstelle zum Ausprobieren der DDE-Mechanismen bietet. Zusätzlich veranschaulicht das Szenario die Vorgehensweise beim Programmieren von DDE-Mechanismen in Visual BASIC 5.0.

Wenn Sie dieses Beispiel nachvollziehen möchten, müssen Sie zunächst das Programm **DDE TestCenter.exe** ausführen, das sich im AssetCenter-Verzeichnis **AmDemo** befindet. Dieses Programm ermöglicht Ihnen die Ausführung der DDE-Befehle vom Typ **Execute** und **Request**.

AssetCenter muss gestartet werden, bevor Sie die vom BASIC-Programm weitergeleiteten DDE-Befehle empfangen können.

Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

- Vorsichtsmaßnahmen
- Programm ausführen
- Quelle des Programms

Vorsichtsmaßnahmen

Installation

Sie müssen Visual BASIC 5.0 auf Ihrem Rechner installiert haben, bevor Sie dieses Programm verwenden können. Dabei müssen einige ActiveX-Steuerelemente in Ihrem Windows-Registry gespeichert werden. Wenn DDE TestCenter Ihnen einen Fehler vom Typ **Das Steuerelement**

XXXX ist nicht gespeichert zurückgibt, müssen Sie die nachstehende Manipulation ausführen:

- Suchen Sie nach der Position des Steuerelements auf Ihrer Festplatte und kopieren Sie es in das entsprechende Verzeichnis.
- Führen Sie den folgenden Befehl aus:

```
regsvr32 XXXX
```

- Starten Sie die Anwendung DDE TestCenter. Sollte auch der zweite Startversuch fehlschlagen, müssen Sie das Handbuch von Visual BASIC 5.0 zu Rate ziehen.

Empfehlungen

Wenn Sie das Beispiel unter optimalen Bedingungen nachvollziehen möchten, sollten Sie nachstehenden Anweisungen folgen:

- Starten Sie AssetCenter, und minimieren Sie das Fenster der Anwendung auf die Hälfte der Größe Ihres Bildschirms.
- Starten Sie das Programm **DDE TestCenter.exe** und platzieren Sie das Fenster mit der Anwendung neben das Fenster mit AssetCenter.

Das Ergebnis der in **DDE TestCenter.exe** eingegebenen Befehle kann direkt in AssetCenter eingesehen werden.

Syntax

Dieses Beispiel ist identisch mit dem vorherigen, mit Ausnahme der DDE-Aufrufe, für die der VBA-Standard (Visual BASIC for Applications) gilt.

Programm ausführen

DDE-Befehl vom Typ *Execute*

Positionieren Sie den Cursor auf der Registerkarte **Execute**.

Geben Sie den auszuführenden Befehl im Feld **Command** unter Berücksichtigung der nachstehenden Syntax ein:

```
Command=<commande>( <Parameter> )
```

Klicken Sie auf die Schaltfläche , um den Befehl auszuführen. Eventuell auftretende Fehler werden im Feld **Last DDE Error** zurückgegeben.

Beispiel Nr.1:

Mit dem nächsten Befehl vom Typ **Execute** wird die Merkmaltabelle geöffnet:

```
OpenTable(amFeature)
```

Beispiel Nr.2:

Mit den nächsten Befehlen vom Typ **Execute** wird die Bugdettabelle geöffnet, ein neuer Datensatz erstellt und das Feld **Name** (SQL-Name: Name) im Detailbildschirm ausgefüllt. Führen Sie diese Befehle nacheinander aus:

```
OpenTable(amBudget)  
amBudget.SetRecordMode(New)  
amBudget.Name.SetValue("Test")
```

DDE-Befehl vom Typ **Request**

Positionieren Sie den Cursor auf der Registerkarte **Request**.

Geben Sie den auszuführenden Befehl im Feld **Command** unter Berücksichtigung der nachstehenden Syntax ein:

```
Command=<Befehl>(<Parameter>)
```

Klicken Sie auf die Schaltfläche , um den Befehl auszuführen. Das Ergebnis der Abfrage erscheint im Feld **Request Result**. Eventuell auftretende Fehler werden im Feld **Last DDE Error** zurückgegeben.

Beispiel Nr.1:

Mit dem nächsten Befehl vom Typ **Request** wird die Liste der SQL-Namen aller, in der aktuellen Verbindung vorhandenen Tabellen angezeigt:

```
ListAllTables()
```

Beispiel Nr.2:

Mit dem nächsten Befehl vom Typ **Request** wird die Liste der SQL-Namen aller Felder der zuvor geöffneten Merkmaltabelle (SQL-Name: amFeature) angezeigt:

```
amFeature.ListAllFields()
```

Quelle des Programms

Sie finden den kommentierten Quellcode des Programms im AssetCenter-Verzeichnis **AmDemo**, und zwar in Form eines Visual BASIC 5.0-Projekts.

Kapitel 21 - Berechnete Felder

Der Zugriff auf den Bildschirm zur Erstellung der berechneten Felder erfolgt über das Menü Extras/ Administration/ Berechnete Felder.

Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

Einführung

- Definition eines berechneten Felds
- Vorteile der berechneten Felder

Erstellen eines berechneten Felds

- Berechnete Felder erstellen
- Einführung
- Vorgehensweise

Verwendung

- Arbeiten mit berechneten Feldern

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. Virtueller 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.

Einführung

Jeder der drei Typen von berechneten Feldern verfügt über unterschiedliche Eigenschaften, die sich auf seine Verwendung auswirken.

In der nachstehenden Tabelle finden Sie die wichtigsten Unterschiede zwischen den drei Typen:

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.
Berechnete Strings	Ja	Ja	Ja	Client	Benutzerfreundlich	Nicht sehr leistungsfähig (einfache Verkettung von Strings und Werten von Feldern und Verknüpfungen).
BASIC	Ja	Ja	Ja	Client	Vielseitig Flexibel	Die Felder dieses Typs sind nur anzeigbar.

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	Klausel SELECT
Sortierbar	Klauseln SELECT, ORDER BY, GROUP BY
In Filtern verwendbar	Klauseln SELECT, ORDER BY, GROUP BY, WHERE, HAVING

Weitere Informationen zu den AQL-Abfragen finden Sie im Handbuch **"Referenzhandbuch: Verwaltung und Verwendung für fortgeschrittene Benutzer"** Kapitel "Schreiben von AQL-Abfragen".

Berechnungen auf dem Server bzw. der Client-Station

Im Fall eines Felds vom Typ **AQL** werden die Berechnungen vom Datenbankserver durchgeführt, der die Ergebnisse an die Client-Station zurückgibt. Die Berechnungen wirken sich dementsprechend nicht auf die Verarbeitungsgeschwindigkeit der Client-Station 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

Wir empfehlen Ihnen, sooft wie möglich den Typ des berechneten Felds mit dem geringsten Aufwand zu verwenden.

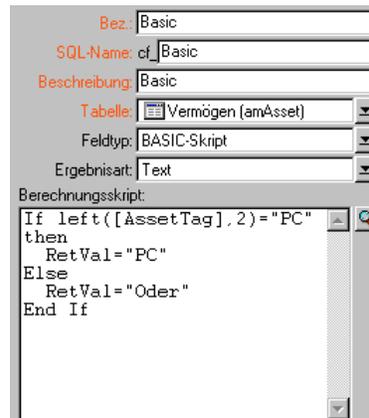
Ein Beispiel:

- 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 **Extras/ Administration/ Berechnete Felder**. AssetCenter zeigt den Bildschirm zur Erstellung der berechneten Felder an.



Detail eines berechneten Felds

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.

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 Felds 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**.

Je nach Typ eines berechneten Felds wird eine andere Sprache verwendet.

Weitere Informationen zu den Sprachen, die zum Schreiben von Berechnungsformeln dienen, finden Sie in den nachstehenden Handbüchern:

- BASIC: "Referenzhandbuch: Verwaltung und Verwendung für fortgeschrittene Benutzer", Kapitel "Arbeiten mit Skripten". Die verwendete Funktion ist **RetVal()**.
- AQL: "Referenzhandbuch: Verwaltung und Verwendung für fortgeschrittene Benutzer" Kapitel "Schreiben von AQL-Abfragen".
- Berechnete Strings: "Referenzhandbuch: Verwaltung und Verwendung für fortgeschrittene Benutzer", Kapitel "Struktur der AssetCenter-Datenbank", Abschnitt "Beschreibung der Tabellen", Unterabschnitt "Beschreibungsfolgen der AssetCenter-Tabellen".

Bestimmen Sie die Standardberechtigungen für das berechnete Feld

Wählen Sie das Menü **Extras/ Administration/ Benutzungsberechtigungen**. AssetCenter zeigt den Bildschirm zur Erstellung der Standardberechtigungen an.

Berechnete Felder sind schreibgeschützt.

