

AssetCenter™

Wersja 3.5

Opis funkcji systemu AssetCenter: Administracja i zaawansowane zastosowania AssetCenter

Część II

listopad 07, 2000

ACT-3,5X-PL-000828



The Infrastructure Management Company™

© Peregrine Systems, Inc., 1999-2000. Wszelkie prawa zastrzeżone.

Sybase SQL Anywhere Runtime: © Sybase, Inc. 1992–1995; Portions © Rational Systems, Inc. 1992–1994.

Informacje podane w niniejszym dokumencie są własnością Peregrine Systems, Inc. i mogą być używane lub ujawniane jedynie za pisemną zgodą firmy Peregrine Systems. Zabrania się reprodukcji niniejszego podręcznika lub dowolnej jego części bez uprzedniego uzyskania pisemnej zgody od firmy Peregrine Systems, Inc.

W niniejszym dokumencie znajdują się liczne nawiązania do produktów z zastosowaniem ich nazw handlowych. W większości, jeśli nie we wszystkich przypadkach, oznaczenia te są znakami towarowymi lub zarejestrowanymi znakami towarowymi należącymi do poszczególnych właścicieli.

Peregrine Systems, ServiceCenter, AssetCenter, InfraCenter for Workgroups i InfraTools są znakami towarowymi Peregrine Systems, Inc.

Programy opisywane w niniejszym podręczniku są dostarczane w ramach licencji lub umowy o nieujawnianiu. Można z nich korzystać lub kopiować wyłącznie z zachowaniem warunków umowy.

Peregrine Systems zastrzega sobie prawo do zmiany informacji podanych w niniejszym dokumencie bez uprzedzenia.

Oprogramowanie może ulec zmianie, dlatego możliwe jest, że dostarczona dokumentacja nie będzie w pełni spójna z dostarczonym oprogramowaniem. Modyfikacje te nie mają ujemnego wpływu na właściwe zrozumienie działania oprogramowania. Dalsze informacje o najbardziej aktualnych zmianach znajdują się w pliku **Readme.txt**.

Nazwy firm i osób zastosowane w demonstracyjnej bazie danych i w przykładach podręcznikowych są fikcyjne, a ich celem jest zilustrowanie działania oprogramowania. Wszelkie podobieństwo do rzeczywistych firm lub osób, istniejących w przeszłości lub obecnie jest czysto przypadkowe.

AssetCenter i InfraCenter for Workgroups - integralność danych

Aplikacje AssetCenter i InfraCenter for Workgroups oferują bardzo zaawansowaną funkcjonalność. Bogactwo funkcji opiera się na skomplikowanej strukturze bazy danych: baza danych zawiera znaczną ilość tabel, pól, łączy i indeksów; niektóre tabele pośrednie nie są wyświetlane w interfejsie użytkownika; pewne łączy, pola i indeksy są automatycznie tworzone, usuwane lub modyfikowane przez oprogramowanie.

Tylko interfejsy zaprojektowane dla AssetCenter i InfraCenter for Workgroups (interfejs graficzny, interfejsy API, program importu, interfejs sieci WWW i bramy) mogą modyfikować bazę danych z zachowaniem jej integralności. **Nigdy nie wolno modyfikować struktury i/lub treści bazy danych żadnymi innymi środkami, oprócz tych, które są przeznaczone do takich celów**, ponieważ takie modyfikacje prowadzą do uszkodzenia bazy danych i doprowadzą do objawów takich jak niezamierzone usunięcie lub modyfikacja danych lub łączy, tworzenie „martwych” łączy lub rekordów, wyświetlanie komunikatów o poważnych błędach, itp.

Zmiany w bazie danych wynikające z tego typu manipulacji unieważniają gwarancję i prawo do korzystania z pomocy technicznej świadczonej przez Peregrine Systems.

Środowiska obsługiwane przez AssetCenter i InfraCenter for Workgroups

Lista środowisk obsługiwanych przez AssetCenter i InfraCenter for Workgroups znajduje się w podręczniku pt. „Podręcznik instalacji i aktualizacji”. Użytkownik może korzystać z aplikacji AssetCenter lub InfraCenter for Workgroups w środowiskach innych niż te, dla których są one przeznaczone wyłącznie na własne ryzyko. Zmiany w bazie danych wynikające z używania AssetCenter lub InfraCenter for Workgroups w środowiskach innych niż te, dla których są one przeznaczone unieważniają gwarancję i prawo do korzystania z pomocy technicznej świadczonej przez Peregrine Systems.

Przedmowa

Prezentacja AssetCenter

AssetCenter jest kompletnym systemem zarządzania zasobami informatycznymi o pięciu zintegrowanych obszarach funkcjonalności, działających w oparciu o wspólne repozytorium:

- Zarządzanie zasobami,
- Zarządzanie nabywaniem,
- Zarządzanie leasingiem,
- Zarządzanie kosztami.

Dzięki integracji wszystkie funkcje dostępne są przez zunifikowany interfejs i z wykorzystaniem tej bazy danych, dzięki czemu można uniknąć redundancji danych.

Lista dostępnych funkcji zależy od nabytych uprawnień licencyjnych.

Funkcjonalność pozwala na dostosowanie interfejsu użytkownika do własnych potrzeb. Interfejs można uprościć, wybierając tylko te zakresy funkcjonalności, które będą używane.

Funkcje uaktywniane są indywidualnie dla każdej stacji roboczej przy pomocy pozycji menu **Plik/ Aktywuj moduły**.

Przedmiot opisu funkcjonalności

Niniejsza dokumentacja zapewnia szczegółowe, systematyczne wyjaśnienia dotyczące funkcjonalności AssetCenter.

Jest ona dostępna w dwóch formach: Jako pomoc on-line oraz w formie drukowanej instrukcji o podobnej treści.

Oznaczenia

Niektóre wyrażenia posiadają szczególne definicje: są one wyświetlone na zielono w systemie pomocy on-line.

Dostępne podręczniki

- Wprowadzenie
- Zarządzanie zasobami
- Zarządzanie nabywaniem
- Zarządzanie leasingiem
- Zarządzanie kosztami
- Administracja i zaawansowane zastosowania AssetCenter

Kontakt z firmą Peregrine Systems

Główna siedziba

Peregrine Systems, Inc.
12670 High Bluff Drive
San Diego, CA 92130
USA
Tel: +1 858 481 5000 lub 800 638 5231
Faks: +1 858 481 1751
Internet: <http://www.peregrine.com>

Pomoc techniczna:
Tel: +1 858 794 7402 lub 800 960 9998
Faks: +1 858 794 6028
Internet: <http://support.peregrine.com>
E-mail: support@peregrine.com
Czynne od poniedziałku do piątku od 5:00 do 17:30 (PST)

Francja, Hiszpania, Grecja i Afryka (z wyjątkiem RPA)

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

Pomoc techniczna:
Tel: +33 (0) 800 505 100
Faks: +33 (0)1 47 73 11 61
E-mail: frsupport@peregrine.fr
Czynne od poniedziałku do piątku od 8:00 do 18:00 (czasu lokalnego)

Niemcy i Europa Wschodnia

Peregrine Systems GmbH
Bürohaus ATRICOM
Lyoner Strasse 15
60528 Frankfurt
Niemcy
Tel: +49 (0)(69) 6 77 34-0
Faks: +49 (0)(69) 66 80 26-26

Pomoc techniczna:
Tel: 0800 2773823
Faks: +49 (0)(69) 66 80 26-26
E-mail: psc@peregrine.de
Czynne od poniedziałku do piątku od 8:00 do 17:00 (czasu lokalnego)

Wielka Brytania

Peregrine Systems, Ltd.
Ambassador House
Paradise Road
Richmond
Surrey TW9 1SQ
UK
Tel: +44 (0)20 8332 9666
Faks: +44 (0)20 8332 9533

Pomoc techniczna:
Tel: +44 (0)20 8334 5844 lub 0800 834 7700
Faks: +44 (0)20 8334 5890
E-mail: uksupport@peregrine.com
Czynne od poniedziałku do piątku od 8:00 do 18:00 (czasu lokalnego)

Dania, Norwegia, Finlandia i Islandia

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

Pomoc techniczna:
Tel: +45 77 31 77 76
Faks: +45 43 46 76 77
E-mail: support.nordic@peregrine.com
Czynne od poniedziałku do piątku od 8:30 do 16:00 (czasu lokalnego)

Holandia, Belgia i Luksemburg

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

Pomoc techniczna:
Tel: 0800 0230889 (Holandia)
lub 0800 74747575 (Belgia i Luksemburg)
Faks: +31 (0) 348 43 7080
E-mail: benelux.support@peregrine.com
Czynne od poniedziałku do piątku od 8:00 do 18:00 (czasu lokalnego)

Singapur

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

Włochy

Peregrine Systems, S.r.l.
Via Monte di Pietà, 21
I-20121 Milano
Włochy

Tel: +39 (02) 6556931
Faks: +39 (02) 65569390

Szwecja

Peregrine Systems AB
Frösundaviks Allé 15, 4th floor
S-169 70 Solna
Szwecja
Tel: +46 (0)8-655 36 04
Faks : +46 (0)8-655 26 10

Pomoc techniczna:
Tel: +45 77 31 77 76
Faks: +45 43 46 76 77
E-mail: nordic@peregrine.com
Czynne od poniedziałku do piątku od 8:30 do 16:30 (czasu lokalnego)

Japonia

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

Tel: +81 (3) 5322-1350
Faks: +81 (3) 5322-1352

Pomoc techniczna:
Tel: +81 (3) 5322 1350
Faks: +81 (3) 5322 1352
E-mail: glipper@peregrine.com

Konwencje

Zastosowano następujący zapis poleceń:

[]	Nawiasy kwadratowe oznaczają opcjonalny parametr. Nie należy ich wpisywać w treści polecenia. Wyjątek: W skryptach języka BASIC używane są na oznaczenie ścieżki dostępu do danych i muszą być wpisane do skryptu: [Link.Link.Field]
< >	Nawiasy ostre oznaczają nazwę parametru w języku naturalnym. Nie należy ich wpisywać. Należy je zastąpić odpowiednim tekstem.
{ }	Nawiasy klamrowe oznaczają szereg parametrów. Można użyć tylko jednego z tych parametrów. W treści polecenia nie należy wpisywać nawiasów klamrowych.
	Znak „ ” używany jest do oddzielenia szeregu parametrów ujętych w nawiasy klamrowe.
*	Gwiazdka po prawej stronie nawiasów kwadratowych oznacza, że daną formułę można powtórzyć kilka razy.

Poniższe style tekstu mają następujące znaczenie:

Czcionka nieproporcjonalna	Polecenie systemu DOS.
Przykład	Przykład kodu lub polecenia
...	Pominięty kod lub polecenie
Nazwa obiektu	Nazwy pól, zakładek menu i plików oznaczone są wyłuszczeniem.
Uwaga	Bardzo ważne informacje.

Prosimy o uwagi

Pragniemy dostarczać jak najdokładniejszą dokumentację.

Cenimy sobie wszelkie uwagi.

Prosimy przesyłać je pod adresem documentation@peregrine.com.

Spis treści

Rozdział 1 - Opcje bazy danych	1
<hr/>	
Rozdział 2 - Struktura bazy danych AssetCenter	7
Przedmowa	7
Definicja bazy danych	9
Struktura plików database.txt i tables.txt	9
Opis tabel	11
Identyfikacja tabel AssetCenter	11
Ciągi znakowe opisu tabeli	11
Opis pól	13
Identyfikacja pól w AssetCenter	13
Typy pól i formaty wprowadzania danych	14
Pola dtLastModif	20
Pola „PełnaNazwa”	20
Opis łączy	21
Identyfikacja łączy	21
Typ łączy	22
Liczba kardynalna łączy	23
Tabele pośrednie	24
Łączy kontekstowe	26
Opis indeksów	26
Identyfikowanie indeksów	26
Unikalność wartości pól indeksowanych	26
Dostosowywanie bazy danych	27
Dostosowywanie tabel	28
Dostosowywanie pól i łączy	28
Dostosowywanie indeksów	29
Wartości domyślne dla pól i łączy	29
Liczniki w domyślnych wartościach pola	30
Błędy w opisie wartości domyślnych	31
Znaczniki HTML rozpoznawane w pomocy dla pól i łączy	32
<hr/>	
Rozdział 3 - Tworzenie, dostosowywanie oraz opisywanie bazy danych AssetCenter	35
Korzystanie z programu AssetCenter Database Administrator	36
Uruchamianie programu AssetCenter Database Administrator	37
Opis menu Plik	37

Opis menu Edycja w programie AssetCenter Database Administrator	40
Opis menu Działanie w programie AssetCenter Database Administrator	41
Tworzenie bazy danych	48
Dostosowywanie bazy danych	48
Naprawianie bazy danych	55
Opis bazy danych AssetCenter	55
Parametry opisu bazy danych	56
Składnia dla szablonów opisu	58
Tekst stały	59
Komentarze	59
Dołączanie innego szablonu	59
Przeglądanie, sortowanie i filtrowanie składników	60
Warunki \$if...\$else...\$elseif...\$endif	62
Funkcje dostępne w szablonach opisu	63
Przetwarzanie wartości właściwości przy użyciu funkcji zdefiniowanej w szablonie	67
Usuwanie znacznika końca akapitu	68
Zliczanie przeglądanych składników	69
Definiowanie zmiennej globalnej dla szablonu	69
Informacje dotyczące niektórych parametrów opisu bazy danych	70
Instancja bazy danych	71
Instancja tabeli	71
Instancja pola	73
Instancja łącza	77
Instancja indeksu	79
Instancja skryptu	79
Zmienne globalne	81
Wartości generowane podczas procesu eksportu	82
<hr/>	
Rozdział 4 - Zarządzanie listami pozycji	83
Listy pozycji dowolnych	83
Systemowe listy pozycji	85
<hr/>	
Rozdział 5 - Jak działa funkcja historii	87
<hr/>	
Rozdział 6 - Zarządzanie dostępem użytkownika do baz danych	
AssetCenter	93
Znaczenie i omówienie uprawnień dostępu do bazy danych	94
W jaki sposób zapewnić bezpieczeństwo i poufność danych	96
Definicje	96
Definicja profilu użytkownika	96
Definicja uprawnień użytkownika	97
Definicja ograniczenia dostępu	97
Definiowanie warunków dostępu	97
Definiowanie profili użytkownika	97
Definiowanie uprawnień użytkownika	98
Definiowanie ograniczeń dostępu	100
Definiowanie użytkowników AssetCenter	102
Zarządzanie połączeniami użytkownika	104
Typy dostępu do bazy danych	104
Jak działają kanały połączeń	106

Usuwanie kanałów połączeń	107
Zarządzanie hasłami	108
Hasło dla loginu Admin	109
Modyfikowanie hasła dla loginu „Admin”	109
Hasło użytkownika	109
Utracone hasła	110
Rozdział 7 - Pisanie zapytań w AQL	111
Przedstawienie	111
Zapytania w AssetCenter	111
AQL	112
Zalecenia dotyczące pisania zapytań AQL	115
Przedstawienie połączeń AQL	115
Przyczyna i użyteczność rekordów klucza podstawowego 0	116
Użycie wartości NULL	119
Wyrażenie Self	119
Funkcja CurrentUser	120
Systemowe listy pozycji	121
Tabele hierarchiczne	122
Uproszczona notacja AQL	123
Sortowania i indeksy	125
Edytor zapytań	128
Omówienie	128
Uzyskiwanie dostępu do edytora zapytań	129
Tworzenie zapytania przy użyciu edytora zapytań	130
Pola używane w zapytaniach	132
Pisanie wyrażień	133
Stałe	134
Składnia AQL	135
Konwencje	135
Składnia zapytań	136
Warunek FROM – Tabele włączone do zapytania	137
Elementy zapytania	139
Warunek WHERE	144
Warunek GROUP BY	145
Warunek HAVING	146
Warunek ORDER BY	147
Odwołanie funkcji AQL	147
Funkcje AQL typu „Ogółem”	148
Funkcje AQL typu „Ciąg znaków”	148
Funkcje AQL typu „Data”	149
Funkcje AQL typu „Numeryczne”	151
Funkcje AQL typu „Test”	152
Przykłady zapytań	152
Rozdział 8 - Edytowanie formularzy	159
Definicja formularza	159
Tworzenie formularzy	160
Edytowanie formularzy i obiektów	160
Właściwości obiektów w formularzach	163
Siatka wzorca	165
Ustawienia strony formularza	166

W jaki sposób łatwo tworzyć regularne raporty	167
<hr/>	
Rozdział 9 - Edytowanie raportów	169
Działanie i instalacja generatora raportów	169
Szczegóły raportu	170
W jaki sposób zmodyfikować raport Seagate Crystal	171
Statystyka Crystal Reports	172
Tworzenie raportu szczegółów	172
<hr/>	
Rozdział 10 - Dostęp tylko do odczytu do bazy danych przy użyciu sterownika ODBC systemu AssetCenter	175
Omówienie dostępu do bazy danych AssetCenter	176
Przykład: Tworzenie raportu w Crystal Reports przy użyciu sterownika ODBC	179
<hr/>	
Rozdział 11 - Definiowanie działań	181
Definicja działania	182
Tworzenie działania	182
Typy działań	182
Metoda ogólna	185
Wypełnianie zakładki DDE	187
Wypełnianie zakładki <i>Poczta</i>	188
Przykłady działań	191
Przykład działania typu „Wykonywalne”	192
Przykład działania typu „DDE”	192
Przykład działania typu <i>Poczta</i>	197
Przykład działania typu Skrypt	197
Używanie zmiennych	199
Specyfika motoru bazy danych Sybase SQL Anywhere	200
Testowanie działania	200
Wykonywanie działania	201
<hr/>	
Rozdział 12 - Zarządzanie terminami ostatecznymi przy użyciu Serwer Asset Center	203
Omówienie programu Serwer Asset Center	204
Uruchamianie programu AssetCenter Server	205
Główny ekran programu Serwer Asset Center	208
Opcje ogólne programu Serwer Asset Center	209
Moduły monitorowane przez Serwer Asset Center	212
Wybór modułów do monitorowania	212
Harmonogramy weryfikacji	213
Linie historii monitorowane przez Serwer Asset Center	215
Linie zapasu monitorowane przez Serwer Asset Center	216
Alarmy monitorowane przez Serwer Asset Center	217
Splata wyliczana przez Serwer Asset Center	219
Wartości straty aktualizowane przez Serwer Asset Center	221
Operacje podziału pozycji kosztowych wykonywane przez Serwer Asset Center	222
Weryfikowanie strefy czasowej serwera bazy danych	224
Sygnalizacja obecności do serwera bazy danych	224
Wyszukiwanie nowych grup wykonawczych workflow	224

Wykonywanie zasad workflow	225
Natychmiastowe uaktywnianie weryfikacji	227
Administrowanie programem Serwer Asset Center poprzez sieć WWW	227
Uruchamianie programu Serwer Asset Center jako usługi NT	227
Uzyskiwanie dostępu do usługi Serwer Asset Center za pośrednictwem sieci WWW	229
Sterowanie usługą Serwer Asset Center za pośrednictwem sieci WWW	230
Rozdział 13 - Zarządzanie wiadomościami	233
Omówienie poczty	234
Warunki wstępne dla wysyłania wiadomości za pośrednictwem zewnętrznego systemu poczty	236
Konfigurowanie AssetCenter do używania systemów poczty	236
Wypełnianie szczegółów pracownika w celu używania zewnętrznego systemu poczty	237
Tworzenie wiadomości	240
Typowe problemy związane z połączeniem	240
Rozdział 14 - Workflow	243
Definicje	244
Workflow	244
Czynność Workflow	245
Zdarzenie Workflow	246
Przejęcie workflow	246
Zadanie Workflow	246
Osoba przypisana do czynności workflow	246
Definicja grupy wykonawczej workflow	247
Ogólny opis	247
Główne tabele związane z workflow	249
Używanie graficznego edytora workflow	250
Czynności	251
Zdarzenia	252
Przejścia	253
Dodatkowa funkcjonalność	254
Jak realizować workflow	254
Przykład zastosowania workflow w akceptacji zapotrzebowania	255
Cel	255
Czynności wstępne	256
Tworzenie czynności	262
Tworzenie zdarzenia początkowego	267
Tworzenie przejść	268
Przykład aktywowania instancji workflow	269
Kontekst instancji workflow	274
Definiowanie kontekstu instancji workflow	274
Obiekt odwołania instancji workflow	274
Ograniczanie instancji workflow w toku dla danego obiektu	275
Role workflow	276
Czynności	278
Czynności typu Pytanie	279
Czynność typu Działanie użytkownika	280
Czynność typu Działanie automatyczne	281
Czynności typu Test/ Skrypt	281
Czynność Początek	283
Szablony czynności	283

Inicjowanie czynności	283
Zadania	284
Tworzenie zadań	284
Czynność typu działanie automatycznie lub test/ skrypt	285
Wyświetlanie listy zadań w toku	285
Wykonywanie zadania użytkownika	286
Przypisywanie zadań użytkownika	286
Administrowanie zadaniem workflow	286
Zdarzenia	287
Zdarzenia systemowe	287
Zdarzenia alarmowe	288
Zdarzenia użytkownika	288
Ogólne warunki aktywacji	290
Przetwarzanie zdarzenia	292
Zastosowanie: Realizacja synchronicznego schematu workflow	295
Zdarzenie końcowe	297
Przejścia workflow	298
Alarmy workflow i limity czasu	298
Limit czasu	298
Alarmy workflow	299
Grupy wykonawcze workflow	300
Śledzenie workflow	301
<hr/>	
Rozdział 15 - Importowanie danych	303
Omówienie importowania danych	304
Zalecenia	306
Ograniczenia, których należy przestrzegać przy importowaniu danych do pól	307
Importowanie plików tekstowych lub bazy danych ODBC	310
Przed importem plików tekstowych	311
Krok 1: Wybór plików tekstowych lub bazy danych ODBC do zaimportowania	312
Krok 2: Definiowanie analizy składniowej plików lub tabel do zaimportowania	313
Krok 3: Opis importowanych pól	317
Krok 4: Mapowanie pól źródłowych do pól docelowych w bazie danych AssetCenter	320
Krok 4: Mapowanie każdego pliku tekstowego lub tabeli źródłowej do tabeli docelowej	321
Krok 4: Mapowanie importowanych pól do pól w bazie danych AssetCenter	321
Krok 4: Dodawanie dodatkowych pól wyliczanych w pliku źródłowym	327
Krok 4: Przypadki szczególne	331
Krok 4: Przykłady używania kluczy	331
Krok 5: Konfiguracja transferu	337
Krok 6: Transfer danych	338
Import bazy danych AssetCenter	339
Zapisywanie i wykonywanie skryptu importu	340
Uruchamianie skryptu importu z wiersza poleceń systemu 32-bitowego	342
<hr/>	
Rozdział 16 - Eksport danych i zarządzanie widokami SQL	345
Definicje	345
Eksport danych z bazy danych AssetCenter	347
Zarządzanie widokami SQL w bazie danych AssetCenter	348
Zalecenia	349
Definiowanie skryptu eksportu	349
Metodologia	350
Definiowanie zapytań eksportowych	351

Format wyjściowy skryptu eksportu	353
Działania dotyczące widoków SQL	355
Wykonywanie skryptu eksportu	356
Wykonywanie skryptu eksportu z modułu AssetCenter Export	356
Wykonywanie skryptów eksportowych w systemie DOS	357
Rozdział 17 - Używanie skryptów	359
Definicja skryptu	360
Zastosowanie skryptów	361
Wprowadzenie do funkcji	362
Definicja funkcji	362
Funkcje wbudowane i programowalne	363
Typy funkcji i parametrów	365
Klasyfikacja funkcji języka Basic	367
Pierwszy krok pisania skryptów	367
Przykładowy scenariusz	367
Krok 1: Utwórz cechę „Podręcznik”	368
Krok 2: Otwórz okno edycji	369
Krok 3: Analiza i definicja algorytmu	369
Krok 4: Napisz skrypt w języku Basic	369
Krok 5: Test skryptu BASIC	370
Wskazówki i ostrzeżenia	370
Uwaga na funkcje programowalne	371
Format stałych typu „Data+Czas” w skryptach	371
Format stałych typu „Czas trwania” w skryptach	372
Dostęp do listy pozycji systemu w trybie odczytu i zapisu	372
Łącze wirtualne „CurrentUser”	374
Komentarze w skrypcie Basic	374
Inicjowanie komunikatu o błędzie	375
Pierwszy przykład	376
Opis problemu	376
Krok 1: Analiza i definicja algorytmu	377
Krok 2: Utworzenie skryptu BASIC	378
Krok 3: Test skryptu BASIC	378
Drugi przykład	379
Opis problemu	379
Krok 1: Analiza i definicja algorytmu	379
Krok 2: Utworzenie skryptu BASIC	380
Krok 3: Test skryptu BASIC	380
Rozdział 18 - Zarządzanie kalendarzami	381
Omówienie kalendarzy	381
Wpływ kalendarzy na niektóre zakresy funkcjonalności	382
Metodologia tworzenia kalendarza	382
Opis tworzenia kalendarza	383
Wprowadzenie informacji ogólnych	383
Wypełnianie zakładki <i>Harmonogram</i>	384
Wypełnianie zakładki <i>Wyjątki</i>	385
Sprawdzanie kalendarza	388

Rozdział 19 - Zarządzanie strefami czasowymi	391
Dlaczego zarządzać strefami czasowymi?	392
Wprowadzanie stref czasowych	393
Tworzenie stref czasowych	393
Zarządzanie strefą czasową	394
Format pola 'Czas letni'	394
Wartości argumentu <Year>	395
Wartości argumentu <DaylightInfo>	395
Przykład	398
Zarządzanie strefami czasowymi w Serwer Asset Center	401
Skutki dla różnych operacji	402
Tworzenie bazy danych	403
Połączenie z bazą danych	404
Import i eksport	405
Rozdział 20 - Używanie AssetCenter jako serwera DDE	407
Definicja serwera DDE	408
Mechanizmy wywołań DDE	408
Usługa DDE	408
Temat DDE	408
Polecenie DDE	409
Wprowadzenie do poleceń DDE	409
Polecenia globalne	410
Connect(<Cnx>, <User>, <Password>)	411
Disconnect()	412
ExecuteAction(<ActionName>)	412
ListAllTables([Mask])	413
ListOpenTables([Mask])	414
OpenView(<ViewName>)	415
Polecenia związane z tabelą	416
OpenTable(<Table>)	416
CloseTable(<Table>)	417
<Table>.GetRecordCount()	417
<Table>.SetViewMode(<Mode>)	418
<Table>.SetRecordMode(<Mode>)	419
<Table>.ListAllFields([Mask])	420
<Table>.ListAllLinks([Mask])	421
<Table>.SetFilter(<Condition>)	422
<Table>.SetSelection(<Condition>)	423
<Table>.GetSelectionId()	424
Polecenia związane z tabelą i polem lub łączem	424
<Table>:<Objet>.AddLink()	425
<Table>:<Object>.GetValue()	425
<Table>:<Object>.Hilight()	426
<Table>:<Object>.RemoveLink()	427
<Table>:<Object>.SetFocus()	428
<Table>:<Object>.SetValue(<Value>)	429
<Table>:<Link>.SetValueWhere(<Condition>)	430
<Table>:<Object>.Show()	431
Wprowadzenie do przykładów wywołań DDE	431
Scenariusz 1: Wewnętrzne wywołania DDE programu AssetCenter	432

Odbiór wywołania	432
Wprowadzanie danych	433
Tworzenie cechy	434
Scenariusz 2: Wywołania DDE z aplikacji Excel 97	434
Opis makra	435
Kod źródłowy makra	435
Scenariusz 3: Tworzenie wywołań DDE za pomocą aplikacji Visual Basic 5.0	436
Przed rozpoczęciem	436
Wykonywanie programu	437
Kod źródłowy programu	438
<hr/>	
Rozdział 21 - Pola wyliczane	439
Definicja pola wyliczanego	439
Użyteczność pól wyliczanych	440
Tworzenie pola wyliczanego	441
Wprowadzenie	441
Metodologia tworzenia pól wyliczanych	443
Używanie pól wyliczanych	446
<hr/>	
Rozdział 22 - Wprowadzenie do kreatorów	449
Definicja kreatora	449
Dla kogo przeznaczone są kreatory?	450
Klasyfikacja kreatorów	450
Kreatory wymieniające informacje z bazą danych AssetCenter	450
Kreatory niezależne	452
<hr/>	
Rozdział 23 - Tworzenie kreatorów	453
Zastosowane konwencje	455
Definicje	455
Model struktury	458
Model strony kreatora	460
Uwagi ogólne	461
Ogólna struktura i składnia	461
Właściwości węzła	462
Sekwencjonowanie kreatorów	468
Funkcje języka Basic	469
Definicja węzła 'Root'	469
Składnia węzła 'Root'	470
Właściwości węzła 'Root'	470
Węzły podrzędne węzła 'Root'	474
Definicja węzła 'Page'	474
Składnia węzła 'Page'	475
Właściwości węzła 'Page'	475
Węzły podrzędne węzła 'Page'	478
Definicja węzła 'Transition'	478
Składnia węzła 'Transition'	479
Właściwości węzła 'Transition'	479
Specyfika węzła 'Transition'	481
Definicja węzła 'Finish'	481

Definicja węzła 'Start'	482
Definicja węzłów typu Long i String	482
Definicja węzła 'Control'	483
Ogólna składnia węzła 'Control'	483
Typy formantów i ich właściwości	483
Wspólne właściwości formantów	484
Formant CheckBox	486
Formant ComboBox	486
Formant ListBox	487
Formant Label	490
Formant OptionButtons	490
Formant ProgressBar	491
Formant CommandButton	492
Formant DBListBox	493
Formant DBQueryBox	495
Formant DBEdit	497
Formant DBTable	498
Formant DBPath	498
Formant LinkEdit	499
Formant TextBox	499
Przykład tworzenia kreatora	500
Krok 1: Analiza potrzeb	500
Krok 2: Określenie organizacji kreatora	501
Krok 3: Zapis struktury kreatora za pomocą języka skryptowego	503
Korzystanie z edytora graficznego	506
Omówienie interfejsu	506
Tworzenie nowego węzła	508
Edycja właściwości węzła	509
Kompilacja, uruchamianie i usuwanie błędów z kreatora	510
Często zadawane pytania	510

Rozdział 24 - Dostrajanie AssetCenter do stosowania w środowisku WAN **515**

Rozdział 25 - Pliki .ini	519
Lista dostępnych plików .ini	519
Edycja plików .ini	521

Rozdział 14 - Workflow

Niniejszy rozdział opisuje szczegółowo, w jaki sposób definiować i zarządzać schematami workflow:

- Definicje
- Ogólny opis
- Główne tabele związane z workflow
- Używanie graficznego edytora workflow
- Jak realizować workflow
- Przykład zastosowania workflow w akceptacji zapotrzebowania
- Kontekst instancji workflow
- Role workflow
- Czynności
- Zadania
- Zdarzenia
- Przejścia workflow
- Alarmy workflow i limity czasu
- Grupy wykonawcze workflow
- Śledzenie workflow

Definicje

Sekcja ta definiuje kilka kluczowych terminów używanych w procesach workflow:

- Workflow
- Czynność Workflow
- Zdarzenie Workflow
- Przejście workflow
- Zadanie Workflow
- Osoba przypisana do czynności workflow
- Definicja grupy wykonawczej workflow

Workflow

Workflow obejmuje formalizację i/lub automatyzację procedur w przedsiębiorstwie.

Za pomocą metod workflow można na przykład opracować modele i zautomatyzować następujące procesy:

- Procedury akceptacji zapotrzebowań zakupu.
- Ruch zasobów.

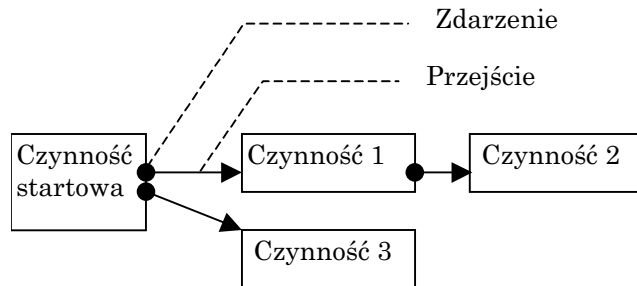
AssetCenter pozwala definiować schematy workflow i zarządzać ich przebiegiem.

Schemat workflow

Tworzenie schematu workflow w AssetCenter polega na zdefiniowaniu:

- Czynności.
- Zdarzeń inicjujących (zdarzeń wynikających z czynności, które pozwalają aktywować przejścia).
- Przejść inicjujących (przejść, które inicjują czynności).
- Kontekstu.

- Limitów czasowych i alarmów.