- Geben Sie im Feld **Beschreibung** (SQL-Name: Description) eine kurze Beschreibung der Standardberechtigung und im Feld **Kommentar** (SQL-Name: Comment) eventuell einen Kommentar ein.
- Öffnen Sie die Baumstruktur der Tabelle, der das berechnete Feld zugeordnet ist. Der mit dem Symbol  gekennzeichnete Zweig enthält eine komplette Liste der berechneten Felder der betreffenden Tabelle.
- 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 **Extras/ Optionen**, Registerkarte **Anzeigen** definiert wurde.

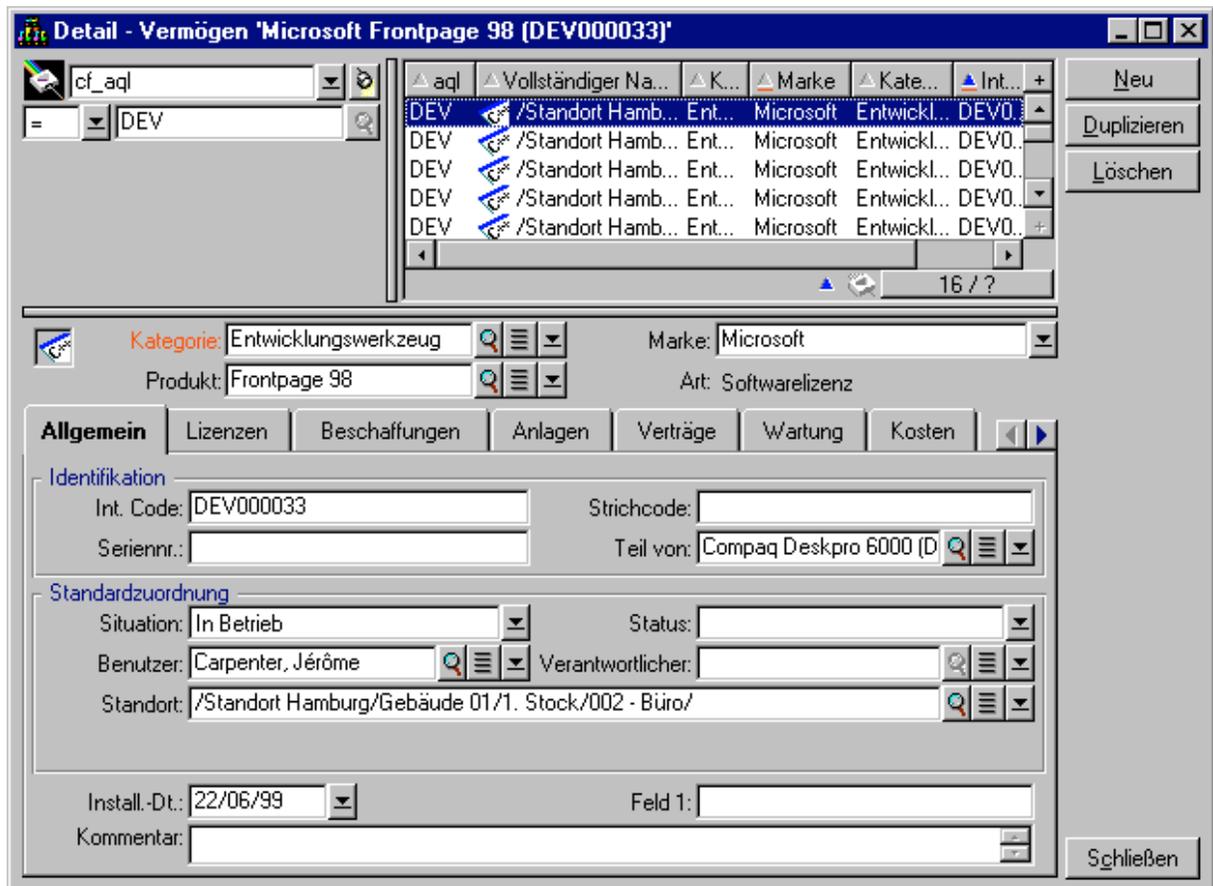
Anschließend können Sie die berechneten Felder wie jedes andere Feld zu der Liste hinzufügen.

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 APIs
- Berichte
- Formulare

Weitere Informationen zu den Modulen bzw. Funktionalitäten finden Sie in den entsprechenden Handbüchern.

Kapitel 22 - Präsentation der Assistenten

Der Zugriff auf die Assistenten erfolgt über das Menü Extras/ Aktionen/ Bearbeiten und die anschließende Auswahl einer Aktion vom Typ Assistent.

Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

Einführung

- Definition eines Assistenten
- An wen richten sich die Assistenten?

Beschreibung verschiedener Typen von Assistenten

- Assistenten zum Austauschen von Daten mit der AssetCenter-Datenbank
- Unabhängige Assistenten

Definition eines Assistenten

Die AssetCenter-Assistenten helfen Ihnen bei der Ausführung einfacher; regelmäßig ausgeführter Aufgaben. Eine intuitive, grafische Benutzeroberfläche führt Sie durch die einzelnen Etappen, die zur Ausführung der Aufgabe erforderlich sind. Im Lieferumfang von AssetCenter sind vordefinierte Assistenten vorhanden, die Ihnen die nachstehenden Aufgaben erleichtern:

- Umzug von Mitarbeitern und Vermögensgegenständen an einen anderen Standort. Der Benutzer wählt einen Mitarbeiter, die Vermögensgegenstände, die zusammen mit dem Mitarbeiter an einen anderen Standort gebracht werden sollen, und den neuen Standort. Sobald der Assistent all diese Informationen erhalten hat, wird der Standort mit den Angaben zu Mitarbeiter und Vermögensgegenstand aktualisiert.
- Vereinfachte Bestandsverwaltung. Der Benutzer wählt die lagerhaltigen Vermögensgegenstände, einen Mitarbeiter und einen Standort. Der Assistent ordnet die Vermögensgegenstände dem ausgewählten Mitarbeiter und Standort zu.
- Benutzerfreundliches Sammeln von Informationen für Aktionen beliebigen Typs.
- Vereinfachte Eingabe von Datensätzen.

AssetCenter lässt Sie neben den standardmäßig gelieferten Assistenten auch völlig neue Assistenten erstellen.

An wen richten sich die Assistenten?

Die Assistenten sind sowohl für den normalen als auch für den fortgeschrittenen Benutzer eine besondere Hilfe:

- Der normale Benutzer kann komplexe Aufgaben ohne Mühe erledigen, ohne dazu über eine gründliche Kenntnis von AssetCenter und seinen Funktionalitäten verfügen zu müssen.
- Der fortgeschrittene Benutzer kann eigene Assistenten erstellen oder vorhandene Assistenten konfigurieren, um sie an interne Geschäftsprozesse anzupassen. Zur Erstellung dedizierter Assistenten ist die Kenntnis einer flexiblen und leistungsstarken Skriptsprache erforderlich.

Beschreibung verschiedener Assistenten

Bei den Assistenten sind zwei Gruppen zu unterscheiden:

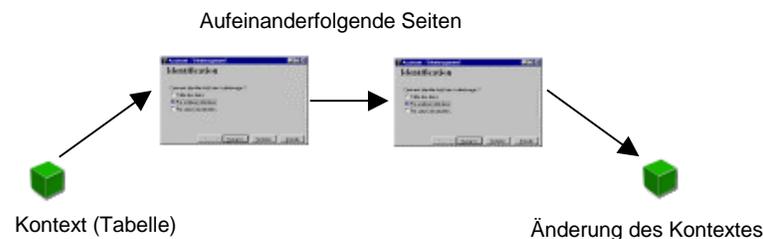
- Assistenten zum Austauschen von Daten mit der AssetCenter-Datenbank
- Unabhängige Assistenten

Assistenten zum Austauschen von Daten mit der AssetCenter-Datenbank

In diese Kategorie fallen zwei Gruppen:

Assistenten zur Unterstützung bei der Dateneingabe

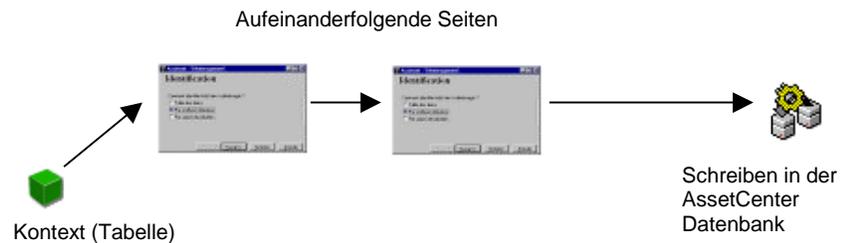
Diese Assistenten erleichtern die Eingabe von Informationen und die Erstellung von Datensätzen anhand dieser Informationen. Dabei übernehmen die Assistenten das Ausfüllen der Bildschirme an Ihrer Stelle. Der Assistent **Neuen Mitarbeiter erstellen** beispielsweise führt den Benutzer durch die Erstellung eines Datensatzes für die Tabelle der Abteilungen und Personen. Die Informationen über den Mitarbeiter werden von dem Assistenten gesammelt, der auch den entsprechenden Datensatz generiert. Der Benutzer gibt keine Daten direkt auf dem Detailbildschirm ein. Diese Aufgabe übernimmt der Assistent.