Uproszczone przedstawienie schematu workflow

Użyj polecenia menu Narzędzia/ Workflow/ Schematy workflow, aby uzyskać dostęp do tabeli schematów workflow (nazwa SQL „amWfScheme”).

Instancja workflow

W niniejszym dokumencie „Instancja workflow” oznacza zdefiniowany schemat workflow, który jest wykonywany.

Czynność Workflow

Workflow składa się z:

- Zadania do wykonania. Zadanie to może wymagać interakcji z użytkownikiem lub może być wykonywane automatycznie przez Serwer Asset Center.
- Zdarzeń, które inicjują przejścia do innych czynności.

Czynności workflow przechowywane są w tabeli czynności workflow (nazwa SQL: amWfActivity).

Użyj edytora graficznego w zakładce Czynności szczegółów workflow, aby uzyskać dostęp do szczegółów czynności workflow.

Zdarzenie Workflow

Zdarzenia workflow wynikają z czynności. Umożliwiają one z kolei aktywowanie przejść, które inicjują inne czynności.

Zdarzenia należące do schematu workflow przechowywane są w tabeli zdarzeń workflow (nazwa SQL: amWfEvent).

Gdy zdarzenia te występują, są one rejestrowane w tabeli zdarzeń elementarnych instancji workflow (nazwa SQL: amWfOccurEvent).

Użyj edytora graficznego w zakładce Czynności szczegółów workflow, aby uzyskać dostęp do szczegółów zdarzenia workflow.

Przejście workflow

Przejście workflow umożliwia przechodzenie z jednej czynności do innej. Przejścia są inicjowane przez zdarzenia.

Zdarzenie może być skojarzone z kilkoma przejściami.

Przejścia należące do schematu workflow są przechowywane w tabeli przejść workflow (nazwa SQL: amWfTransition).

Użyj edytora graficznego w zakładce Czynności szczegółów workflow, aby uzyskać dostęp do szczegółów zdarzenia workflow.

Zadanie Workflow

Zadanie workflow jest to przypisane zadanie do wykonania, wynikające z zainicjowania czynności.

Aby zadania workflow były rejestrowane w tabeli zadań workflow (nazwa SQL: amWfWorkItem), musi być zaznaczona opcja **Rejestruj zadanie** (nazwa SQL:bLogWorkItem) w zakładce **Ogólne**.

Użyj polecenia menu Narzędzia/ Zadania w toku, aby uzyskać dostęp do zadań workflow do wykonania.

Osoba przypisana do czynności workflow

Osoby przypisane wyznaczone są do wykonywania zadań wynikających z czynności workflow typu **Pytanie** lub **Działanie użytkownika**. Osoby przypisane nie biorą udziału w czynnościach typu **Działanie automatyczne** lub **Test / skrypt**.

Informacje o osobach przypisanych do workflow są przechowywane w tabeli ról workflow (nazwa SQL: amWfOrgRole). Użyj polecenia menu Narzędzia/ Workflow/ Role, aby uzyskać dostęp do tabeli ról workflow.

Definicja grupy wykonawczej workflow

Grupy wykonawcze workflow umożliwiają klasyfikację zdefiniowanych schematów workflow. Grupa wykonawcza, do której należy dany schemat workflow podana jest w polu **Grupa wykonawcza** (nazwa SQL: GroupName) w zakładce **Ogólne** szczegółów workflow.

Ogólny opis

Pierwszym etapem realizacji workflow w AssetCenter jest zdefiniowanie schematów workflow za pomocą edytora graficznego dostępnego przez polecenie menu **Narzędzia/ Workflow/ Schematy workflow**. Schematy workflow definiują czynności, zdarzenia i przejścia. Mogą one się odwoływać do działań AssetCenter i pracowników (osoby przypisane do workflow).

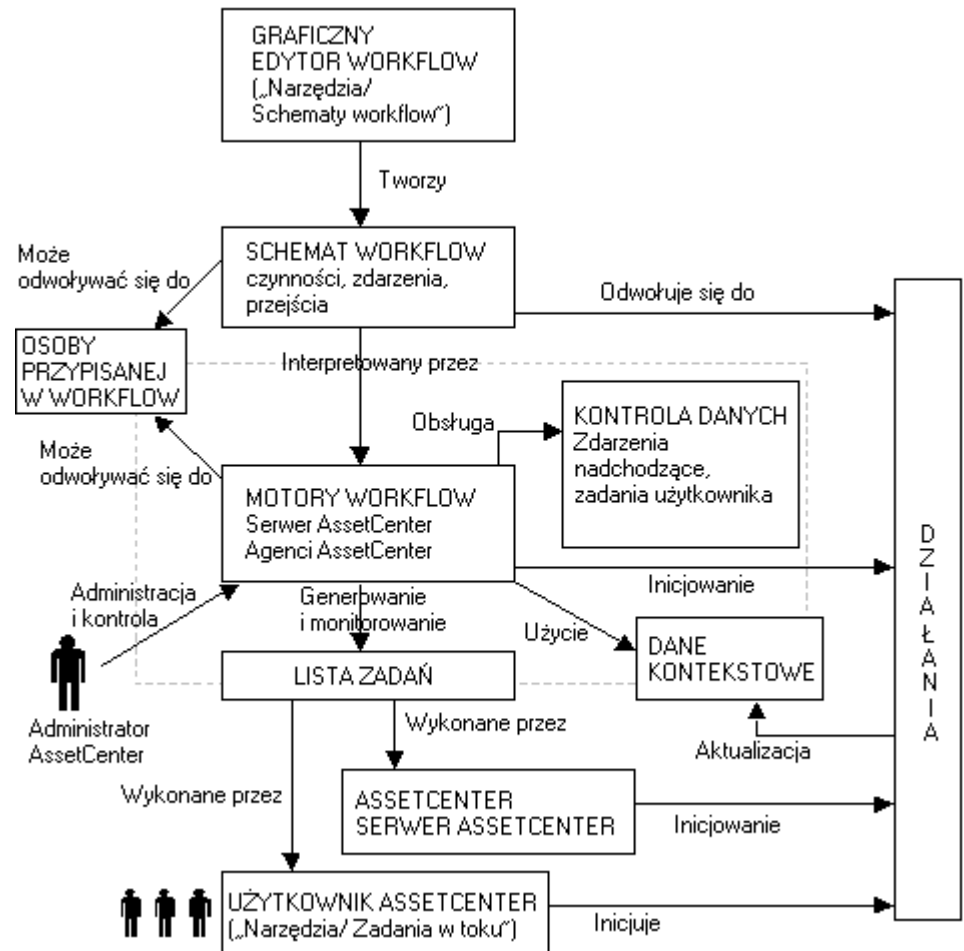
Schematy workflow są interpretowane przez motory workflow. Motory workflow są uruchamiane przez Serwer Asset Center lub agentów AssetCenter.

W reakcji na zdarzenia, motory workflow inicjują i monitorują instance workflow:

- Motory workflow generują zadania do wykonania.
- Monitorują one te zadania i zdarzenia prowadzące do czynności.
- Rejestrują one również przebieg zdarzeń, zapisując w dzienniku nadchodzące zdarzenia i zadania użytkownika do wykonania.

Zadania workflow są wykonywane przez motory workflow lub przez użytkowników AssetCenter. Po ich wykonaniu, aktywowane są zdarzenia, które są następnie uwzględniane przez motory workflow.

Poniższy diagram ilustruje realizację workflow w AssetCenter:

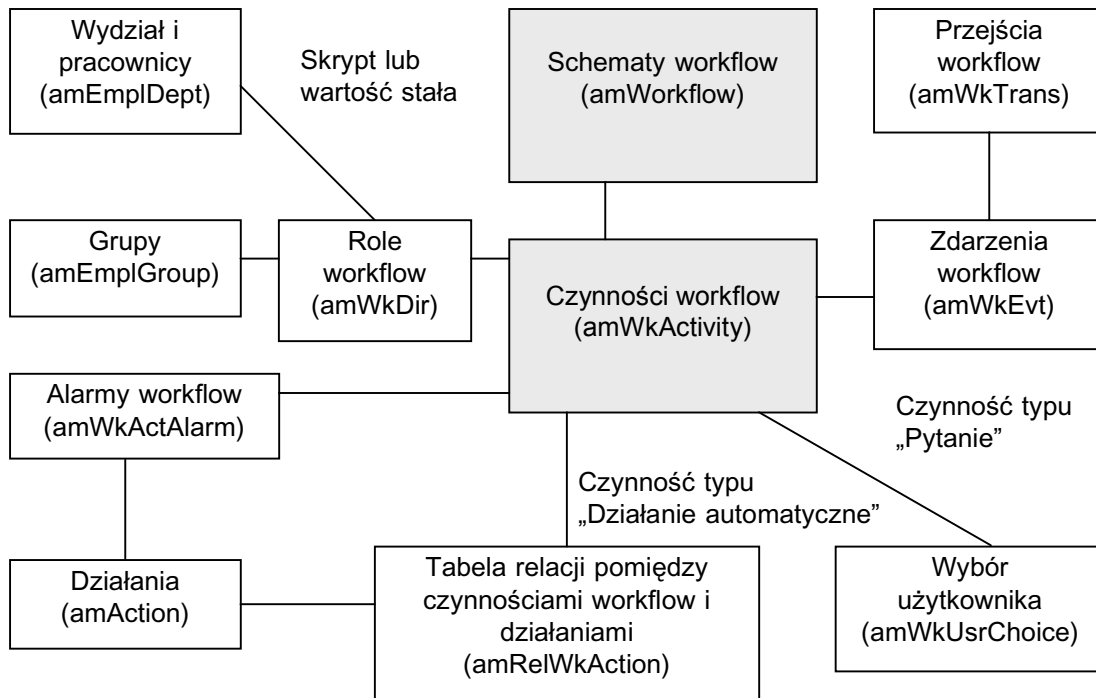


Ogólny opis workflow w AssetCenter

Przetwarzanie instancji workflow różni się w zależności od właściwości czynności i zdarzeń oraz sposobu w jaki skonfigurowano sposób ich kontrolowania przez motory workflow.

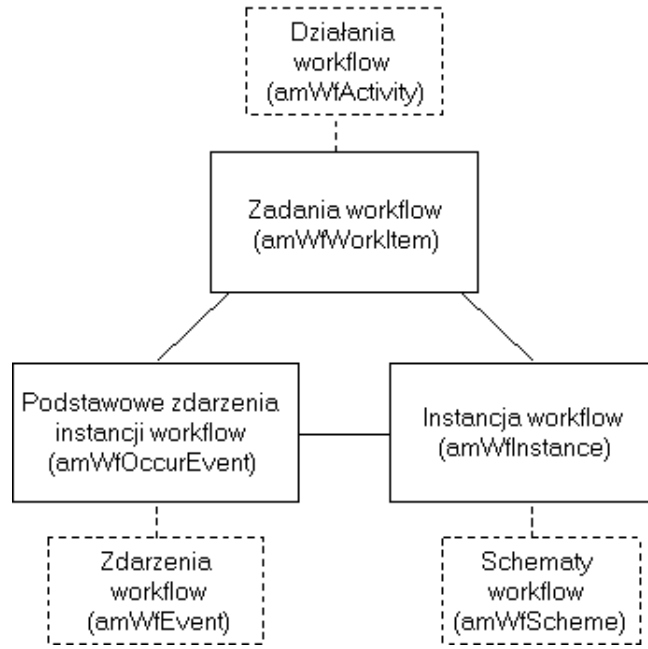
Główne tabele związane z workflow

Na poniższym diagramie przedstawiono główne tabele związane z workflow i łącza pomiędzy nimi. Tabele są oznaczane przez etykietę i nazwę SQL:



Główne tabele umożliwiające zdefiniowanie schematu workflow

Na poniższym diagramie przedstawiono główne tabele związane z instancją workflow będącą w toku:



Główne tabele związane z instancją workflow będącą w toku

Używanie graficznego edytora workflow

Użyj polecenia menu Narzędzia/ Workflow/ Schematy workflow, aby uzyskać dostęp do schematów workflow. To polecenie menu jest zarezerwowane dla administratorów AssetCenter.

Zakładka **Czynności** w szczegółach schematu workflow zawiera edytor graficzny, który umożliwia łatwe dla użytkownika tworzenie schematów workflow.


W tej sekcji opisano, jak używać edytora graficznego, aby tworzyć, modyfikować lub usuwać schematy workflow:

- Czynności
- Zdarzenia


- Przejścia
- Dodatkowa funkcjonalność

Czynności

Aby utworzyć czynność:

- Kliknij prawym przyciskiem myszy zakładkę **Czynności**, a następnie wybierz polecenie **Dodaj czynność** z menu kontekstowego. Zostaną wyświetlone szczegóły czynności.
- Możesz również kliknąć przycisk , a następnie kliknij obszar graficzny. Zostaną wyświetlone szczegóły czynności.

Aby usunąć czynność:

- Wybierz czynność klikając ją, a następnie naciśnij przycisk **Delete**.
- Możesz również wybrać czynność, a następnie wybrać polecenie **Usuń** z menu kontekstowego.
- Możesz również wybrać czynność i kliknąć przycisk .

Aby zmodyfikować szczegóły czynności:


- Kliknij czynność prawym przyciskiem myszy i wybierz polecenie **Szczegóły czynności** z menu kontekstowego, lub
- Dwukrotnie kliknij czynność.




Zdarzenia

Zdarzenie typu Baza danych lub Okresowo

Aby dodać zdarzenie wynikowe do czynności:

- Kliknij prawym przyciskiem myszy czynność, a następnie wybierz polecenie **Dodaj zdarzenie** z menu kontekstowego.
- Możesz również wybrać czynność, a następnie kliknąć przycisk .

Aby usunąć zdarzenie wynikowe typu **Baza danych** lub **Okresowo** z czynności:

- Wybierz zdarzenie, a następnie naciśnij przycisk „Delete” lub
- Dwukrotnie kliknij zdarzenie i wybierz polecenie **Usuń** z menu kontekstowego lub
- Wybierz zdarzenie i kliknij przycisk .

Aby zmodyfikować szczegóły zdarzenia typu **Baza danych** lub **Okresowo**:

- Dwukrotnie kliknij zdarzenie lub
- Wybierz zdarzenie, a następnie wybierz polecenie **Szczegóły zdarzenia** z menu kontekstowego.
- Aby zmodyfikować szczegóły „Zdarzenia systemowego”, możesz również poddać edycji szczegóły czynności będącej u źródła zdarzenia.

Zdarzenie System

Można tworzyć i usuwać zdarzenia **System** na podstawie ich czynności źródłowych.

Aby zmodyfikować zdarzenie **System** użyj odpowiednio jednej z następujących metod:


- Aby zmodyfikować metodę przetwarzania zdarzenia (pole **Przetwarz.** (nazwa SQL: seProcessingMode) w szczegółach zdarzenia), postępuj tak jak przy modyfikacji zdarzenia typu **Baza danych** lub **Okresowo**.
- Możesz również poddać edycji i zmodyfikować szczegóły czynności źródłowej zdarzenia.

Przejścia

Aby utworzyć przejście:

- Kliknij zdarzenie początkowe, aby wybrać przejście.
- Przytrzymaj przycisk myszy wciśnięty i przeciągnij je do czynności docelowej.

Aby usunąć przejście:

- Kliknij przejście, aby je wybrać, a następnie naciśnij przycisk „Delete” lub
- Wybierz przejście, a następnie wybierz polecenie **Usuń** z menu kontekstowego, albo
- Wybierz przejście i kliknij przycisk .

Aby zmodyfikować szczegóły przejścia:

- Kliknij przejście, aby je wybrać.
- Wybierz polecenie **Szczegóły przejścia** z menu kontekstowego.


Aby zmodyfikować źródło i/ lub cel przejścia:

- Wybierz przejście.
- Przeciągnij koniec, który chcesz zmodyfikować.

Dodatkowa funkcjonalność

Edytor graficzny pozwala również:

- Przeciągać i upuszczać czynności połączone i przejścia.
- Powiększać lub zmniejszać schemat za pomocą suwaka

Powiększenie lub przycisku .

Jak realizować workflow

Workflow w AssetCenter pozwala zautomatyzować procedury w przedsiębiorstwie. Oto etapy tego procesu:

- ↳ Przeanalizuj procedury przedsiębiorstwa, które mają ulec formalizacji.
- ↳ Utwórz:
 - ❖ Role workflow.
 - ❖ Działania.
- ↳ Utwórz schematy workflow. Zdefiniuj:
 - ❖ Czynności, zdarzenia i przejścia.
 - ❖ Alarmy.
- ↳ Zdefiniuj odpowiednie grupy wykonawcze workflow. Skojarz każdy schemat workflow z grupą wykonawczą workflow.
- ↳ Uruchom Serwer Asset Center na jednej lub kilku stacjach roboczych. Dla każdej instancji Serwer Asset Center, zdefiniuj grupy wykonawcze workflow, które będą monitorowane oraz parametry monitorowania.

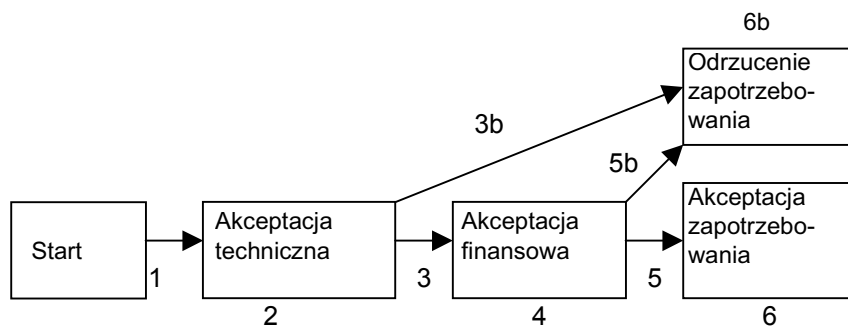
Przykład zastosowania workflow w akceptacji zapotrzebowania

W tej sekcji podano prosty przykład workflow.

- Cel
- Czynności wstępne
- Tworzenie czynności
- Tworzenie zdarzenia początkowego
- Tworzenie przejść
- Przykład aktywowania instancji workflow

Cel

Celem tego schematu workflow jest automatyzacja procesu zapotrzebowania zakupu według poniższych założeń:



Schemat Workflow

Etapy schematu workflow są następujące:

- ↳ Instancja workflow zaczyna działać w momencie wystąpienia potrzeby akceptacji zasadności zapotrzebowania zakupu, to znaczy gdy pole **Stan zatwierdzenia** (nazwa SQL: seApprStatus) w zakładce **Zatwierdzenie** zapotrzebowania zakupu jest ustawione jako **Zatwierdzić**.

- ↪ Najpierw zapotrzebowanie podlega akceptacji technicznej. Etap ten polega na przedłożeniu zapotrzebowania kierownikowi wydziału do akceptacji. Kierownik informowany jest za pomocą wiadomości. Programowany jest alarm przypominający, jeśli zapotrzebowanie nie zostanie rozpatrzone przez osobę zatwierdzającą do końca następnego dnia roboczego po wysłaniu wiadomości dotyczącej akceptacji zapotrzebowania.
- ↪ Jeśli osoba odpowiedzialna zaakceptuje zapotrzebowanie pod względem technicznym, następnym etapem jest akceptacja finansowa.
W przeciwnym wypadku zapotrzebowanie jest odrzucane.
- ↪ Akceptacja finansowa polega na przedstawieniu zapotrzebowania kontrolerowi finansowemu firmy, Gerardowi Borewiczowi. Jest on również informowany za pomocą poczty i programowany jest alarm przypominający.
- ↪ Jeśli kontroler finansowy zaakceptuje zapotrzebowanie zakupu, jest ono zatwierdzone.
W przeciwnym wypadku zapotrzebowanie jest odrzucane.
- ↪ Jeśli zapotrzebowanie zakupu zostało zatwierdzone, AssetCenter ustawia pole **Stan zatwierdzenia** (nazwa SQL: seApprStatus) w zakładce **Zatwierdzenie** szczegółów zapotrzebowania zakupu jako **Zatwierdzone**.
Jeśli zapotrzebowanie zakupu zostało odrzucone, AssetCenter ustawia pole **Stan zatwierdzenia** (nazwa SQL: seApprStatus) w zakładce **Zatwierdzenie** szczegółów zapotrzebowania zakupu jako **Odmowa**.
W obu przypadkach, AssetCenter informuje osobę zgłaszającą zapotrzebowanie.

Czynności wstępne

Zalecamy utworzenie niniejszego schematu workflow w przykładowej bazie danych dołączonej do AssetCenter i nawiązanie połączenia z tą bazą danych za pomocą nazwy logowania „Admin”.

Tworzenie osób przypisanych do workflow

Osoby przypisane do tego schematu workflow to:

- Kierownik wydziału osoby zgłaszającej zapotrzebowanie.
- Kontroler finansowy firmy, Gerard Borewicz.

Informacje o osobach przypisanych do workflow są przechowywane w tabeli ról workflow (nazwa SQL: amWfOrgRole). Aby je utworzyć, wybierz polecenie menu **Narzędzia/ Workflow/ Role**, a następnie kliknij

Nowy

Kierownik wydziału osoby zgłaszającej zapotrzebowanie

Osoba jest wyliczana przez skrypt. Aby to zdefiniować wypełnij następujący ekran zgodnie z podanym przykładem:

The screenshot shows a dialog box titled "Szczegóły roli 'Kierownik wydziału osoby zgł. zapotrzebowanie'". It contains the following fields and controls:

- Oznaczenie:** Kierownik wydziału osoby zgł. zapotrzebowana
- Nr ref.:** REQAPPR-ROLE01
- Kontekst:** Zapotrzebowania (amRequest)
- Typ:** Osoba kalkulowana
- Skrypt:** RetVal=[Requester.Supervisor]
- Buttons: Nowy, Powiel, Zamknij

Kontroler finansowy firmy

Osoba ta nazywa się Gerard Borewicz. Aby zdefiniować go jako osobę przypisaną, wypełnij następujący ekran szczegółów zgodnie z podanym przykładem:

The screenshot shows a dialog box titled "Szczegóły roli 'Kontroler finansowy firmy'". It contains the following fields and controls:

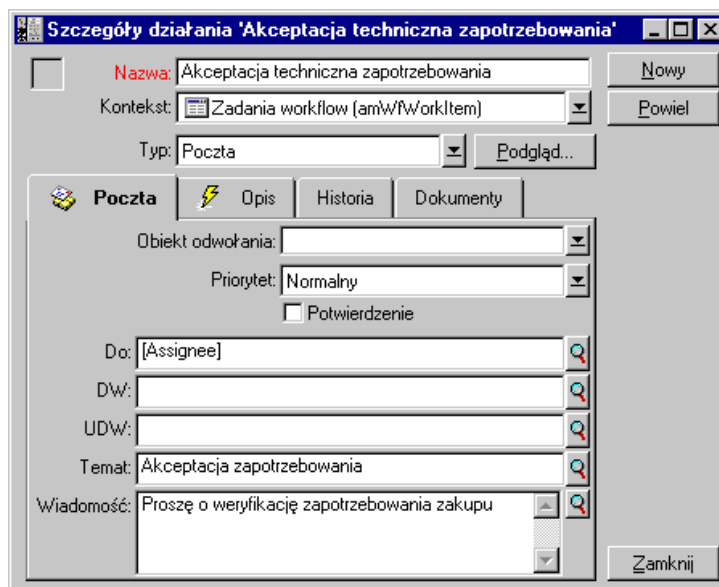
- Oznaczenie:** Kontroler finansowy firmy
- Nr ref.:** D000002
- Kontekst:** Zapotrzebowania (am)
- Typ:** Wyznaczona osoba
- Przypisano:** Borewicz, Gerald
- Buttons: Nowy, Powiel, Zamknij

Tworzenie działań

Schemat workflow odwołuje się do wielu działań. Aby je utworzyć, wybierz polecenie menu **Narzędzia/ Działania/ Edytuj**.

Przesłanie zapotrzebowania do akceptacji technicznej do kierownika wydziału osoby zgłaszającej zapotrzebowanie

To działanie jest używane na etapie akceptacji technicznej. Umożliwia ono powiadomienie osoby odpowiedzialnej za akceptację techniczną o potrzebie sprawdzenia zapotrzebowania:



The screenshot shows a software window titled "Szczegóły działania 'Akceptacja techniczna zapotrzebowania'". It contains several input fields and buttons. At the top, there are buttons for "Nowy" and "Powiel". The "Nazwa" field is filled with "Akceptacja techniczna zapotrzebowania". The "Kontekst" dropdown is set to "Zadania workflow (amiwf/workItem)". The "Typ" dropdown is set to "Poczta". Below these are tabs for "Poczta", "Opis", "Historia", and "Dokumenty". The "Poczta" tab is active, showing fields for "Obiekt odwołania:", "Priorytet: Normalny", and a "Potwierdzenie" checkbox. The "Do:" field contains "[Assignee]". The "Dw:" and "UDw:" fields are empty. The "Temat:" field contains "Akceptacja zapotrzebowania". The "Wiadomość:" field contains "Proszę o weryfikację zapotrzebowania zakupu". A "Zamknij" button is located at the bottom right.

Działanie „Zapotrzebowanie akceptacji technicznej”

Przesłanie zapotrzebowania do akceptacji finansowej do kontrolera finansowego

Działanie „Zapotrzebowanie akceptacji finansowej” jest używane w procesie akceptacji finansowej. Jest podobne do działania „Zapotrzebowanie akceptacji funkcji” opisanego powyżej. Powoduje ono wysłanie wiadomości do osoby odpowiedzialnej za akceptację finansową o potrzebie sprawdzenia zapotrzebowania:

Akceptacja zapotrzebowania zakupu

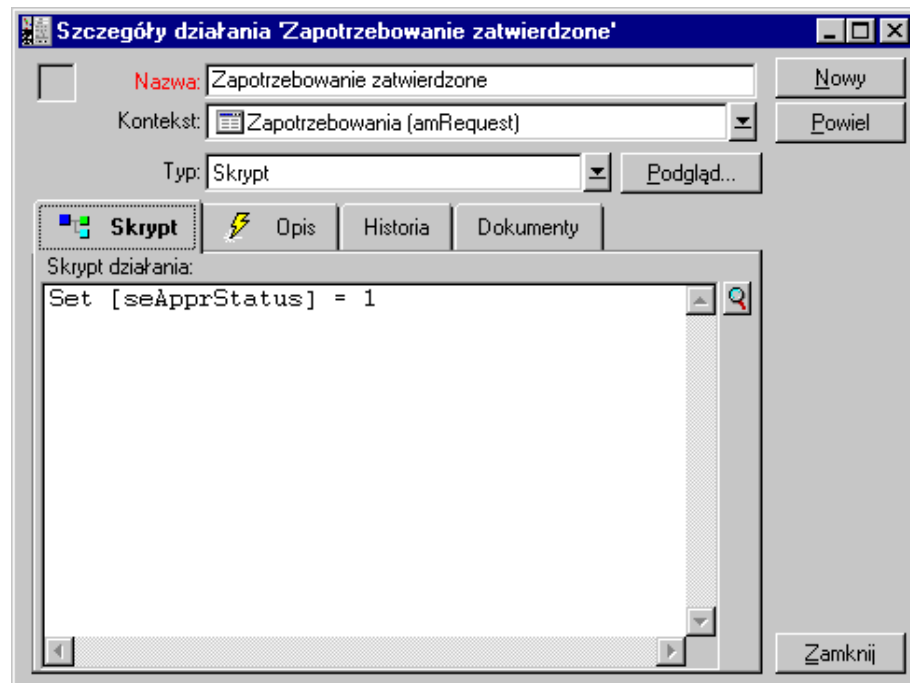
To działanie jest używane na poziomie czynności „Zapotrzebowanie zatwierdzone”.

Ustawia ono pole **Stan zatwierdzenia** (nazwa SQL: seApprStatus) w zakładce **Zatwierdzenie** szczegółów zapotrzebowania zakupu jako **Zatwierdzone**. Jest to działanie typu **Skrypt**.

Pole **Stan zatwierdzenia** jest listą pozycji systemu. Aby zobaczyć listę jego wartości, wyświetl pomoc dla tego pola:

- ❖ Kliknij prawym przyciskiem myszy pole **Stan zatwierdzenia** w szczegółach zapotrzebowania.
- ❖ Wybierz polecenie menu **Pomoc dla tego pola** z menu kontekstowego: Wartość wyświetlana jako **Zatwierdzone** jest przechowywana w bazie danych jako „1”.

Działanie przedstawia się następująco:

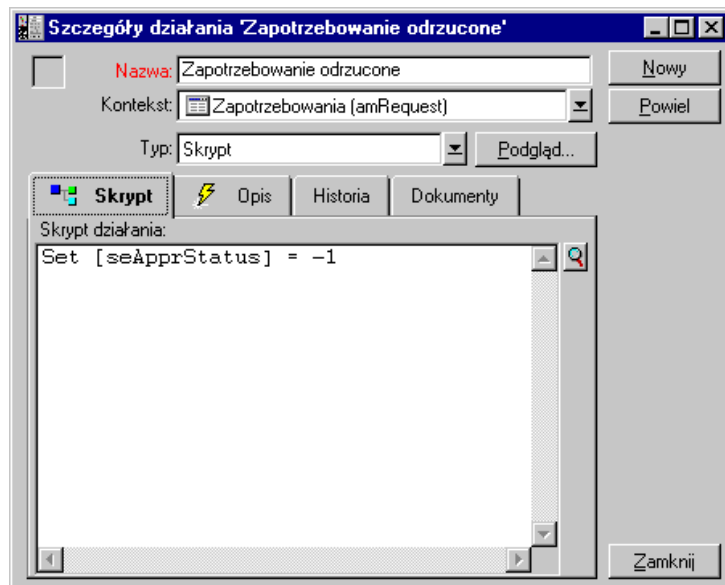


Działanie „Zapotrzebowanie zatwierdzone”

Odrzucenie zapotrzebowania zakupu

Działanie to jest używane na poziomie czynności „Zapotrzebowanie odrzucone”.

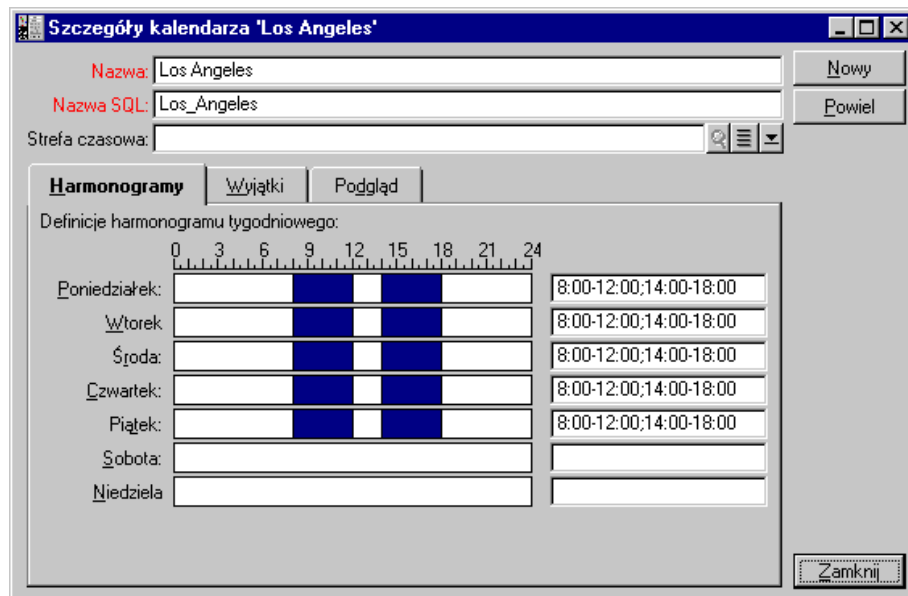
Jest ono podobne do działania „Zapotrzebowanie zatwierdzone”, z tym, że pole **Stan zatwierdzenia** (nazwa SQL: seApprStatus) w zakładce **Zatwierdzenie** szczegółów zapotrzebowania musi być ustawione na wartość **Odmowa**.