Assistenten zum Sammeln von Informationen vor dem Ausführen einer Aktion

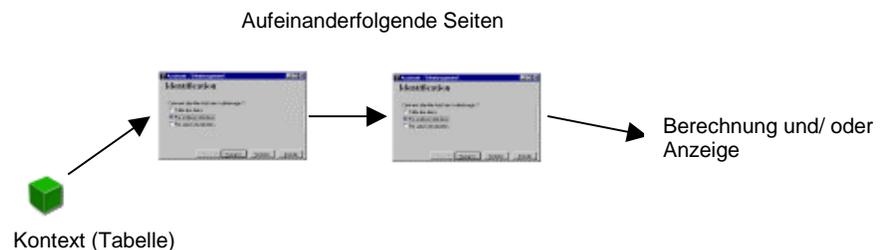
Für bestimmte Aktionen ist der Wert eines Felds der AssetCenter-Datenbank oder der Wert einer Variablen erforderlich. Diese Assistenten helfen Ihnen beim Sammeln der erforderlichen Informationen, und leiten

diese Informationen an die Aktion weiter. So sammelt der im Lieferumfang von AssetCenter enthaltene Assistent zur Verwaltung eines Umzugs beispielsweise Informationen über die Vermögensgegenstände, die an einen neuen Standort gebracht werden sollen, den Benutzer dieser Gegenstände sowie den neuen Standort. Anschließend führt der Assistent die entsprechenden Änderungen in der AssetCenter-Datenbank durch.



Unabhängige Assistenten

Bei den unabhängigen Assistenten handelt es sich in der Regel um Assistenten zur Berechnung und Anzeige. So können Sie beispielsweise den Assistenten **Addieren** erstellen, der die Summe von zwei, vom Benutzer eingegebenen Zahlen anzeigt.



Kapitel 23 - Erstellen von Assistenten

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.

Auf den folgenden Seiten finden Sie Informationen zu den nachstehenden Punkten:

Konventionen und Definitionen

- Verwendete Konventionen
- Definitionen zu Assistenten

Strukturmodelle

- Strukturmodelle
- Modell der Seite eines Assistenten

Struktur von Assistenten

- Allgemeine Informationen zu Assistenten
- Allgemeine Struktur und Syntax eines Assistentenknotens
- Eigenschaften eines Knotens
- Verkettung von Assistenten

- BASIC-Funktionen

Der Knoten Root

- Definition eines Knotens vom Typ Root
- Syntax eines Knotens vom Typ Root
- Eigenschaften des Knotens Root
- Dem Knoten Root untergeordnete Knoten

Der Knoten Page

- Definition des Knotens Page
- Syntax eines Knotens vom Typ Page
- Eigenschaften eines Knotens vom Typ Page
- Dem Knoten Page untergeordnete Knoten

Der Knoten Transition

- Definition eines Knotens vom Typ Transition
- Syntax eines Knotens vom Typ Transition
- Eigenschaften eines Knotens vom Typ Transition
- Besonderheiten eines Knotens vom Typ Transition

Der Knoten Finish

- Definition eines Knotens vom Typ Finish

Der Knoten Start

- Definition eines Knotens vom Typ Start

Der Knoten Steuerelement

- Definition eines Knotens vom Typ Steuerelement
- Allgemeine Syntax eines Knotens vom Typ Steuerelement
- Steuerelementtypen mit ihren zugeordneten Eigenschaften

Erstellung eines Assistenten

- Beispiel für die Erstellung eines Assistenten
- Arbeiten mit dem Grafikeditor
- Häufig gestellte Fragen

Verwendete Konventionen

Die Struktur des Assistenten wird mit folgenden Notationen beschrieben:

[]	Die eckigen Klammern werden verwendet, um den Wert eines Felds der Datenbank (bei kontextbezogenen Assistenten) oder eines der Spezialfelder CurrentSelection und CurrentTable als Referenz auszuweisen.
< >	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.
'	Das Apostroph bezeichnet eine Zeile mit einem Kommentar, 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 wird. 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 Nebennoten 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.

Das Verwenden von Sonderzeichen in Knotennamen ist unzulässig. 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 ein Objekt A ein Objekt B enthält, dann gilt Folgendes:

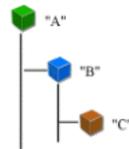
- Das Objekt A ist dem Objekt B übergeordnet.

- 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 Namen 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 im Assistenten auf die Schaltfläche klickt, wird der erste gültige Übergang, also der Übergang, für den alle Bedingungen erfüllt sind, ausgeführt. Wenn kein gültiger Übergang vorliegt, wird die Schaltfläche deaktiviert.

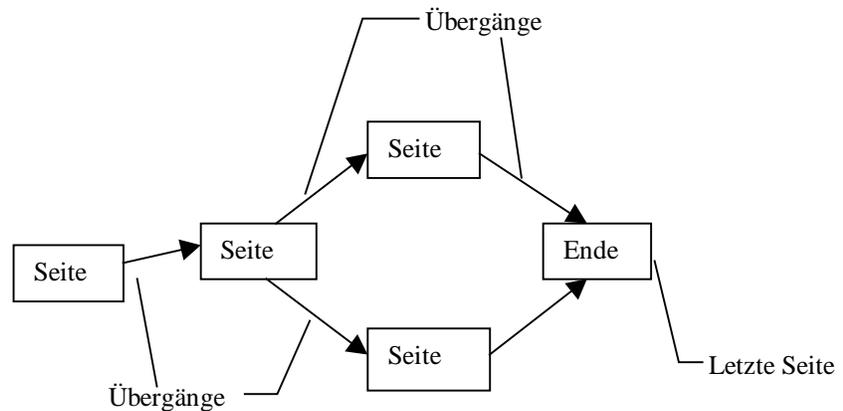
Strukturmodelle

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.
 - Die Eigenschaften entsprechen dem Typ **LONG** (Zahl) oder **STRING** (Zeichenfolge).
 - Der Wert einer Eigenschaft wird entweder mit Hilfe einer Konstanten oder über ein BASIC-Skript festgelegt.

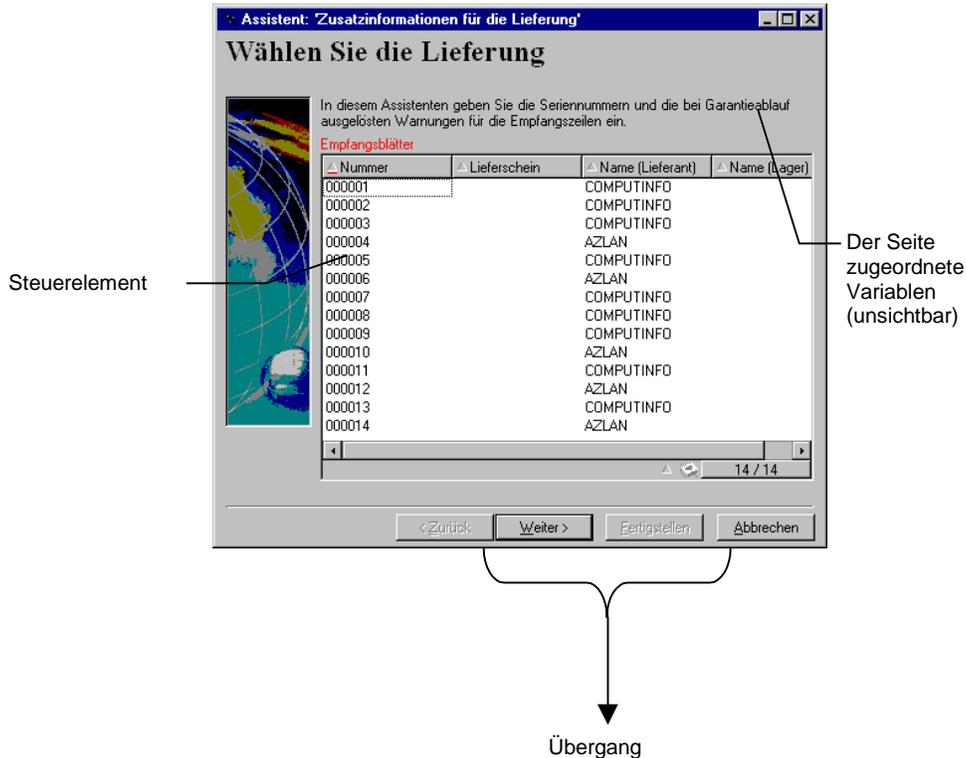
Assistenten setzen sich aus mehreren Seiten zusammen, die durch Übergänge miteinander verbunden sind. Die Verzweigung von einer Seite zur nächsten wird durch die eingegebenen Informationen oder die vom Benutzer getroffenen Auswahlen bestimmt. Die nachstehende Abbildung zeigt das Strukturmodell eines Assistenten:



Modell der Seite eines Assistenten

Die Seite eines Assistenten besteht aus folgenden Elementen:

Seite



Allgemeine Informationen zu Assistenten

Der Code eines Assistenten wird im Feld **Assistentenskript** (SQL-Name: WizardScript) 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.