Działanie „Zapotrzebowanie odrzucone”

Tworzenie kalendarza

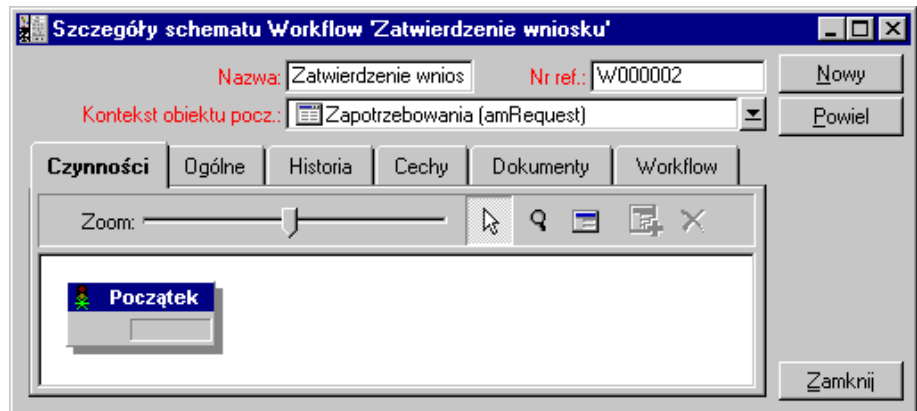
Użyj polecenia menu **Narzędzia/Kalendarze**, aby uzyskać dostęp do listy kalendarzy. Kalendarz jest związany z czynnościami schematu workflow. Umożliwia on określanie terminów końcowych dla instancji schematu workflow:



Kalendarz „Kalendarz LA”

Przygotowanie schematu workflow

- ↪ Uruchom polecenie menu **Narzędzia/Workflow/ Schematy workflow**.
- ↪ Kliknij **Nowy**.
- ↪ Podaj nazwę schematu workflow.
- ↪ Wskaż kontekst obiektu początkowego, który będzie odnosił się domyślnie do wszystkich czynności schematu workflow. W naszym przypadku, jest to tabela zapotrzebowań (nazwa SQL: amRequest).
- ↪ Kliknij **Utwórz**: Czynność początkowa („Start”) jest tworzona automatycznie przez AssetCenter w edytorze graficznym w zakładce **Czynności**.



Schemat workflow „Zatwierdzenie wniosku”

Tworzenie czynności

Czynności mogą być tworzone graficznie w zakładce **Czynności** szczegółów workflow:

- ↪ Kliknij prawym przyciskiem myszy.
- ↪ Wybierz polecenie menu **Dodaj czynność** z menu kontekstowego:
Zostaną wyświetlone szczegóły czynności.

Akceptacja techniczna

- ↪ Podaj nazwę czynności.
- ↪ Ponieważ czynność polega na przekazaniu zapotrzebowania do kierownika wydziału do akceptacji, wybierz wartość **Pytanie** z listy pozycji **Typ** (nazwa SQL: seType).
- ↪ Pole **Kontekst** (nazwa SQL: ContextTable) w zakładce **Ogólne** nie jest modyfikowane.

Zadania do wykonania

Wypełnij zakładkę **Pytanie** w sposób pokazany poniżej:

The screenshot shows a software window titled "Czynność w schemacie Workflow 'Zatwierdzenie wniosku'". It contains several input fields and buttons. At the top, there are fields for "Nazwa" (Akceptacja funkcyjna), "Szablon", "Typ" (Pytanie), and "Nr ref." (W000002-ACTIV02). Below these are tabs for "Ogólne", "Pytanie", "Limit czasu", and "Alarmy". The "Pytanie" tab is active. Under "Osoba odpowiedzialna za pytanie", there is a "Przypisano" field with the value "Kontroler finansowy firmy" and a "Zawiadom osobę" dropdown set to "Tak". The "Pytanie lub instrukcje" field contains the text "Czy zatwierdzasz zapotrzebowanie [ReqNumber]?". The "Możliwe odpowiedzi" section shows one response with "Oznaczenie: Akceptuj" and "Nazwa SQL: ReqApprove". At the bottom, there are buttons for "Akceptuj", "Odmowa", and "Zamknij".

Podaj pytanie, które będzie zadane:

- ❖ Tekst pytania odwołuje się do numeru zapotrzebowania zakupu.
- ❖ Możliwe są dwie odpowiedzi: Odrzucenie lub zatwierdzenie. Aby dodać zakładkę podrzędną opisującą odpowiedź na pytanie, kliknij prawym przyciskiem myszy obszar etykiet zakładek podrzędnych i wybierz polecenie **Dodaj związany rekord** lub **Powiel łącze**.
- ↪ W polu **Przypisano** (nazwa SQL: Assignee) wskaż, do kogo pytanie jest kierowane. W naszym przypadku osoba przypisana to kierownik wydziału osoby składającej zapotrzebowanie. Osoba ta została utworzona w tabeli ról workflow podczas czynności wstępnych.
- ↪ Aby osoba przypisana została poinformowana o potrzebie rozpatrzenia zapotrzebowania:
 - ❖ Ustaw wartość pola **Zawiadom osobę** (nazwa SQL: bNotifAssignee) jako **Tak**.

- ❖ Określ działanie do wykonania: Jest to działanie „Zapotrzebowanie akceptacji technicznej” utworzone podczas czynności pośredniej. Działanie to jest inicjowane automatycznie, gdy zapotrzebowanie zakupu jest przekazywane do akceptacji technicznej.

Limit czasu

W zakładce **Limit czasu** szczegółów czynności:

- Podaj kalendarz dni roboczych dołączony do czynności. Kalendarz ten jest uwzględniany przy obliczaniu dni roboczych. Wybierz kalendarz „Kalendarz LA” utworzony podczas czynności wstępnej.
- Zdefiniuj czas, po którym decyzja musi zostać podjęta, w odniesieniu do czasu, kiedy czynność została zainicjowana. W naszym przypadku, osoba przypisana musi odpowiedzieć na pytanie przed końcem następnego dnia roboczego.

The screenshot shows a software interface for configuring a workflow task. The window title is "Czynność w schemacie Workflow 'Zatwierdzenie wniosku'". At the top, there are input fields for "Nazwa" (Akceptacja funkcyjna), "Szablon" (empty), and "Nr ref." (W000002-ACTIV02). Below these are buttons for "Nowy" and "Powiel". A tabbed interface is visible, with "Limit czasu" selected. Under this tab, there is a "Kalendarz" dropdown set to "Los Angeles" and a "Typ opóźnienia" dropdown set to "Koniec następnego dnia rob.". A "Zamknij" button is located at the bottom right of the window.

Czynność typu limit czasu dla „Akceptacji funkcjonalnej”

Alarm

W zakładce **Alarmy** szczegółów czynności, zdefiniuj alarm przypominający, na wypadek nie podjęcia decyzji przed upływem limitu czasu określonego w zakładce **Limit czasu**.

Dla uproszczenia, alarm zainicjuje działanie „Zapotrzebowanie akceptacji technicznej”:

The screenshot shows a software interface for configuring an alarm. At the top, there are four tabs: 'Pytanie' (with a question mark icon), 'Limit czasu' (with a clock icon), 'Alarmy' (with a bell icon), and 'Historia'. The 'Alarmy' tab is selected. Below the tabs, the configuration is as follows:

- Nazwa:** Nie dokonano jeszcze akceptacji
- Nr ref.:** REQAPPR-FUNCALRM01
- Inicjowanie działania:**
 - Typ:** Czas do zakończenia zadania
 - Limit czasu:** 0g
- Uaktywnienie:** Działanie
- Działanie:** Wyślij przypomnienie akceptacji funkcjon

At the bottom of the configuration area, there is a preview box containing the text: Nie dokonano jeszcze akceptacji

Alarm „Nie rozpatrzono zapotrzebowania”

Można zdefiniować kolejne alarmy za pomocą polecenia **Dodaj związany rekord** z menu kontekstowego.

Zdarzenia

Po utworzeniu czynności, AssetCenter tworzy dwa zdarzenia systemowe „Zatwierdź” i „Odrzuć” zgodnie z dwiema możliwymi odpowiedziami na pytanie.

Gdy zdarzenia te występują, agent AssetCenter rejestruje je w tabeli zdarzeń elementarnych workflow (nazwa SQL „amWfOccurEvent”). Serwer Asset Center dba o zainicjowanie następujących czynności (domyślnie, pole **Przetwarz.** (nazwa SQL: seProcessingMode) w zakładce **Ogólne** jest ustawione jako **Rejestracja zdarzenia i przetwarzanie przez serwer**).

Zapotrzebowanie odrzucone

Czynność „Zapotrzebowanie odrzucone” jest podobna do czynności „Zapotrzebowanie zatwierdzone”.

W tym przypadku, szczegóły zapotrzebowania są modyfikowane tak, aby wskazywać, że zapotrzebowanie zostało odrzucone. Działanie do wykonania to „Zapotrzebowanie odrzucone” utworzone na początku.

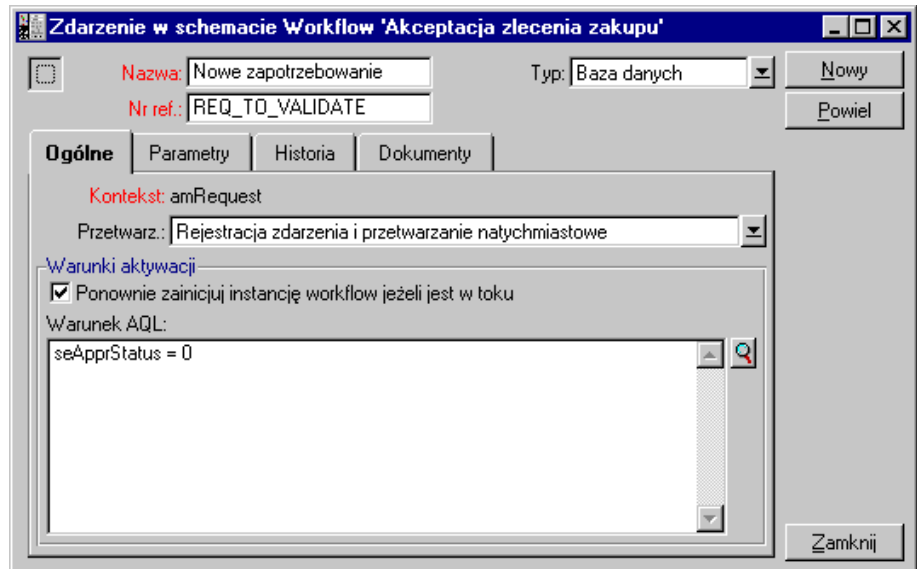
Tworzenie zdarzenia początkowego

Zdarzenia, które inicjują instancję workflow są związane z czynnością „Start”.

Aby zdefiniować zdarzenie „Start”, kliknij prawym przyciskiem myszy pusty obszar zdarzenia w czynności „Start” i wybierz opcję **Szczegóły zdarzenia**.

↳ W naszym przypadku, instancja workflow jest inicjowana, gdy pole **Stan zatwierdzenia** (nazwa SQL: seApprStatus) w zakładce **Zatwierdzenie** szczegółów zapotrzebowania jest ustawione jako **Zatwierdzić**.

Dlatego zdarzenie początkowe jest zdarzeniem typu „Baza danych”, a jego parametry są opisane w zakładce **Ogólne**, tak jak pokazano na poniższym ekranie:



Parametry aktywacji zdarzenia początkowego

- ↳ Ustaw wartość pola **Przetwarz.** (nazwa SQL: seProcessingMode) w zakładce **Ogólne** zdarzenia jako **Rejestracja zdarzenia i przetwarzanie**.
- W zakładce **Parametry** zdarzenia:
 - ❖ Zaznacz pole **Aktualizuj** (nazwa SQL: bUpdate).
 - ❖ Podaj „seApprStatus” jako wartość pola **Monitorowane pola** (nazwa SQL: MonitFields).

Tworzenie przejść

Po utworzeniu czynności, trzeba je połączyć za pomocą przejść.

Aby utworzyć przejście:

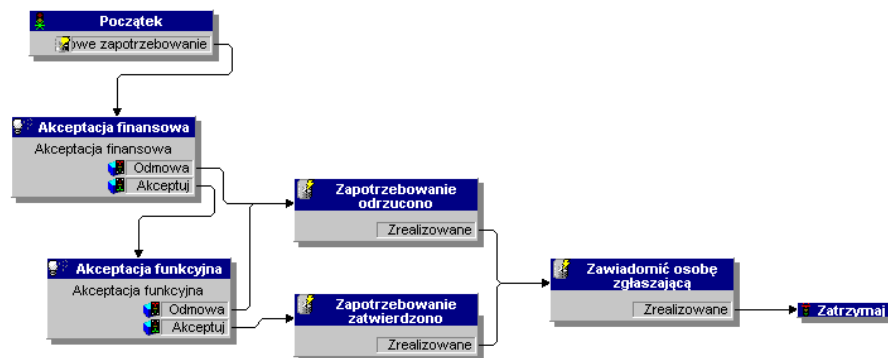
- Kliknij zdarzenie początkowe przejścia.
- Przytrzymując przycisk myszy przeciągnij je do czynności docelowej.

W tym przypadku należy utworzyć następujące przejścia:

- Od zdarzenia początkowego do czynności „Akceptacja techniczna”.

- Od zdarzenia „Zatwierdzić” w czynności „Akceptacja techniczna” do czynności „Akceptacja finansowa”.
- Od zdarzenia „Zatwierdzić” w czynności „Akceptacja finansowa” do czynności „Zapotrzebowanie zatwierdzone”.
- Od zdarzeń „Odrzucić” w czynności „Akceptacja techniczna” i „Akceptacja finansowa” do czynności „Zapotrzebowanie odrzucone”.

Uzyskaliśmy następujący schemat workflow:



Przykład aktywowania instancji workflow

Teraz trzeba sprawdzić, czy schemat workflow działa prawidłowo.

Aby to zrobić, wykonaj następujące czynności:

- ↪ Uruchom i skonfiguruj Serwer Asset Center.
- ↪ Utwórz osoby objęte schematem w tabeli wydziałów i pracowników.
- ↪ Utwórz zapotrzebowanie zakupu do zatwierdzenia.
- ↪ Zainicjuj instancję workflow.
- ↪ Skontroluj instancję.

Uruchamianie Serwer Asset Center

- ↪ Uruchom Serwer Asset Center.
- ↪ Połącz go z bazą danych AssetCenter.

- ↪ Wybierz polecenie **Działanie/ Aktywuj**. Wybierz opcję **Szukaj nowych grup wykonawczych workflow** z listy modułów do aktywowania.
- ↪ Kliknij : Serwer Asset Center sprawdzi, czy utworzono nowe grupy wykonawcze workflow i tworzy specjalne moduły monitorowania dla każdej z tych grup.

Uwaga: Grupy wykonawcze workflow służą do klasyfikowania schematów workflow. Grupa, do której należy schemat workflow jest podana w polu **Grupa wykonawcza** (nazwa SQL: GroupName) w zakładce **Ogólne** szczegółów workflow.

W naszym przykładzie, nie zdefiniowaliśmy grupy wykonawczej dla schematu workflow. Serwer Asset Center tworzy więc moduł monitorowania o nazwie „Wykonywanie reguł workflow dla schematów workflow bez grupy wykonawczej”.

- ↪ Pozostaw działający program Serwer Asset Center.

Tworzenie rekordów w tabeli wydziałów i pracowników

Przed utworzeniem zapotrzebowania zakupu do zatwierdzenia, ważne jest, aby zdefiniować osobę zgłaszającą zapotrzebowanie oraz właściwego kierownika wydziału w tabeli wydziałów i pracowników. Kierownik musi mieć odpowiednie uprawnienia, aby wykonać następujące czynności:

- ↪ Utworzyć zgłaszającego zapotrzebowanie o nazwisku „Jerzy Stolorz”, należącego do wydziału „Informatyka”.
- ↪ Przypisać **Login**, hasło i profil użytkownika umożliwiające wprowadzenie zapotrzebowania zakupu (zakładka **Profil** w szczegółach właściwego rekordu).
- ↪ Kierownik wydziału „Informatyka” nazywa się „Filip Czajka”.
- ↪ Aby uprościć dalsze operacje, przyznaj uprawnienia administratora bazy danych Filipowi Czajce: wyświetl zakładkę **Profil** w szczegółach właściwego rekordu i zaznacz pole **Uprawnienia administratora** (nazwa SQL: bAdminRight). Określ **Login** (nazwa SQL: UserLogin) oraz hasło Filipa Czajki.

Tworzenie zapotrzebowania zakupu do zatwierdzenia

Kolejny etap polega na utworzeniu zapotrzebowania zakupu do zatwierdzenia:

- ↪ Nawiąż połączenie z demonstracyjną bazą danych używając loginu Jerzy Stolorz.
- ↪ Uruchom polecenie menu **Finanse/ Zapotrzebowania** AssetCenter.
- ↪ Kliknij .
- ↪ W polu **Zgłaszający** (nazwa SQL: Requester) w zakładce **Ogólne** szczegółów zapotrzebowania, wybierz rekord „Jan Stolorz”.
- ↪ Ustaw pole **Stan zatwierdzenia** (nazwa SQL: seApprStatus) w zakładce **Zatwierdzenie** szczegółów zapotrzebowania jako **Zatwierdzić**.
- ↪ Potwierdź zapotrzebowanie zakupu: Następuje zdarzenie początkowe, a AssetCenter rejestruje zdarzenie w tabeli zdarzeń elementarnych workflow (nazwa SQL: amWfOccurEvent).

Aktywacja workflow

- ↪ Uruchom polecenie menu **Działanie/ Aktywuj** w Serwer Asset Center. Wybierz z listy moduł **Wykonaj zasady workflow dla schematów bez grupy wykonawczej**.
- ↪ Kliknij OK: Serwer Asset Center wykrywa zdarzenie i aktywuje przejście do czynności „Akceptacja techniczna” związanej z tym zdarzeniem. Robiąc to, serwer tworzy zadanie do wykonania, które jest przypisywane do kierownika wydziału osoby zgłaszającej zapotrzebowanie.

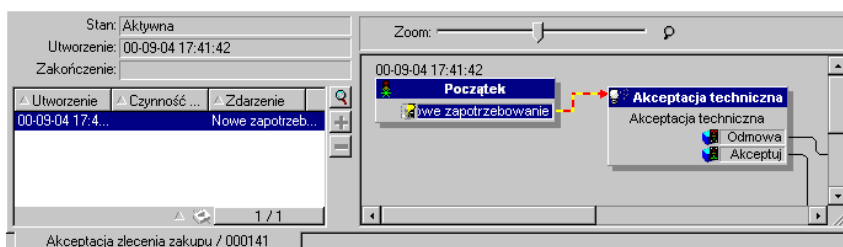
Kontrolowanie instancji

Aby sprawdzić, czy instancja workflow działa prawidłowo, uruchom AssetCenter i nawiąż połączenie z demonstracyjną bazą danych używając loginu Filip Czajka, kierownika wydziału „Jerzego Stolorza”.

Podgląd instancji workflow

Wyświetl szczegóły utworzonego zapotrzebowania zakupu: W zakładce **Workflow** podane są bieżące instancje workflow dla zapotrzebowania. Każda instancja jest opisana w pojedynczej zakładce podrzędnej.

- ❖ W lewej części zakładki podane są zdarzenia, które nastąpiły.
- ❖ W prawej części zakładki podany jest stan instancji. W naszym przypadku, powinno pulsować zadanie „Akceptacja techniczna”.



Bieżąca instancja workflow zapotrzebowania zakupu


Podgląd zadania do wykonania



- ↪ Wybierz polecenie menu **Narzędzia/ Zadania w toku**: Kierownik wydziału „Filip Czajka” może w ten sposób przeglądać szczegóły przypisanego zadania:

Szczegóły zadania wynikające z czynności akceptacji funkcjonalnej

W zakładce **Ogólne** wyświetlane jest pytanie zdefiniowane w zakładce **Pytanie** czynności „Akceptacja techniczna”.

W zakładce **Przypisanie** opisano, kto odpowiada za zadanie i podano właściwy termin. Data ta jest wyliczana na podstawie informacji podanych w zakładce **Limit czasu** w szczegółach czynności „Akceptacja techniczna”, oraz daty utworzenie zadania (tzn. daty aktywacji przejścia przez Serwer Asset Center).

Możesz kliknąć , aby uzyskać dostęp do szczegółów zapotrzebowania tworzącego zadanie.

↪ Kliknij przycisk  lub , aby wykonać zdanie. Można również wprowadzić komentarz dotyczący decyzji w zakładce **Ogólne**.

Kontekst instancji workflow

Każda instancja workflow ma swój specyficzny kontekst.

W tej sekcji omówiono następujące punkty:

- Definiowanie kontekstu instancji workflow
- Obiekt odwołania instancji workflow
- Ograniczanie instancji workflow w toku dla danego obiektu

Definiowanie kontekstu instancji workflow

Definiując schemat workflow, definiuje się:

- Domyślny kontekst podany u góry szczegółów workflow.
- Kontekst dla wszystkich zdarzeń, przejść i czynności workflow (na ekranie szczegółów przejścia lub w zakładce **Ogólne** szczegółów zdarzeń i czynności). Kontekst ten jest połączony z domyślnym kontekstem instancji workflow.

W obu przypadkach, kontekst jest tabelą.

Obiekt odwołania instancji workflow

Jeśli rekord spełnia warunki aktywacji zdefiniowane w schemacie workflow, inicjowana jest instancja workflow. Rekord stanowi obiekt odwołania na poziomie zdarzenia początkowego.

Gdy instancja workflow jest w toku, obiekt odwołania może się zmieniać zgodnie z kontekstem zdefiniowanym na poziomie czynności, zdarzenia i przejścia workflow.

Przykład: Instancja workflow jest inicjowana przy zatwierdzeniu zapotrzebowania zakupu. Tworzy ona zlecenie zakupu zgodnie z zapotrzebowaniem. Jeśli zatwierdzone zostało zapotrzebowanie R1, stanowi ono obiekt odwołania zdarzenia początkowego. Następnie obiektem odwołania staje się zlecenie zakupu PO1, to znaczy zlecenie wygenerowane z zapotrzebowania zakupu.

Ograniczanie instancji workflow w toku dla danego obiektu

Opcja *Pojedyncza aktywna instancja workflow dla obiektu* (nazwa SQL: **bUniqueActive**)

AssetCenter umożliwia ograniczenie liczby jednoczesnych instancji workflow dla danego obiektu za pomocą opcji **Pojedyncza aktywna instancja workflow dla obiektu** w zakładce **Ogólne** szczegółów workflow.

Jeśli występuje zdarzenie czynności początkowej powodujące powstanie drugiej instancji workflow dla obiektu, opcje **Pojedyncza aktywna instancja workflow dla obiektu** i **Ponownie zainicjuj instancję workflow jeżeli jest w toku** (nazwa SQL: **bReinitialize**) (zakładka **Ogólne** szczegółów zdarzenia), określają wynik:

W poniższej tabeli podsumowano różne możliwe przypadki:

		Opcja <i>Pojedyncza aktywna instancja workflow dla obiektu</i> w zakładce <i>Ogólne</i> schematu workflow.	
		Zatwierdzona	Nie zatwierdzona
Opcja <i>Ponownie zainicjuj instancję workflow jeżeli jest w toku</i> w zakładce <i>Ogólne</i> zdarzenia wynikowego czynności „Start”.	Zatwierdzona	Jeśli instancja workflow dla obiektu jest już w toku, jest ona zatrzymywana i uruchamiana jest nowa instancja workflow.	
	Nie zatwierdzona	Jeśli instancja workflow dla obiektu jest już w toku, zdarzenie jest ignorowane (brak nowej instancji workflow).	Tworzona jest nowa instancja workflow.

Przykład zastosowania:

W przypadku schematu workflow, w którym mają być prowadzone zatwierdzenia zapotrzebowań zakupu, może się okazać użyteczne, aby:

- Zaznaczyć opcję **Pojedyncza aktywna instancja workflow dla obiektu**, aby dane zapotrzebowanie zakupu podlegało jednemu procesowi akceptacji.
- Zaznaczyć opcję **Ponownie zainicjuj instancję workflow jeżeli jest w toku** na poziomie zdarzenia początkowego, aby uruchomić ponownie instancję, jeśli treść zapotrzebowania zakupu została zmodyfikowana.

Role workflow

Zadania wynikające z niektórych czynności workflow muszą być wykonywane przez osobę przypisaną.

Uwaga: Osoby przypisane do czynności biorą udział tylko w czynnościach typu **Pytanie** lub **Działanie użytkownika**. Czynności typu **Działanie automatyczne** lub **Test / skrypt** nie mają osób przypisanych.

Osoby przypisane do czynności są wybierane w tabeli ról workflow (nazwa SQL: amWfOrgRole). Użyj polecenia menu Narzędzia/Workflow/ Role, aby uzyskać dostęp do tabeli ról workflow.

Typ roli workflow

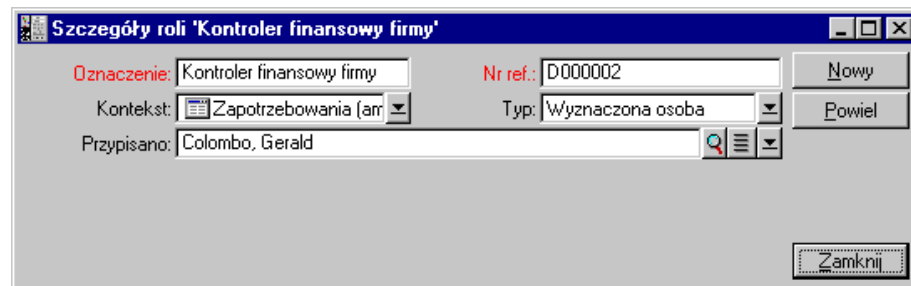
Istnieje kilka typów ról workflow (pole **Typ** (nazwa SQL: seType) w szczegółach roli workflow):

- **Wyznaczona osoba.**
- **Osoba kalkulowana.**
- **Grupa.**

Wyznaczona osoba

W tym przypadku, osoba przypisana wybierana jest bezpośrednio z tabeli wydziałów i pracowników.

Przykład:

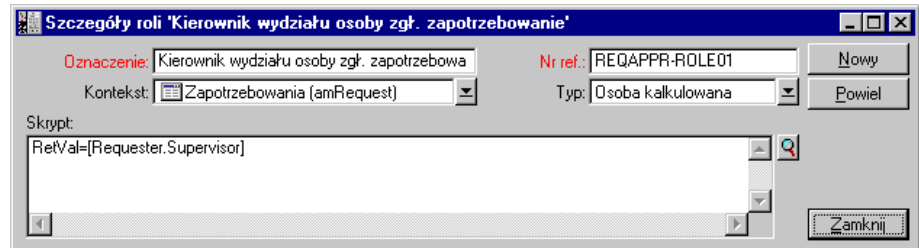


The screenshot shows a window titled "Szczegóły roli 'Kontroler finansowy firmy'". It contains several input fields and buttons. The "Oznaczenie" field is set to "Kontroler finansowy firmy". The "Nr ref." field is "D000002". The "Kontekst" dropdown is set to "Zapotrzebowania [am]". The "Typ" dropdown is set to "Wyznaczona osoba". The "Przypisano" field is "Colombo, Gerald". There are buttons for "Nowy", "Powiel", and "Zamknij".

Osoba kalkulowana

W tym przypadku, osoba przypisana należy do tabeli wydziałów i pracowników, ale jest wyliczana przez skrypt.

Przykład:



Grupa

W tym przypadku, osoba przypisana wybierana jest z tabeli grup pracowników (nazwa SQL: „amEmplGroup”).

Definiowanie osoby przypisanej do czynności

Pole **Przypisano** (nazwa SQL: Assignee) w zakładce **Pytanie** (czynność typu **Pytanie**) lub zakładka **Działanie** (czynność typu **Działanie użytkownika**) pozwala zdefiniować osobę przypisaną do czynności typu **Pytanie** lub **Działanie użytkownika**.

Czynności

Czynności można podzielić na dwie grupy:

- Wymagające interakcji użytkownika: Czynności typu **Pytanie** i **Działanie użytkownika** (pole **Typ** (nazwa SQL: seType) u góry szczegółów czynności).
- Wykonywane automatycznie: czynności typu **Działanie automatyczne** i **Test / skrypt**.

Wartość pola **Typ** czynności warunkuje, które zakładki są wyświetlane w szczegółach czynności.

W tej sekcji opisano następujące czynności:

- Czynności typu Pytanie
- Czynność typu Działanie użytkownika
- Czynność typu Działanie automatyczne

- Czynności typu Test/ Skrypt
- Czynność Początek
- Szablony czynności
- Inicjowanie czynności

Czynności typu Pytanie

Czynności te wymagają udziału użytkownika określonego w polu **Przypisano** (nazwa SQL: Assignee).

Czynności typu **Pytanie** są definiowane przez:

- Pytania lub instrukcje.
- Możliwe odpowiedzi.

Przykłady:

- W toku akceptacji zapotrzebowania, do kierownika wydziału dostarczono zapotrzebowanie wystawione przez pracownika.
- Czynność typu **Pytanie** może również służyć jako punkt kontrolny potwierdzający, że zadanie zostało wykonane. W tym przypadku możliwa byłaby jedna odpowiedź.

Ustaw wartość pola **Typ** (nazwa SQL: seType) jako **Pytanie**. Zostanie wyświetlona zakładka **Pytanie**.

Określ:

- Rekord w tabeli ról workflow odpowiadający osobie przypisanej. Osoba przypisana może być poinformowana za pomocą działania AssetCenter. Aby to zrobić, trzeba odpowiednio wypełnić pole **Zawiadom osobę** (nazwa SQL: bNotifAssignee) w zakładce **Pytanie**.

Uwaga: Działanie informujące osobę przypisaną jest inicjowane, gdy tylko zostanie utworzone zadanie do wykonania, to znaczy, gdy tylko przejście inicjujące czynność zostanie aktywowane.

Osoba przypisana za pomocą polecenia menu **Narzędzia/ Zadania w toku** uzyskuje dostęp do szczegółów zadania do wykonania.

- Tekst pytania lub instrukcji do wykonania.

- **Możliwe odpowiedzi.** Każda odpowiedź jest opisana w zakładce podrzędnej. Jest ona oznaczona opisem i nazwą SQL. Aby dodać, powielić lub usunąć odpowiedź, kliknij prawym przyciskiem myszy obszar etykiety zakładki podrzędnej i wybierz polecenie **Dodaj związany rekord**, **Powiel związany rekord** lub **Usuń łącze** z menu kontekstowego.

Uwaga: Każda odpowiedź tworzy automatycznie zdarzenie wynikowe czynności.

Czynność typu Działanie użytkownika

Czynności te wymagają udziału użytkownika, zwanego „osobą przypisaną”. Osoba przypisana widnieje w polu **Przypisano** (nazwa SQL: Assignee).

Ich definicje obejmują:

- Instrukcje do wykonania.
- Kreator do uruchomienia.

Ustaw wartość pola **Typ** jako **Działanie użytkownika**, aby została wyświetlona zakładka **Działanie**.

Określ:

- Instrukcje do wykonania.
- Kreator do uruchomienia.
- Rekord w tabeli ról workflow właściwy dla osoby przypisanej. Osobę przypisaną można poinformować za pomocą działania AssetCenter. Aby to zrobić, trzeba odpowiednio wypełnić pole **Zawiadom osobę** (nazwa SQL: bNotifAssignee) w zakładce **Działanie**.

Uwaga: Działanie informujące osobę przypisana jest inicjowane, gdy tylko zostanie utworzone zadanie do wykonania, to znaczy, gdy tylko zostanie uaktywnione przejście inicjujące czynność.

Osoba przypisana za pomocą polecenia menu **Narzędzia/ Zadania w toku** uzyskuje dostęp do szczegółów zadania do wykonania.

Uwaga: Zdarzenie „wykonane” jest automatycznie tworzone jako zdarzenie wynikowe czynności.

Przykład: Przy zarządzaniu dostawami, można zastosować kreatora, aby pomóc użytkownikowi przyjąć dostawę w całości lub części linii zlecenia zakupu oczekujących na dostawę.

Czynność typu Działanie automatyczne

Czynności są wykonywane automatycznie przez AssetCenter lub Serwer Asset Center.

Opis

Czynności typu **Działanie automatyczne** podają działania do wykonania.

Przykład: W operacji „Ruch zasobu”, czynność typu **Działanie automatyczne** automatycznie modyfikuje lokalizację wszystkich zasobów, których zasoby zwierzchnie zostały przeniesione.

Ustaw wartość pola **Typ** jako **Działanie automatyczne**, aby została wyświetlona zakładka **Działania**.

Podaj tam listę działań do wykonania.

Uwaga: Zdarzenie „wykonane” jest tworzone automatycznie jako zdarzenie wynikowe czynności.

Wykonanie

- Jeśli zaznaczysz opcję **Wykonaj działania natychmiast** (nazwa SQL: bExecImmediately), motor workflow, który aktywuje przejście inicjujące czynność, automatycznie wykonuje działania należące do czynności: Serwer Asset Center lub agent AssetCenter wykonuje działania, w zależności od wybranego trybu przetwarzania dla zdarzenia, które inicjuje przejście.
- W przeciwnym razie, zadania są wykonywane przez Serwer Asset Center podczas następnego cyklu weryfikacji.

Czynności typu Test/ Skrypt

Czynności te są wykonywane automatycznie przez AssetCenter lub Serwer Asset Center.

Opis

Są one definiowane przez skrypt i możliwe wyniki.