Warnung: 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.

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**:
 - ❖ $\langle \text{Name der Eigenschaft} \rangle = \langle \text{Text} \rangle$
- 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>}

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 einzeiliges oder mehrzeiliges BASIC-Programm, das einen Wert in der globalen Variablen **RetVal** zurückgibt. Im Fall eines einzeiligen 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 einzeiligen Skripts

```
<Name der Eigenschaft>=<Skript>
```

Beispiel:

```
Variable="Der Name lautet: " & {Name}
```

Das zuvor gezeigte einzeilige Skript ist identisch mit dem nun folgenden mehrzeiligen Skript:

```
{Variable=  
RetVal="Der Name lautet: " & {Name}  
}
```

Syntax eines mehrzeiligen Skripts

```
{ <Name der Eigenschaft>=  
  <Skript>  
}
```

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 lautet wie folgt:

```
{Knoten.Knoten.Knoten[.Eigenschaft][.Methode()]}
```

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()
```

Die nachstehende Tabelle zeigt die Eigenschaften, die über eine Methode verfügen:

Liste der verwendbaren Methoden		
Betroffene Eigenschaft	Methode	Beschreibung
DBLISTBOX.TABLE	LABEL()	Zurückgeben der Bezeichnung der Tabelle, über die das Steuerelement DBLISTBOX generiert werden kann.
DBEDIT.TABLE	LABEL()	Zurückgeben der Bezeichnung der Tabelle, über die das Steuerelement DBEDIT generiert werden kann.
DBEDIT.FIELD	LABEL()	Zurückgeben der Bezeichnung des Felds der Tabelle, über die das Steuerelement DBEDIT generiert werden kann.
LISTBOX	COUNT()	Zurückgeben der Anzahl der Zeilen des Steuerelements LISTBOX.
LISTBOX	CELL(i,j)	Zurückgeben des Inhalts der Zelle (i,j) der Eigenschaft vom Typ Tabelle .
LISTBOX	VALUE(i)	Zurückgeben des Werts der Zeile i der Eigenschaft vom Typ Tabelle .
LISTBOX	LINE(i)	Zurückgeben des Werts der Zeile i der Eigenschaft vom Typ Tabelle .
LINKEDIT.TABLE	LABEL()	Zurückgeben der Bezeichnung der Tabelle, über die das Steuerelement LINKEDIT generiert werden kann.

Eigenschaft vom Typ **Tabelle**

Die Eigenschaften vom Typ **Tabelle** sind die Eigenschaften, deren Wert im nachstehenden Format definiert ist:

```
<Spalte|Spalte|Spalte|...>=<Wert der Zeile>,
<Spalte|Spalte|Spalte|...>=<Wert der Zeile>, ...
```

Der <Wert der Zeile> entspricht der Kennung (**Id**) des entsprechenden Datensatzes.

Der Wert dieser Eigenschaften kann in Form einer Tabelle angezeigt werden:

		Spalte 1	Spalte 2	Spalte 3
Nummer der Zeile: 1	Wert der Zeile (Bsp.: 18)	Zelle (1,1)	Zelle (2,1)	Zelle (3,1)
Nummer der Zeile: 2	Wert der Zeile (Bsp: 29)	Zelle (1,2)	Zelle (2,2)	Zelle (3,2)
Nummer der Zeile: 3	Wert der Zeile (Bsp: 78)	Zelle (1,3)	Zelle (2,3)	Zelle (3,3)
...

Beispiel

Wir nehmen eine Eigenschaft **VALUES**, 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 also davon aus, dass diese Eigenschaft den folgenden Wert aufweist:

```
VALUES= "Colombo|Gerald=32,Lübeck|Alexandre=64,Daquin|William=24"
```

Dieser Wert kann in Form einer Tabelle angezeigt werden:

		Name	Vorname
1	32	Colombo	Gerald
2	64	Lübeck	Alexandre
3	24	Daquin	William

Die globalen Variablen CTXTTABLE und CTXTSELECTION verwenden

Der Inhalt dieser Variablen kann unter Verwendung der nachstehenden Syntax abgerufen werden:

```
[ CTXTTABLE ]
[ CTXTSELECTION ]
```

Die folgende Tabelle zeigt die Merkmale dieser beiden Variablen:

Name der Variablen	Beschreibung der Variablen	Kommentar
CTXTTABLE	Enthält die Tabelle, die beim Starten des Assistenten aktiv ist. Variable vom Typ String .	Diese Variable wird von AssetCenter automatisch abgerufen. Der Benutzer kann den Wert nicht erzwingen.
CTXTSELECTION	Enthält die Liste der internen Datensatzkennungen, die zum Zeitpunkt des Starts des Assistenten ausgewählt waren. Variable vom Typ String .	Diese Variable wird von AssetCenter automatisch abgerufen. Der Benutzer kann den Wert nicht erzwingen.

Verkettung von Assistenten

Ein Assistent kann 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 **Ausführender 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()
- AmProgress()
- AmLog()
- AmMsgBox()
- AmExecTransition()

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 Knotens 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 allgemeinen Eigenschaften aufgeführt, die in einem Knoten vom Typ *Root* definiert werden können:

Logische Eigenschaften des Knotens <i>Root</i>			
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
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.

Logische Eigenschaften des Knotens <i>Root</i>			
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
GLOBAL=<Skript>	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. Eigenschaft vom Typ Script .	{GLOBAL=Dim Filter As String}	
SERIALIZE=<TRUE FALSE>	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 ini . Eigenschaft vom Typ Boolean .	SERIALIZE=TRUE	Standardmäßig gilt für diese Eigenschaft der Wert FALSE .

Logische Eigenschaften des Knotens <i>Root</i>			
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
Physische Eigenschaften des Knotens <i>Root</i>			
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 auch kein Bild. Der Pfad der grafischen Datei ergibt sich aus dem AssetCenter-Verzeichnis Config . 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	
HEIGHT=<Höhe>	Definiert die Höhe (< Height >), die standardmäßig für das Assistentenfenster gelten soll. Die Angaben werden in Twips gemacht. Eigenschaft des Typs Long .	HEIGHT=5000	

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.

Dem Knoten <i>Root</i> untergeordnete Knoten	
Typ des Knotens	Beschreibung
PAGE <Name der Seite>	Beschreibt eine Seite des Assistenten.
TRANSITION <Name des Übergangs>	Beschreibt einen Übergang zwischen zwei Seiten.
FINISH <Name des Knotens>	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.
START <Name des Knotens>	Enthält beispielsweise ein Skript, das beim Starten des Assistenten über die Eigenschaft DO ausgeführt werden soll, sowie den Namen der Startseite des Assistenten (Eigenschaft TO).

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 mit den 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 allgemeinen Eigenschaften aufgeführt, die in einem Knoten vom Typ Page definiert werden können:

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. Dieser Titel erscheint fett gedruckt in der oberen Hälfte der Seite. 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 ausgeführt wird. Eigenschaft vom Typ Skript .	{ONENTER=AmMsgBox("Hello")}	
Physische Eigenschaft eines Knotens vom Typ Page			
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
IMAGE="<Pfad der Bitmap-Datei>" IMAGE16="<Pfad der Bitmap-Datei >"	Definiert die Bitmap-Datei (.bmp), die auf der Seite des Assistenten angezeigt werden soll. Eigenschaft vom Typ Skript .	IMAGE="C:\Bilder\Seitel.bmp"	Wenn für diese Eigenschaft kein Wert definiert wurde, erbt sie den Wert der Eigenschaft IMAGE des Knotens vom Typ Root . 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:

Dem Knoten <i>Page</i> untergeordnete Knoten	
Typ des Knotens / Objekt	Beschreibung
<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.

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.

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:

Logische Eigenschaften eines Knotens vom Typ <i>Transition</i>			
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
FROM="<Name der ursprünglichen Seite>"	Definiert die ursprüngliche Seite des Übergangs. Eigenschaft vom Typ String .	FROM="Seite2"	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 die Zielseite des Übergangs. Eigenschaft vom Typ String .	TO="Seite3"	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. Frage 19	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 Skript .	{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.

Logische Eigenschaft vom Typ <i>Finish</i>			
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
CHAIN=< SQL-Name des auszuführenden Assistenten>	Definiert den Assistenten, der am Ende des aktuellen Assistenten ausgeführt werden soll. Eigenschaft vom Typ String	CHAIN= "Umzug"	

Der Knoten vom Typ **PARAMS** ermöglicht die Weitergabe der Parameter an den Assistenten, der ausgeführt werden soll.

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**.

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.

Logische Eigenschaft vom Typ <i>Long</i> oder <i>String</i>			
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
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	

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 Positionieren 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. Im folgenden Abschnitt finden Sie Informationen zu den nachstehenden Punkten:

- Gemeinsame Eigenschaften
- Das Steuerelement CheckBox
- Das Steuerelement ComboBox
- Das Steuerelement ListBox
- Das Steuerelement Label
- Das Steuerelement ProgressBar
- Das Steuerelement CommandButton
- Das Steuerelement OptionButtons
- Das Steuerelement DBListBox
- Das Steuerelement DBQueryBox
- Das Steuerelement DBEdit
- Das Steuerelement DBTable
- Das Steuerelement DBPath
- Das Steuerelement LinkEdit
- Das Steuerelement TextBox

Gemeinsame Eigenschaften

In der nachstehenden Tabelle werden die optionalen Eigenschaften aufgeführt, die allen Steuerelementen gemein sind:

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. Eigenschaft vom Typ Boolean .	MANDATORY=TRUE	Für die Steuerelemente CHECKBOX und LABEL ist diese Eigenschaft nicht verfügbar.
VALUE=<Wert>	Definiert den Standardwert des Steuerelements bei seiner Erstellung. Der <Wert> ergibt sich aus dem betroffenen Steuerelement. Eigenschaft vom Typ Boolean oder String .	Wenn es sich beispielsweise um das Steuerelement CHECKBOX handelt, kann der <Wert> TRUE oder FALSE entsprechen.	
Logische 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 (=FALSE). Eigenschaft vom Typ Boolean .	Label1.Visible=TRUE	
ENABLED=<TRUE FALSE>	Definiert die Aktivierung des Steuerelements (=TRUE) oder (=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	
LABEL=" <Text der Beschreibung> "	Definiert einen optionalen Text, der über dem Steuerelement erscheint. Eigenschaft vom Typ String .	Choice1.Label="Auswahl der Person"	Diese Beschreibung unterstützt das Format HTML.

Das Steuerelement *CheckBox*

Das Steuerelement **CHECKBOX** definiert ein Kontrollkästchen.

Eigenschaften

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften erkennt das Steuerelement **CHECKBOX** die nachstehende Eigenschaft:

Eigenschaft des Steuerelements <i>CHECKBOX</i>		
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
TEXT=" <Text> "	Definiert den Text des Kontrollkästchens. 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:

Physische Eigenschaften des Steuerelements <i>COMBOBOX</i>			
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
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 Vermögensgegenstände=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"

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:

Physische Eigenschaften des Steuerelements <i>LISTBOX</i>			
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
LISTHEIGHT=<Prozentsatz>	Definiert die relative Größe des Steuerelements LISTBOX im Verhältnis zu den anderen Steuerelementen vom Typ LISTBOX , die im Assistenten vorhanden sind. Eigenschaft vom Typ 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 Überschrift und die Eigenschaften der Spalten in der Liste. Spalte definiert den in der Spalte erscheinenden Text. Eigenschaft vom Typ String .	COLTITLE="Name FirmName"	
COLWIDTH=" <Breite Breite Breite...> "	Definiert die Breite der Spalte proportional zur Breite des Steuerelements. Die Angaben erfolgen in Prozenten. Eigenschaft vom Typ String .	COLWIDTH="50 50"	

Physische Eigenschaften des Steuerelements <i>LISTBOX</i>			
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 Vermögensgegenstände=asset, Benutzer=user, "	Wenn Sie den Wert nicht eingeben, wird er von AssetCenter automatisch zugeordnet. Ein Beispiel: VALUES="A,B,C" entspricht VALUES="A=1,B=2,C=3" 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.

Obligatorische, logische Eigenschaft des Steuerelements <i>LISTBOX</i>		
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
TABLE<Name der Tabelle>	Name der Tabelle, die zum Extrahieren der Spaltenüberschriften verwendet wird. Eigenschaft vom Typ String .	TABLE"amAsset "
COLNAME=<Titel Titel ...>	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. AssetCenter greift bevorzugt auf die Werte der Eigenschaft COLTITLE zurück, um die Spaltenüberschrift zu definieren, sofern eine solche Überschrift vorhanden ist. Eigenschaft vom Typ String .	COLNAME="Name FirstName "
EDIT<"boolesch boolesch ... boolesch">	Legt fest, ob die Liste für die einzelnen Spalten bearbeitbar ist. Eigenschaft vom Typ String .	EDIT"1 0 1 1"

Das Steuerelement *Label*

Das Steuerelement **LABEL** definiert lediglich eine Bezeichnung. Dieses Element verfügt über die nachstehende Eigenschaft:

Physische Eigenschaft des Steuerelements <i>LABEL</i>			
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 *OptionButtons*

Das Steuerelement **OPTIONBUTTONS** definiert die Gruppe der Optionsfelder.

Eigenschaften

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften erkennt das Steuerelement **OPTIONBUTTONS** die nachstehenden Eigenschaften:

Physische Eigenschaften des Steuerelements <i>OPTIONBUTTONS</i>		
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
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 Vermögensgegenstände=asset, Benutzer=user"
BORDER=<TRUE FALSE>	Legt fest, ob die Gruppe der Optionsfelder umrahmt wird (=TRUE) oder nicht (=FALSE) Eigenschaft vom Typ Boolean .	BORDER= TRUE

Das Steuerelement *ProgressBar*

Das Steuerelement **PROGRESSBAR** definiert die Verlaufsanzeige.

Eigenschaften

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften erkennt das Steuerelement **PROGRESSBAR** die nachstehende Eigenschaft:

Physische Eigenschaften des Steuerelements <i>PROGRESSBAR</i>		
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 vom Typ 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:

Physische Eigenschaften des Steuerelements <i>COMMANDBUTTON</i>		
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
WIDTH=<Breite>	Definiert die Breite der Schaltfläche in Twips. Eigenschaft vom Typ Long .	WIDTH=250
HEIGHT=<Höhe>	Definiert die Höhe der Schaltfläche in Twips. Eigenschaft vom Typ Long .	HEIGHT=125
CAPTION=<Text>	Definiert den Text innerhalb der Schaltfläche. Eigenschaft vom Typ String .	CAPTION="Starten"
CLICK=<BASIC-Skript >	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.

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:

Physische Eigenschaften des Steuerelements <i>DBLISTBOX</i>			
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
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 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 ...>"	Definiert das Anzeigeformat für die aus der Datenbank extrahierten Daten, und zwar als Prozentsatz der Gesamtgröße des Steuerelements DBLISTBOX . Eigenschaft vom Typ String .	COLWIDTH="40 60"	

Physische Eigenschaften des Steuerelements <i>DBLISTBOX</i>			
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
LISTHEIGHT=<Prozentwert>	Definiert die relative Größe des Steuerelements DBLISTBOX im Verhältnis zu den anderen Steuerelementen vom Typ DBLISTBOX , die im Assistenten vorhanden sind. Eigenschaft vom Typ 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.
TREE=<TRUE FALSE>	Zeigt die Daten im Format einer Baumstruktur (=TRUE) an 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 (=TRUE) unterstützt oder nicht (=FALSE). Eigenschaft vom Typ Boolean .	MULTISEL=TRUE	
DBCLICK=<TRUE FALSE>	Wenn für diese Eigenschaft der Wert TRUE gilt, simuliert AssetCenter einen Mausklick auf die Schaltfläche  der aktuellen Seite.	DBCLICK=FALSE	

Physische Eigenschaften des Steuerelements <i>DBLISTBOX</i>			
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
FILTER=" <Klausel> "	Definiert die AQL-Klausel WHERE, um die Datensätzen zu filtern, die bei der Abfrage bearbeitet werden sollen. Eigenschaft vom Typ String .	FILTER="U ser.lEmpl DeptId='C olombo, Gerard' "	

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:

Physische Eigenschaften des Steuerelements <i>DBQUERYBOX</i>			
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
QUERY=" <Vollständige AQL-Abfrage> "	Definiert die AQL-Abfrage, die die im Steuerelement DBLIST anzuzeigenden Informationen zurückgibt. Eigenschaft vom Typ String .	QUERY="SELECT Name, FirstName 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 "	

Physische Eigenschaften des Steuerelements <i>DBQUERYBOX</i>			
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
LISTHEIGHT=<Prozentsatz>	Definiert die relative Größe des Steuerelements DBQUERYBOX im Verhältnis zu den anderen Steuerelementen vom Typ DBQUERYBOX , die im Assistenten vorhanden sind. Eigenschaft vom Typ Long .	LISTHEIGHT=50	Wenn zwei Steuerelemente vom Typ DBQUERYBOX 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>	Zeigt die Daten im Format einer Baumstruktur (=TRUE) an 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 (=TRUE) unterstützt oder nicht (=FALSE).	MULTISEL=TRUE	

Physische Eigenschaften des Steuerelements <i>DBQUERYBOX</i>			
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel	Kommentar
DBCLICK=<TRUE FALSE>	Wenn für diese Eigenschaft der Wert TRUE gilt, simuliert AssetCenter einen Mausklick auf die Schaltfläche  der aktuellen Seite. Eigenschaft vom Typ Boolean .	DBCLICK=F ALSE	

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, ...).

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).

Eigenschaft

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften muss das Steuerelement **DBLIST** obligatorisch die nachstehenden Eigenschaften aufweisen:

Obligatorische, physische Eigenschaften des Steuerelements <i>DBEDIT</i> im Modus <i>Normal</i>		
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
TABLE "<SQL-Name der Tabelle>"	Name der Tabelle mit dem Feld, dessen Steuerelement kopiert werden soll. Eigenschaft vom Typ String .	TABLEamAsset
FIELD="<SQL-Name des Felds>"	Name des Felds, dessen Steuerelement kopiert werden soll. Eigenschaft vom Typ String .	FIELD=seAcquMethod

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 Felds der AssetCenter-Datenbank.

Eigenschaften

Neben den allen Steuerelementen gemeinsamen optionalen Eigenschaften muss das Steuerelement **DBPATH** obligatorisch die nachstehenden Eigenschaften aufweisen:

Obligatorische, physische Eigenschaften des Steuerelements <i>DBPATH</i>		
Name der Eigenschaft=Wert	Beschreibung der Eigenschaft	Beispiel
TABLE"<SQL-Name der Tabelle>"	Name der Tabelle, in der ein Feld ausgewählt werden soll. Eigenschaft vom Typ String .	TABLEamAsset

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:

Obligatorische, physische Eigenschaften des Steuerelements <i>LINKEDIT</i>		
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 .	TABLEamAsset
FILTER=<Klausel WHERE einer AQL-Abfrage>	Definiert einen Filter auf der Basis einer AQL-Abfrage. Eigenschaft vom Typ String .	Diese Eigenschaft ist optional.

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:

Physische Eigenschaft des Steuerelements <i>TEXTBOX</i>		
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

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.

- Schritt 1 – Analysieren der Anforderungen
- Schritt 2 – Definieren der Struktur des Assistenten
- 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:

- ↪ Identifizieren Sie die Vermögensgegenstände, die an einen anderen Standort gebracht werden sollen.
- ↪ 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 Vermögenstabelle.
- Identifizieren der Gegenstände anhand ihres Standorts. Sie wählen zuerst den Standort und dann die Gegenstände, die beim Umzug berücksichtigt werden sollen.

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.

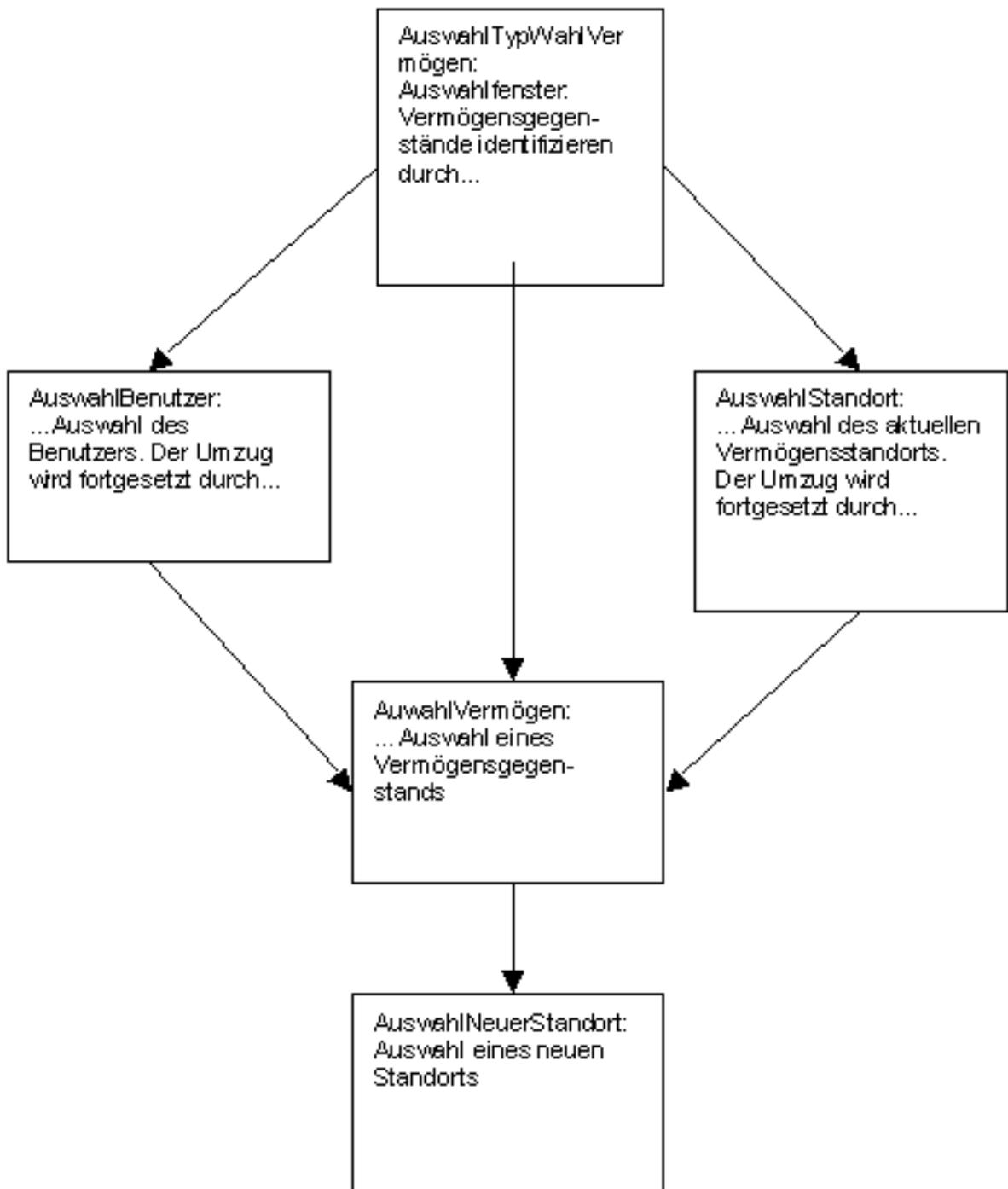
Schritt 2 – Definieren der Struktur des Assistenten

In diesem Schritt müssen Sie je nach den im Schritt 1 definierten Anforderungen die Struktur des Assistenten festlegen. Dazu gehören folgende Elemente:

- Die Anzahl der Seiten
- Die Abfolge der verschiedenen Seiten
- Der Inhalt einer jeden Seite

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:



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 CHOICEBOX
AuswahlVermögen	Der Benutzer kann die Vermögensgegenstände in der Liste der Datensätze in der Tabelle mit den Vermögensgegenstände 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