Przykład: W dziedzinie zarządzania zapasem oraz zapotrzebowaniem zakupu, czynność typu test/ skrypt może być zastosowana, aby sprawdzić, czy produkty wymienione w liniach zlecenia zakupu są dostępne w zapasie i nie zarezerwowane. Jeśli tak, czynność może zainicjować czynność typu **Pytanie**, która pyta osoby zgłaszające zapotrzebowanie, czy chcą zarezerwować produkt w zapasie.

Ustaw **Typ** jako **Test / skrypt**, aby została wyświetlona zakładka **Test**.

Określ:

- Skrypt testu do wykonania.
- Możliwe wyniki. Każdy wynik jest opisany w osobnej zakładce podrzędnej. Jest on identyfikowany przez opis i nazwę SQL. Aby dodać, powielić lub usunąć możliwy wynik, kliknij prawym przyciskiem myszy obszar etykiety zakładki podrzędnej i wybierz polecenie **Dodaj związany rekord**, **Powiel związany rekord** lub **Usuń łącze** z menu kontekstowego.

Ważne: Nazwy SQL każdego wyniku muszą być zgodne z wartościami zwracanymi przez skrypt testu.

Uwaga: Każdy wynik automatycznie tworzy zdarzenie wynikowe czynności.

Wykonanie

- Jeśli zaznaczysz opcję **Wykonaj działania natychmiast** (nazwa SQL: bExecImmediately), motor workflow, który aktywuje przejście inicjujące czynność, automatycznie wykonuje działania należące do czynności: Serwer Asset Center lub agent AssetCenter wykonuje działania, w zależności od wybranego trybu przetwarzania dla zdarzenia, które inicjuje przejście.
- W przeciwnym razie, zadania są wykonywane przez Serwer Asset Center podczas następnego cyklu weryfikacji.

Czynność Początek

Czynność **Początek** jest punktem wyjściowym schematu workflow.

Jest ona obowiązkowa i tworzona automatycznie przy utworzeniu schematu workflow. Edycja szczegółów nie jest możliwa.

Nie definiuje ona zadania do wykonania.

Zdarzenia wynikowe czynności **Początek** inicjują instancję workflow.

Szablony czynności

Szablony czynności ułatwiają tworzenie czynności schematów workflow.

Są one przechowywane w tabeli czynności (nazwa SQL: „amWfActivity”).

Użyj polecenia menu Narzędzia/ Workflow/ Szablony czynności, aby uzyskać dostęp do listy szablonów czynności.

Ostrzeżenie: Aby informacje podane w szczegółach szablonu czynności (typ czynności, itp) były automatycznie kopiowane do poziomu czynności odwołujących się do tego szablonu (pole **Szablon** (nazwa SQL: Template) w szczegółach czynności), administrator AssetCenter musi zdefiniować odpowiednie wartości domyślne pól i łącz w szczegółach czynności.

Inicjowanie czynności

Aby czynność została zainicjowana, pole **Warunek wprowadzania** (nazwa SQL: seInCond) w zakładce **Ogólne** szczegółów czynności musi być wypełnione. Warunek ten dotyczy przejść, które inicjują czynność.

- Jeśli jest tylko jedno przejście, które może zainicjować czynność, przejście musi być uaktywnione (przez AssetCenter lub Serwer Asset Center), aby czynność została zainicjowana.
- Jeśli czynność może zostać zainicjowana przez kilka przejść:
 - ❖ Jeśli **I** jest warunkiem wejściowym czynności, muszą zostać uaktywnione wszystkie przejścia, aby czynność została zainicjowana.
 - ❖ Jeśli **LUB** jest warunkiem wejściowym czynności, tylko jedno przejście musi być uaktywnione, aby czynność została zainicjowana.

Uwaga: Jeśli warunki wejściowe czynności są złożone (połączenie I i LUB), można utworzyć sekwencję pośrednich czynności typu **Test / skrypt**, aby je zrealizować.

Zadania

Niniejszy rozdział wyjaśnia, w jaki sposób są tworzone i wykonywane zadania workflow:

- Tworzenie zadań
- Czynność typu działanie automatycznie lub test/ skrypt
- Wyświetlanie listy zadań w toku
- Wykonywanie zadania użytkownika
- Przypisywanie zadań użytkownika
- Administrowanie zadaniem workflow

Tworzenie zadań

Kiedy uaktywniane jest przejście inicjujące czynność, motor workflow, który uaktywnił przejście automatycznie tworzy zadanie do wykonania.

Zgodnie z opcją wybraną w polu **Rejestruj zadanie** (nazwa SQL: bLogWorkItem) w zakładce **Ogólne** czynności, zadanie to jest rejestrowane w tabeli zadań workflow (nazwa SQL: WkElem).

Opcja **Rejestruj zadanie** jest sprawdzana automatycznie:

- Dla czynności typu **Pytanie** lub **Działanie użytkownika**.
- Dla czynności typu **Działanie automatyczne** lub **Test / skrypt**, dla których opcja **Wykonaj działania natychmiast** (nazwa SQL: bExecImmediately) nie została wybrana.

Ostrzeżenie: Jeśli zadanie nie jest rejestrowane, nie można tworzyć alarmów workflow związanych z tym zadaniem: Zakładki **Limit czasu** i **Alarmy** w szczegółach czynności nie są wyświetlane, jeśli opcja **Rejestruj zadanie** nie została wybrana.

Zadanie jest wykonywane odmiennie, w zależności od tego, czy udział użytkownika jest wymagany, czy nie.

Czynność typu działanie automatycznie lub test/ skrypt

Jeśli zadanie wynika z czynności typu **Działanie automatyczne** lub **Test / skrypt**, dla której wybrano opcję **Wykonaj działania natychmiast** (nazwa SQL: bExecImmediately), zadanie jest wykonywane automatycznie przez motor workflow, który uaktywnił przejście tworzące zadanie. Może to być Serwer Asset Center lub agent AssetCenter.

W przeciwnym razie, Serwer Asset Center sprawdza w regularnych odstępach czasu, czy istnieje potrzeba wykonania zadań workflow i w razie potrzeby wykonuje je.

Częstotliwość, z którą Serwer Asset Center monitoruje funkcje workflow jest zdefiniowana w opcjach Serwer Asset Center.

Wyświetlanie listy zadań w toku

Polecenie menu Narzędzia/ Zadania w toku umożliwia wyświetlanie listy zadań do wykonania.

Lista wyświetlanych zadań zależy od tego, kto nawiązał połączenie z bazą danych:

- Administrator AssetCenter ma dostęp do wszystkich zadań w toku dla wszystkich instancji workflow.
- Osoba przypisana do workflow widzi:
 - ❖ Zadania, które ma wykonać.
 - ❖ Zadania, które są przypisane do grup, do których należy, ale nie są przypisane do danej osoby.

Administrator ma również dostęp do listy zadań w toku dla danego zadania ze szczegółów schematu workflow. Aby to zrobić:

- Kliknij czynność prawym przyciskiem myszy.
- Wybierz opcję **Zadania w toku** z menu kontekstowego.

Uwaga: Wyświetlana lista jest jedynie ograniczonym widokiem tabeli zadań workflow (nazwa SQL: „amWfWorkItem”): Ukazuje ona zadania do wykonania.

Wykonywanie zadania użytkownika

*Osoba przypisana do workflow może uzyskać dostęp do listy zadań do wykonania za pomocą polecenia menu **Narzędzia/ Zadania w toku**.*

Ostrzeżenie: Jeśli użytkownik połączony z bazą danych jest administratorem AssetCenter, polecenie menu **Narzędzia/ Zadania w toku** wyświetla wszystkie zadania do wykonania. Jeśli użytkownik połączony z bazą danych nie jest administratorem, polecenie menu **Narzędzia/ Zadania w toku** wyświetla tylko te zadania, które są do niego przypisane i zadania przypisane do grupy, do której użytkownik należy.

Aby uzyskać dostęp do szczegółów obiektu, do którego odwołuje się zadanie, kliknij [Szczegóły](#).

Aby wykonać zadanie oczekujące na wykonanie, wyświetl zakładkę **Ogólne** zadania:

- Jeśli czynność wynikająca z zadania jest zadaniem typu **Pytanie**, zakładka **Ogólne** wyświetla tekst pytania lub instrukcji do wykonania. Możliwe wyniki przedstawione są za pomocą przycisków. Kliknij właściwy przycisk. W razie potrzeby, możesz również wpisać komentarz dotyczący podjętej decyzji.
- Jeśli czynność jest typu **Działanie użytkownika**, kliknij przycisk **Kreator**, aby uruchomić kreatora.

Przypisywanie zadań użytkownika

Informacje dotyczące przypisywania zadań użytkownika znajdują się w zakładce **Przypisanie** w szczegółach zadania.

Jeśli masz niezbędne uprawnienia, możesz zmienić przypisanie zadania użytkownika:

- Wartość pola **Przypisanie** (nazwa SQL: seAssignment).
- Osoba przypisana do zadania.

Administrowanie zadaniem workflow

Informacje dotyczące administrowania zadaniem workflow znajdują się w zakładce **Administracja** szczegółów zadania.

Informacje te są dostępne tylko dla użytkowników z uprawnieniami administratora.

Zdarzenia

Zdarzenia są związane z czynnościami. Inicjują one przejścia do innych czynności.

Istnieją trzy możliwe typy systemowe zdarzenia na poziomie czynności. Typ systemowy zdarzenia jest określany przez pole **Typ systemowy** (nazwa SQL: seType) w szczegółach zdarzenia:

- Zdarzenie **System**.
- Zdarzenie **Użytkownik**.
- Zdarzenie **Alarm**.

W tej sekcji opisano zdarzenia workflow i sposób ich przetwarzania:

- Zdarzenia systemowe
- Zdarzenia alarmowe
- Zdarzenia użytkownika
- Ogólne warunki aktywacji
- Przetwarzanie zdarzenia
- Zastosowanie: Realizacja synchronicznego schematu workflow
- Zdarzenie końcowe

Zdarzenia systemowe

Zdarzenia typu **System** są automatycznie definiowane przez AssetCenter przy tworzeniu/ modyfikowaniu czynności.

Odpowiadają one różnym możliwym wynikom (rezultatam) pracy wykonanej w czynności:

- Reakcje na czynność typu **Pytanie**,
- Wyniki czynności typu **Test / skrypt**,
- Zdarzenie „wykonane” w przypadku czynności typu **Działanie użytkownika** lub **Działanie automatyczne**.

Przykład: Jeśli czynność stawia pytanie, na które można odpowiedzieć „Tak” lub „Nie”, tworzone są dwa zdarzenia systemowe na poziomie czynności, zwane „Tak” i „Nie”.

Zdarzenia alarmowe

Zdarzenia alarmowe czynności tworzone są przy definiowaniu alarmów czynności inicjujących zdarzenia.

Alarm jest definiowany w zakładce **Alarmy** szczegółów czynności. Zdarzenie ma tę samą nazwę co alarm.

Zdarzenia użytkownika

Zdarzenia użytkownika są niezależne od zadań wykonywanych w ramach czynności. Są one tworzone ręcznie za pomocą graficznego edytora workflow (polecenie **Dodaj zdarzenie** w menu kontekstowym).

Uwaga: Zdarzenia związane z czynnością **Początek** są zdarzeniami użytkownika.

Istnieją dwa typy zdarzeń **Użytkownik** (pole **Typ** (nazwa SQL: seMonitoringType) u góry szczegółów zdarzenia):

- **Baza danych**
- **Okresowo**

Zdarzenie typu Baza danych

Zdarzenia typu **Baza danych** pozwalają aktywować instancje workflow w odniesieniu do specyficznych rekordów.

Zdarzenie typu **Baza danych** występuje:

- Gdy zostają spełnione ogólne warunki aktywacji, określone w zakładce **Ogólne**.
- Gdy zostają zweryfikowane pewne parametry inicjujące na poziomie monitorowanych rekordów.

Parametry inicjujące zdarzenie typu **Baza danych** są opisane w zakładce **Parametry** szczegółów zdarzenia. Podane są tam następujące informacje:

- Monitorowane rekordy (mogą to być rekordy w tabeli podanej w kontekście lub rekordy związane.) Jeśli monitorowane rekordy są rekordami związanymi z tabelą podaną w kontekście, określ właściwe łącze w polu **Łącze / kontekst** (nazwa SQL: LinkToMonitTable).
- Warunki aktywacji zdarzenia dotyczące monitorowanego rekordu. Aby określić warunki aktywacji, możesz:
 - ❖ Zaznaczyć jedno lub więcej odpowiednich pól spośród **Wstaw** (nazwa SQL: bInsert), **Aktualizuj** (nazwa SQL: bUpdate) i **Usuń** (nazwa SQL: bDelete).

Jeśli zaznaczysz pole **Wstaw**, uwzględniane są tworzone rekordy.

Jeśli zaznaczysz pole **Aktualizuj**, możesz określić pola, które przy modyfikacji muszą być uwzględnione w polu **Monitorowane pola** (nazwa SQL: MonitFields). Aby wskazać kilka nazw pól, zastosuj przecinki, aby je oddzielić. Jeśli pole pozostanie puste, pola modyfikowane nie są uwzględniane.

Jeśli zaznaczysz pole **Usuń**, uwzględniane są rekordy usuwane.

Ostrzeżenie: Nie jest możliwe, aby warunkiem aktywacji zdarzenia było usunięcie obiektu odwołania kontekstu.

- ❖ Wpisz skrypt w obszarze **Skrypt** (nazwa SQL: memScript). Jeśli wpiszesz skrypt i zaznaczysz jedno lub więcej pól spośród **Wstaw**, **Aktualizuj** i **Usuń**, skrypt ogranicza warunki aktywacji.

Przykład: Jeśli zdarzenie ma być inicjowane, gdy cena ogółem istniejącego zapotrzebowania zostanie zmodyfikowana, wypełnij zakładkę **Parametry** w następujący sposób:

The screenshot shows a window with four tabs: 'Ogólne', 'Parametry', 'Historia', and 'Dokumenty'. The 'Parametry' tab is active. Under 'Rekordy do monitorowania', there is a dropdown menu for 'Łącze / kontekst' and a text field for 'Monitorowana tabela: amRequest'. Under 'Warunki aktywacji dla monitorowanych rekordów', there are three checkboxes: 'Wstaw' (unchecked), 'Aktualizuj' (checked), and 'Usuń' (unchecked). Below these is a text field for 'Monitorowane pola: dtTotalCostCv'. At the bottom is a large text area for 'Skrypt'.

Zakładka **Parametry** działania typu **Baza danych**

Gdy tylko wystąpi zdarzenie typu **Baza danych**, jest ono uwzględniane przez stację roboczą AssetCenter, na której ono wystąpiło. Sposób jego przetwarzania zależy od opcji wybranej w polu **Przetwarz.** (nazwa SQL: seProcessingMode) w zakładce **Ogólne** szczegółów zdarzenia.

Więcej informacji można znaleźć w sekcji pt. „Przetwarzanie zdarzenia”, na stronie 292 niniejszego podręcznika.

Uwaga: W przypadku zdarzenia typu **Baza danych**, zakładka **Parametry** zazwyczaj wystarcza do określenia monitorowanych rekordów. Rzadko zachodzi potrzeba użycia pola **Warunek AQL** (nazwa SQL: AQLCond) w zakładce **Ogólne** szczegółów zdarzenia.

Zdarzenie typu *Okresowo*

Zdarzenia typu **Okresowo** dotyczą wybranych rekordów w danej tabeli. Pozwalają one okresowo inicjować instancję workflow dla każdego rekordu z zaznaczonego obszaru.

Przykład: Co miesiąc aktualizowane są pozostałe wartości zasobów z kategorii „PC”.

Zdarzenie typu **Okresowo** występuje, jeśli spełnione są warunki aktywacji podane w zakładce **Ogólne**.

W tym przypadku, Serwer Asset Center inicjuje zdarzenie.

Częstotliwość, z jaką Serwer Asset Center inicjuje zdarzenia typu **Okresowo** jest definiowana w polu planowania w zakładce **Parametry** szczegółów zdarzenia.

Sposób przetwarzania zdarzenia opisano w sekcji pt. „Przetwarzanie zdarzenia”, na stronie 292 niniejszego podręcznika.

Ogólne warunki aktywacji

Warunki aktywacji dla wszystkich typów zdarzeń można definiować w zakładce **Ogólne**:

Warunek AQL (nazwa SQL: AQLCond)

Pole **Warunek AQL** określa wybór rekordów objętych schematem workflow.

Ponownie zainicjuj instancję workflow jeżeli jest w toku (nazwa SQL: bReinitialize)

Uwaga: Opcja **Ponownie zainicjuj instancję workflow jeżeli jest w toku** jest wyświetlana tylko w szczegółach tych zdarzeń, które wynikają z czynności „Początek”.