```

```

context 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
amEmplDept 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
FullName LIKE LikeParam(amEmplDept_2:FullName)+'%' AND
amEmplDept_2:lEmplDeptId 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.VALUES(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

    ErrHandler:
    On Error Goto 0
    AmLog(AmLastError() & " - " & AmLastErrorMsg())
    AmLog("Die Transaktion wurde abgebrochen")
    RetVal = 1
    Exit function
  }
}
SUMMARY = 1
}
}

```

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

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.

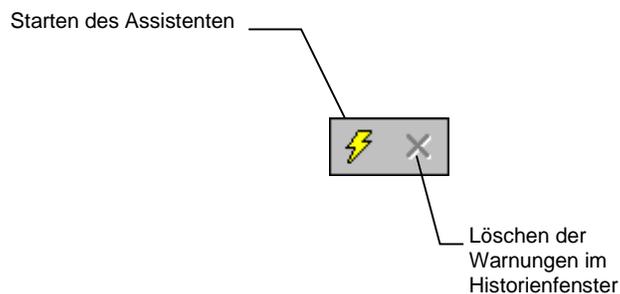
Befehle zur Bearbeitung

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:



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.

Im Textmodus erfolgt die Suche unter Berücksichtigung des vollständigen Textes. Im Grafikmodus bezieht sich die Suche nur auf den Namen der Eigenschaft.

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 Farbcode erzwingen. Name und Wert erscheinen daraufhin z. B. 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.

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

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.

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.

Bei kontextbezogenen Assistenten steht diese Schaltfläche nicht zur Verfügung.

Häufig gestellte Fragen

In diesem Kapitel finden Sie die Antworten auf die Fragen, die bei der Erstellung eines Assistenten am häufigsten auftauchen.

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)
```

Frage

Der nachstehende Code, mit dem der Wert einer Eigenschaft getestet wird, scheint nicht zu funktionieren:

```
If {checkbox}=TRUE Then ...
```

Antwort

Zum Lesen des Werts einer Eigenschaft sollten Sie die Syntax **{Eigenschaft}** oder die Funktion **AmValueOf(Eigenschaft)** verwenden. Der Wert wird im Format **String** (Zeichenfolge) zurückgegeben, dessen Typ mit einem booleschen Vergleich (**TRUE** oder **FALSE**) nicht kompatibel ist. Der Code lautet richtig:

```
If {checkbox} Then ...
```

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:

- Sammeln von Informationen mit Hilfe aufeinander folgender Seiten. Der Schreibzugriff ist auch für den AssetCenter-Administrator unzulässig.
- 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 Vermögenstabelle innerhalb eines Assistenten:

```
Dim irc as Long  
irc = AmActionDDE("aam", "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.

Kapitel 24 - Einstellen der Leistungsstärke von AssetCenter in einem WAN

WANs zeichnen sich durch folgende Eigenschaften aus:

- Geringe Übertragungsraten
- Hohe Latenzzeiten

Durch eine geeignete Konfiguration von AssetCenter können Sie diese Nachteile ausgleichen. Diese Konfiguration führen jedoch zu einer Beeinträchtigung bestimmter AssetCenter-Funktionen.

In diesem Kapitel finden Sie einige Hinweise, mit denen Sie WANs besser nutzen können. Dabei ist es jedoch besonders wichtig, dass Sie Tests durchführen, um den Mittelweg zwischen der optimalen Antwortzeit der Anwendung und dem Verlust bestimmter Funktionalitäten zu finden.

Optionen des Menü *Extras/ Optionen*

Die Einschränkung der Zugriffsdauer auf die Datenbank erfolgt über die nachstehenden Optionen:

- Option **Schreibvorlauf nach** auf der Registerkarte **Navigation**: Sie können den Schreibvorlauf entweder deaktivieren oder eine besonders lange Dauer eingeben, wie z. B. den Schreibvorlauf nach Ablauf von 10 000 ms).
- Option **Baumstruktur in Dropdown-Listen** auf der Registerkarte **Navigation**: Sie können diese Option deaktivieren, da die Anzeige von Baumstrukturen mehr Speicherplatz in Anspruch nimmt als die Anzeige von Listen.

Das geht jedoch auf Kosten der Benutzerfreundlichkeit von Baumstrukturen in Dropdown-Listen.

Sie können den Informationsfluß zwischen der Server- und der Client-Station der Datenbank mit Hilfe der nachstehenden Optionen einschränken:

- Optionen **Nicht laden für** und **Kein Laden von mehr als** in den Registerkarten mit den Namen **Listen** (Hauptliste und andere): Wir empfehlen Ihnen, die Anzahl der zu ladenden Zeilen einzuschränken. So können Sie beispielsweise festlegen, dass in den Hauptlisten nicht mehr als 50 Zeilen, und in den Nebenlisten nicht mehr als 15 Zeilen geladen werden sollen. Es steht Ihnen frei, die Anzahl der Zeilen festzulegen, die geladen werden sollen. Beachten Sie dabei die auf die angezeigten Listen angewendeten Filter, und die Chancen, die gewünschten Informationen in einer vorgegebenen Anzahl von Zeilen zu finden.
- **Neue Nachrichten testen**, Registerkarte **Nachrichtensystem**: Sie können sich dafür entscheiden, die neuen Nachrichten nur beim Herstellen einer Verbindung mit der Datenbank oder aber in regelmäßigen Abständen, beispielsweise alle 10 Minuten zu lesen.
- Registerkarte **Caches**: In dieser Registerkarte können Sie die Auffrischungsintervalle (Spalte **Alle**) der Cache-Speicher festlegen. Sie können sich jedoch auch dafür entscheiden, die Cache-Speicher während einer AssetCenter-Sitzung nicht aufzufrischen. In diesem Fall erfolgt die Auffrischung nur bei der Herstellung der Verbindung.

Wenn Sie die Cache-Speicher nicht regelmäßig auffrischen, sind die auf dem Bildschirm erscheinenden Daten möglicherweise nicht aktuell. Die meisten der von den Cache-Speichern betroffenen Daten werden jedoch bei der Installation von AssetCenter eingerichtet, und unterliegen den täglich vorgenommenen Änderungen nur in geringem Maße. (Liste der Aufzählungen, Wörterbuch der Merkmale, Währungen, Arbeitstagekalender usw.).

Listen sortieren

Die Hauptlisten und Registerkartenlisten können mit Hilfe des Kontextmenüs und der Option **Liste konfigurieren** in verschiedenen Situationen konfiguriert werden:

- Listen, die über die Menüs zum Zugriff auf die Tabellen angezeigt werden (Menü **Ausrüstung/ Vermögensgegenstände**)
- Listen, die über die Ansichten angezeigt werden (Menü **Extras/ Ansichten**)
- Auswahllisten (Kontextmenü und Option **Verknüpfung wählen**).

Das Sortieren dieser Listen kann auf zwei verschiedene Weisen erfolgen:

- Auswählen eigener Sortierkriterien (Spalte **Sortieren**)
- Verwenden vordefinierter Indizes (Feld **Nach Index**)

Bei der Leistungsfähigkeit der Anzeige machen sich zum Teil geringfügige Unterschiede bemerkbar, ohne dass es möglich wäre, die beste Vorgehensweise vorherzubestimmen.

Probieren Sie einfach die beiden Möglichkeiten für die AssetCenter-Listen aus, und entscheiden Sie sich für die Ihren Anforderungen am ehesten entsprechende Methode.

Vereinfachung der Anzeige

Um die Antwortzeit der Anwendung zu erhöhen, können Sie auch die Anzahl der auf dem Bildschirm erscheinenden Informationen einschränken, indem Sie nur die tatsächlich erforderlichen Listen und Registerkarten einblenden.

Verbindungs-Cache

Aktivieren Sie den Verbindungs-Cache über die Registerkarte **Cache** im Detailfenster der Verbindung:

Die Aktivierung des Verbindungs-Caches bietet nachstehende Vorteile:

- Reduzieren der Verbindungsdauer mit der Datenbank
- Gewinnen von Zeit bei der Verwendung von Bildern und Symbolen

Die standardmäßig vorgeschlagene Cache-Größe ist in der Regel gut angepasst.

Zugriffsbeschränkung

Die Schnelligkeit bei der Anzeige von Daten wird durch das Vorhandensein von Zugriffsbeschränkungen für das mit der Datenbank verbundene Login beeinträchtigt. Diese Beeinträchtigung ist auf den Test zurückzuführen, den AssetCenter vor der Anzeige der Daten durchführt.

Im Zweifelsfall sollten Sie die Liste bzw. das Detail mit einem Login anzeigen, für das keine Zugriffsbeschränkungen vorhanden sind, und die Leistung bei der Anzeige vergleichen.

Löschen Sie gegebenenfalls die fakultativen Zugriffsbeschränkungen.

Parameter einer Station auf eine andere Station anwenden

Wenn Sie mit der Optimierung der Leistung Ihrer Station fertig sind, brauchen Sie nur noch die Änderung der Parameter auf die anderen Stationen in Ihrem Unternehmen zu übertragen.

Um Zeit zu gewinnen, können Sie die ***.ini**-Dateien mit den von Ihnen darin vorgenommenen Änderungen kopieren.

Weitere Informationen zur Liste, ihrem Inhalt und dem Standort der *.ini-Dateien finden Sie im Handbuch "Referenzhandbuch: Verwaltung und Verwendung für fortgeschrittene Benutzer", Kapitel ".ini-Dateien".

Kapitel 25 - .ini-Dateien

Die Konfigurationsdateien für AssetCenter-Programme tragen die Erweiterung **.ini**.

In diesem Kapitel finden Sie Informationen zu den folgenden Punkten:

- Liste der verfügbaren .ini-Dateien
- .ini-Dateien ändern

Liste der verfügbaren *.ini*-Dateien

Nachstehend finden Sie eine Liste der verfügbaren **.ini**-Dateien:

Programm	.ini-Datei	Beschreibung
AssetCenter aam32.exe aam16.exe	aamdisk30.ini	Anzeigeoptionen des Benutzers Wenn Sie erneut die Standard-Anzeigeoptionen für alle Fenster verwenden möchten, können Sie diese Datei löschen.
	aam.ini	Optionen für die Verwendung des AssetCenter-Programms
AssetCenter Database Administrator amdba32.exe	amdba32.ini	Optionen für die Verwendung des AssetCenter Database Administrators Anzeigeoptionen des Benutzers
AssetCenter Export amexp32.exe amexpl32.exe	amexp32.ini amexpl32.ini	Optionen für die Verwendung von AssetCenter Export Anzeigeoptionen des Benutzers
AssetCenter Server aamsrv32.exe	amsrv.ini	Optionen für die Verwendung von AssetCenter Server Anzeigeoptionen des Benutzers
	amsrvcf.ini	Parameter zum Ausführen von AssetCenter Server als Web Server
AssetCenter Web Server	amw3.ini	Parameter zum Ausführen von AssetCenter Web Server
aamapi30.dll	aamapi30.ini	Optionen für die Verwendung des Programms
Alle vorstehenden Programme	amdb.ini	Liste der Verbindungen zu den Datenbanken

Alle **.ini**-Dateien mit Ausnahme der folgenden Dateien sind im Windows Installationsverzeichnis gespeichert:

- Die Datei **amsrvcf.ini** ist im Unterverzeichnis **amsrv/bin32** des AssetCenter-Installationsverzeichnisses abgelegt.
- Die Datei **amw3.ini** ist im Verzeichnis **websrv/bin32** des AssetCenter-Installationsverzeichnisses gespeichert.

Alle AssetCenter-Rechner weisen eigene **.ini**-Dateien auf.

.ini-Dateien ändern

Die Einträge in den **.ini**-Dateien können wie folgt geändert werden:

- Durch Änderung der Software: Die Änderungen werden beim Bestätigen oder beim Verlassen der Software gespeichert. Wird die Software im letztgenannten Fall nicht über die Option **Datei/Beenden** beendet, werden die Änderungen nicht gespeichert.
- Manuell

Es ist ratsam, die Einträge der **.ini**-Dateien mit Hilfe von AssetCenter zu ändern.

In einigen Fällen ist dies jedoch nicht möglich und die Einträge müssen manuell erstellt und geändert werden.

Warnung: Da **ini**-Dateien für den Ablauf der Software essentiell sind, sollten sie nur von Mitarbeitern geändert werden, die entsprechende Kompetenzen aufweisen.

In den folgenden Tabellen werden die Einträge in den **ini**-Dateien beschrieben, die nur manuell geändert werden können.

Warnung: Die im Folgenden aufgeführten **ini**-Einträge sind nicht vollständig, sondern stellen lediglich eine Auswahl dar. Alle nicht beschriebenen Abschnitte und Einträge dürfen auf keinen Fall manuell geändert werden.

Bestimmte boolesche Einträge werden mit den Werten **1** oder **0** beschrieben. Dabei steht der Wert **1** für **True** und der Wert **0** für **False**.

Einträge der Datei *aam.ini*

Abschnitt [OPTION]

Eintrag	Bedeutung
bSaveOptionOnExit	Definieren Sie den Wert 0 für diesen Eintrag, wenn die an den Einträgen des Abschnitts [Option] mit Hilfe der Software vorgenommenen Änderungen beim Verlassen von AssetCenter nicht gespeichert werden sollen. Standardmäßig werden die Änderungen gespeichert.
CAdbLinkEdit.opt_bCreation	Definiert, ob die Schnellerstellung verknüpfter Datensätze möglich ist: 0: Schnellerstellung ist nicht möglich 1: Schnellerstellung ist möglich Standardwert: 1
CmdComboLines	Begrenzt die Anzahl der in den Ansichts- und Aktionslisten angezeigten Zeilen, die über die Symbolleiste aufgerufen werden.
CNTbkTabCfg.bShowFlyby	Anzeigen der Hilfetexte auf den Registerkarten der Detailfenster: 0: Nicht anzeigen 1: Anzeigen
KeyIniFileName	Gibt den Pfad der Datei aamdisk30.ini an. Beispiel: KeyIniFileName=aamdisk30.ini Diese Option wird verwendet, wenn die AssetCenter-Benutzer die gleiche aamdisk30.ini -Datei verwenden sollen, die beispielsweise auf einer Festplatte im Netz gespeichert ist. Die Datei kann schreibgeschützt werden, damit die Benutzer ihre Konfiguration nicht ändern können.
NewMailLastCheck	Zeitpunkt, zu dem die AssetCenter-Nachrichten zum letzten Mal gelesen wurden. Einheit: Sekunden seit dem 1. Januar, 00:00 h.
opt_bAskForConcurrentModifications	Dieser Eintrag legt fest, ob AssetCenter eine Bestätigung anfordert, wenn der Benutzer auf die Schaltfläche Ändern klickt während ein anderer Benutzer den gleichen Datensatz ändert. 1: Bestätigungsfenster wird angezeigt 0: Bestätigungsfenster wird nicht angezeigt und Änderungen werden gespeichert
opt_bCommitDeletesOneByOne	Diese Option wird beim Löschen von Datensätzen eingesetzt. Ist die Option aktiv, löscht AssetCenter die Datensätze nacheinander (eine Transaktion pro gelöschtem Datensatz). Ist sie nicht aktiv, werden alle Datensätze in einer Transaktion gelöscht. Standardwert: 0.
opt_ImportCacheSize	Beim Importieren von Daten mit Hilfe von

Eintrag	Bedeutung
	Abstimmungsschlüsseln gibt dieser Eintrag die Größe des Cache-Speichers zur Beschleunigung des Imports an. Einheit: Anzahl der gefundenen Datensätze Standardwert: 100
StartSunday	Legt fest, ob die Woche am Montag (StartSunday=0) oder am Sonntag (StartSunday=1) beginnt. Diese Option wird in den Kalendern eingesetzt.

Abschnitt [SQL]

Eintrag	Bedeutung
OracleDLL	Ermöglicht die Angabe der zu ladenden Oracle DLL für die Kommunikation mit Oracle.

Einträge der Datei *amsrv.ini*

Abschnitt [OPTION]

Eintrag	Bedeutung
MaxRentPerTrans	Dieser Eintrag dient zur Generierung der Miet- bzw. Leasingraten. Er legt die maximale Anzahl der Ratenberechnungen pro Transaktion fest. Standardwert: 200
MaxMsgInList	Definiert die Anzahl der in der Liste des Hauptfensters von AssetCenter Server angezeigten Zeilen. Standardwert: 5000
<Module>LastCheck oder <Module> kann folgende Werte annehmen: Alarms, CostCenter, History, LostVal, Rent, Stats, Stock, TimeZone, UpdateToken, WkGroup, WkGroup <xxx>, WorkflowFinder	Die auf LastCheck endenden Zeilen entsprechen dem letzten Ausführungsdatum des Programms. Sie dienen zum Berechnen des nächsten Programm-Ausführungsdatums beim Start von AssetCenter Server. In einigen Fällen ist es sinnvoll, die Zeile WkGroup <xxx> LastCheck (bzw. die Zeile WkGroupLastCheck) zu löschen, wenn die Ausführungsgruppe <xxx> nicht mehr vorhanden ist (bzw. kein Workflow-Modell ohne Ausführungsgruppe vorhanden ist). Diese Elemente werden von der Software nicht automatisch gelöscht.

Einträge der Datei *amsrvf.ini* et *amw3.ini*

Die Datei **amsrvf.ini** enthält ähnliche Einträge wie die Datei **amw3.ini**. Die Einträge der letzteren Datei sind im Handbuch "AssetCenter Web-Benutzerhandbuch", Kapitel "Administrating AssetCenter Web", Abschnitt "Configuring AssetCenter Web via "amw3.ini" beschrieben.

Einträge der Datei *amexp32.ini*

Abschnitt [OPTION]

Eintrag	Bedeutung
MaxOldDoc	Maximale Anzahl früher verwendeter Dokumente, die im Menü Datei angezeigt werden.

Einträge der Datei *amdb.ini*

Für jeden, eine AssetCenter-Verbindung beschreibenden Abschnitt können folgende Einträge geändert werden:

Eintrag	Bedeutung
AmApiDll	Gibt den Pfad zur Datei aamapi30.dll der AssetCenter-APIs an. Dieser Eintrag ist hilfreich, wenn AssetCenter Web gestartet werden soll und die Datei aamapi30.dll weder im Windows NT Systemordner noch im Ordner des ausführbaren Programms amw3.exe gespeichert ist.
FetchingArraySize	Anzahl der Zeilen, die beim Ausführen von SQL-Befehlen pro Paket abgerufen werden. Standardwert: 80
OdbcLockingTime	Gibt die Zeitspanne an, nach deren Ablauf ein Datensatz in einer Microsoft SQL Server- oder Sybase SQL Anywhere-Datenbank als durch einen anderen Benutzer gesperrt gilt. Einheit: Sekunden Standardwert: 60 Warnung: Ist der Wert zu niedrig, wird der Import auf einem relativ ausgelasteten Server möglicherweise unterbrochen.
OldStyleCatalog	Überschreibt für Oracle-Datenbanken die Verwendung der Standardansicht All_Catalog mit der Ansicht Tab . Der Eintrag kann zwei Werte annehmen: 1: Tab verwenden 0: All_Catalog verwenden
PostConnectSql	Gibt die nach Aufbau der Verbindung zur Datenbank auszuführende SQL-Folge an. Beispiel für die Verwendung: Zum Öffnen einer SQL Anywhere 5.0-Datenbank mit einer SQL Anywhere 5.5-Laufzeitversion muss folgende Zeile am Ende jedes Abschnitts, der eine frühere Datenbankversion als 3.0 beschreibt, hinzugefügt werden: PostConnectSql=SET OPTION NON_KEYWORDS='LOGIN'