Opcja **Ponownie zainicjuj instancję workflow jeżeli jest w toku** określa, co się dzieje gdy zdarzenie wynikowe czynności **Początek** dotyczy obiektu bazy danych, który już jest objęty instancją tego schematu workflow.

To, co się dzieje, zależy nie tylko od tej opcji, lecz także od opcji **Pojedyncza aktywna instancja workflow dla obiektu** (nazwa SQL: bUniqueActive) w zakładce **Ogólne** schematu workflow.

W poniższej tabeli podsumowano różne możliwe przypadki:

		Opcja <i>Pojedyncza aktywna instancja workflow dla obiektu w zakładce Ogólne</i> schematu workflow.	
		Zatwierdzona	Nie zatwierdzona
Opcja <i>Ponownie zainicjuj instancję workflow jeżeli jest w toku w zakładce Ogólne</i> zdarzenia wynikowego czynności „Start”.	Zatwierdzona	Jeśli instancja workflow dla obiektu jest już w toku, jest ona zatrzymywana i uruchamiana jest nowa instancja workflow.	
	Nie zatwierdzona	Jeśli instancja workflow dla obiektu jest już w toku, zdarzenie jest ignorowane (brak nowej instancji workflow).	Tworzona jest nowa instancja workflow.

Przetwarzanie zdarzenia

Po spełnieniu ogólnych warunków aktywacji, sposób przetwarzania zdarzeń zależy od:

- „Typu” zdarzenia (pole **Typ** (nazwa SQL: seMonitoringType) u góry szczegółów zdarzenia).
- Opcji wybranej w polu **Przetwarz.** (nazwa SQL: seProcessingMode) w zakładce **Ogólne** w szczegółach zdarzenia.

W poniższej tabeli podano różne sposoby przetwarzania zdarzenia:

	<i>Rejestracja zdarzenia i przetwarzanie przez serwer</i>	<i>Rejestracja zdarzenia i przetwarzanie</i>	<i>Przetwarzanie natychmiastowe bez rejestracji</i>
Zdarzenie typu Okresowo	<p>Serwer Asset Center inicjuje zdarzenie, jeśli są spełnione warunki aktywacji. Częstotliwość inicjowania jest zdefiniowana w zakładce Parametry szczegółów zdarzenia.</p> <p>Natychmiast po jego wystąpieniu, Serwer Asset Center rejestruje zdarzenie w tabeli o nazwie SQL „amWfOccurEvent”.</p> <p>Przejście jest aktywowane później przez Serwer Asset Center (częstotliwość, z którą Serwer Asset Center monitoruje przejścia do uaktywnienia jest zdefiniowana w opcjach programu Serwer Asset Center).</p>	<p>Serwer Asset Center inicjuje zdarzenie, jeśli są spełnione warunki aktywacji. Częstotliwość inicjowania jest zdefiniowana w zakładce Parametry szczegółów zdarzenia.</p> <p>Natychmiast po jego wystąpieniu, Serwer Asset Center rejestruje zdarzenie w tabeli o nazwie SQL „amWfOccurEvent”.</p> <p>Przejście jest aktywowane natychmiast przez Serwer Asset Center.</p>	<p>Serwer Asset Center inicjuje zdarzenie, jeśli są spełnione warunki aktywacji. Częstotliwość inicjowania jest zdefiniowana w zakładce Parametry szczegółów zdarzenia.</p> <p>Natychmiast po jego wystąpieniu, Serwer Asset Center nie rejestruje zdarzenia w tabeli o nazwie SQL „amWfOccurEvent”. Jednak przejście jest aktywowane natychmiast przez Serwer Asset Center.</p>
Zdarzenie typu Baza danych lub Zdarzenia systemowe inicjowane przez AssetCenter (wynik czynności typu Pytanie lub Działanie użytkownika, wynik czynności typu Działanie)	<p>Natychmiast po wystąpieniu zdarzenia, jest ono rejestrowane w tabeli o nazwie SQL „amWfOccurEvent” przez stację roboczą AssetCenter.</p> <p>Przejście jest aktywowane później przez Serwer Asset Center (częstotliwość, z którą Serwer Asset Center monitoruje przejścia do</p>	<p>Natychmiast po wystąpieniu zdarzenia, jest ono rejestrowane w tabeli o nazwie SQL „amWfOccurEvent” przez stację roboczą AssetCenter.</p> <p>Przejście jest aktywowane natychmiast przez stację roboczą AssetCenter.</p>	<p>Po wystąpieniu zdarzenia, nie jest ono rejestrowane w tabeli o nazwie SQL „amWfOccurEvent”, jednak przejście jest aktywowane natychmiast przez stację roboczą AssetCenter.</p>

	Rejestracja zdarzenia i przetwarzanie przez serwer	Rejestracja zdarzenia i przetwarzanie	Przetwarzanie natychmiastowe bez rejestracji
automatyczne lub Test / skrypt wykonywanej przez AssetCenter)	uaktywnienia jest zdefiniowana w opcjach programu Serwer Asset Center).		
Zdarzenie systemowe inicjowane przez Serwer Asset Center (wynik czynności typu Działanie automatyczne lub Test / skrypt wykonywanej przez Serwer Asset Center) lub zdarzenia wyzwalanego przez alarm czynności	Natychmiast po wystąpieniu zdarzenia, jest ono rejestrowane w tabeli o nazwie SQL „amWfOccurEvent” przez Serwer Asset Center. Przejście jest uaktywniane później przez Serwer Asset Center (częstotliwość, z którą Serwer Asset Center monitoruje przejścia do uaktywnienia jest zdefiniowana w opcjach programu Serwer Asset Center).	Natychmiast po wystąpieniu zdarzenia, jest ono rejestrowane w tabeli o nazwie SQL „amWfOccurEvent” przez Serwer Asset Center. Przejście jest aktywowane natychmiast przez Serwer Asset Center.	Gdy występuje zdarzenie, nie jest ono rejestrowane w tabeli o nazwie SQL „amWfOccurEvent”, jednak przejście jest aktywowane natychmiast przez Serwer Asset Center.

Korzystając z poszczególnych trybów przetwarzania, można precyzyjnie określić przebieg instancji workflow.

Zgodnie z wyborami dokonanymi na poziomie:

- Typów zdarzeń,
- Sposobów przetwarzania zdarzeń,
- Czynności,

Możesz zdefiniować zarówno synchroniczne jak i asynchroniczne schematy workflow lub użyć obu tych metod łącznie.

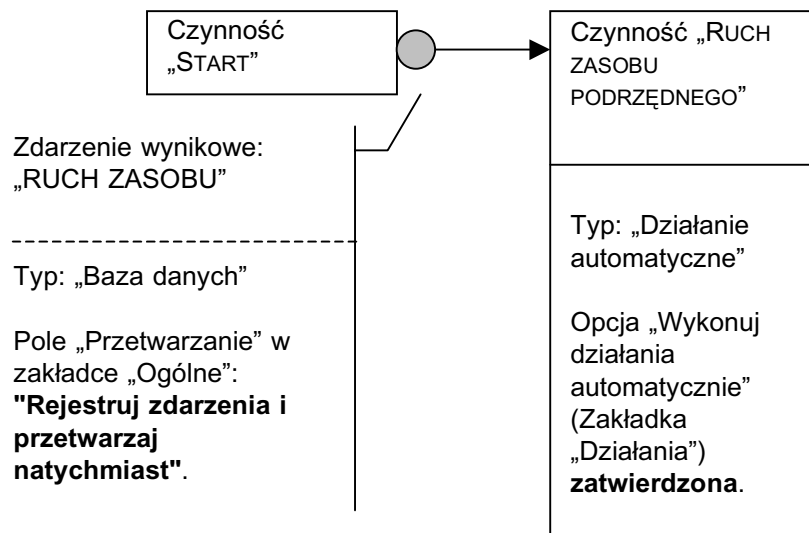
Zastosowanie: Realizacja synchronicznego schematu workflow

Aby zrealizować synchroniczny schemat workflow, trzeba zdefiniować:

- Zdarzenia typu **Baza danych**, a mianowicie **Rejestracja zdarzenia i przetwarzanie** (pole **Przetwarz.** (nazwa SQL: seProcessingMode) w zakładce **Ogólne** szczegółów zdarzenia).
- Czynności typu **Działanie automatyczne** lub **Test / skrypt**, dla których została wybrana opcja **Wykonaj działania natychmiast** (nazwa SQL: bExecImmediately) i które są inicjowane przez te zdarzenia.

Przykład:

Za pomocą schematu workflow opisanego w poniższym diagramie, natychmiast po zmianie lokalizacji przez zasób, jego zasoby podrzędne przenoszone są automatycznie do tej samej lokalizacji:



Przykład synchronicznego schematu workflow

W tym przypadku, gdy następuje zmiana lokalizacji zasobu i klikniesz

.

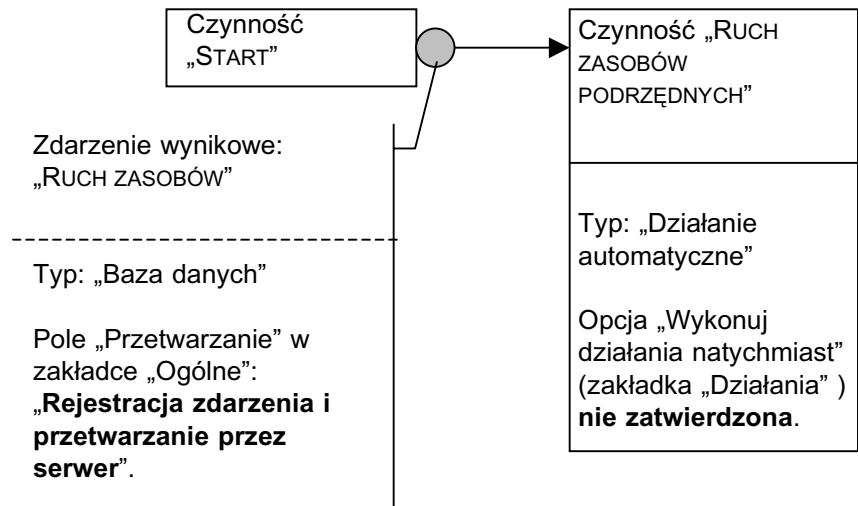
- ↳ Rozpoczyna się transakcja bazy danych.
- ↳ Lokalizacja zasobu zostaje zmodyfikowana.

- ↳ Uruchamiana jest instancja workflow.
- ↳ Aktywowane jest przejście workflow.
- ↳ Zmieniana jest lokalizacja zasobów podrzędnych.
- ↳ Następnie cała transakcja jest sprawdzana.

Jeśli podczas jednego z tych etapów wystąpi błąd, zarówno lokalizacje zasobu, jak i zasobów podrzędnych pozostają bez zmian.

Jeśli wykonanie czynności kończy się powodzeniem, zmodyfikowane zostają wszystkie lokalizacje.

Z drugiej strony, jeśli ta sama procedura jest realizowana za pomocą asynchronicznego schematu workflow opisanego poniżej, i jeśli wystąpi błąd, lokalizacja zasobu może ulec zmianie bez zmiany lokalizacji zasobów podrzędnych.



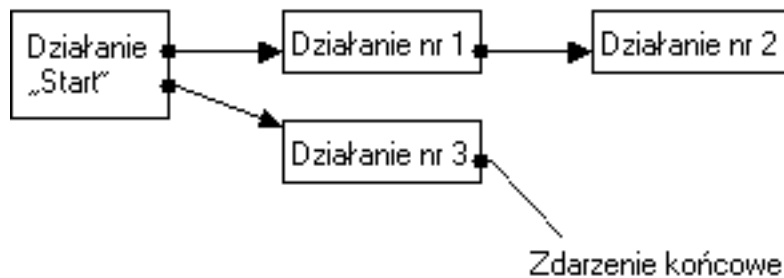
Przykład asynchronicznego schematu workflow

Zdarzenie końcowe

Definicja

Zdarzenie końcowe kończy instancję workflow, nawet jeśli pozostają zadania do wykonania.

Przykład:



Schemat workflow ze zdarzeniem końcowym

Jeśli występuje instancja workflow tak jak powyżej i:

- Następuje zdarzenie wynikowe czynności 1 i inicjuje czynność 2, tworząc zadanie do wykonania.
- Występuje zdarzenie końcowe czynności 3.

Instancja workflow zostaje zakończona, nawet jeśli zadanie wynikające z czynności 2 nie zostało wykonane.

Określanie zdarzenia końcowego

Tworząc schemat workflow za pomocą edytora graficznego w zakładce **Czynności** schematu workflow, możesz w następujący sposób określić zdarzenie końcowe:

- Kliknij zdarzenie prawym przyciskiem myszy.
- Wybierz opcję **Zdarzenie końcowe** z menu kontekstowego.

Przejścia workflow

Przejścia łączą zdarzenia wynikowe danej czynności z innymi czynnościami.

Zdarzenie może się łączyć z kilkoma przejściami.

W razie potrzeby, można wykorzystać pole **Warunek AQL** (nazwa SQL: AQLCond) w szczegółach przejścia, aby określić warunki aktywacji przejścia.

Alarmy workflow i limity czasu

Dla każdej czynności workflow można zdefiniować:

- Ostateczny termin wykonania.
- Alarmy połączone z tym terminem lub daty przechowywane w bazie danych. Alarmy te uruchamiają działania.

Ostrzeżenie: Bez zatwierdzenia opcji **Rejestruj zadanie** (nazwa SQL: bLogWorkItem) w zakładce **Ogólne** szczegółów czynności, nie można definiować limitów czasu lub alarmów.

W tej sekcji podano następujące punkty:

- Limit czasu
- Alarmy workflow

Limit czasu

Limit czasu na wykonanie czynności workflow jest zdefiniowany w zakładce **Limit czasu** szczegółów czynności.

Uwaga: Zakładka **Limit czasu** w szczegółach czynności jest wyświetlana wyłącznie, jeśli opcja **Rejestruj zadanie** (nazwa SQL: bLogWorkItem) w zakładce **Ogólne** szczegółów czynności jest zatwierdzona.

Limit czasu jest definiowany w odniesieniu do czasu, w którym czynność została zainicjowana.

Jest on związany z kalendarzem dni roboczych.

Możesz określić okres czasu lub wybrać jedną z trzech gotowych opcji:

- **Koniec następnego dnia roboczego**
- **Koniec tygodnia roboczego**
- **Koniec miesiąca roboczego**

Ostrzeżenie: Jeśli określisz okres czasu, AssetCenter uznaje go za wyrażony w czasie pracy i przelicza go na godziny robocze. Przykład: Jeśli wpiszesz „2 dni” jest to interpretowane jako 48 godzin roboczych.

Alarmy workflow

Alarmy można kojarzyć z każdą czynnością w zakładce **Alarmy** szczegółów czynności.

Uwaga: Zakładka **Alarmy** w szczegółach czynności jest wyświetlana wyłącznie, jeśli jest zatwierdzona opcja **Rejestruj zadanie** (nazwa SQL: bLogWorkItem) w zakładce **Ogólne** szczegółów czynności.

Limity czasu

Limity czasu inicjujące alarmy mogą być definiowane zgodnie z:

- Okresem czasu po dacie przechowywanej w bazie danych (typ **Upływ czasu od**).
- Okres czasu przed datą przechowywaną w bazie danych (typ **Czas do**).
- Procent limitu czasu na wykonanie czynności (pole **Limit czasu** (nazwa SQL: tsResolDelay) w zakładce **Limit czasu**).

Uwaga: Okresy czasu definiujące limity czasu workflow wyrażone są w dniach roboczych.

Natychmiast po utworzeniu zadania, generowane są związane z nim alarmy workflow.

Limity czasu workflow są monitorowane przez Serwer Asset Center. Częstotliwość monitorowania jest zdefiniowana w opcjach Serwer Asset Center.

Wynik alarmów

Alarmy inicjują:

- Działania AssetCenter.
- Lub zdarzenia. Zdarzenia inicjowane przez alarmy są zdarzeniami typu **Alarm**. Noszą one nazwę alarmów, które je zdefiniowały.

Grupy wykonawcze workflow

Grupy wykonawcze workflow pozwalają klasyfikować schematy workflow. Grupa wykonawcza, do której należy dany schemat workflow jest podana w polu **Grupa wykonawcza** (nazwa SQL: GroupName) w zakładce **Ogólne** szczegółów workflow.

Serwer Asset Center monitoruje tworzenie nowych grup wykonawczych workflow.

Gdy tylko Serwer Asset Center wykryje nową grupę wykonawczą G, tworzy nowy moduł monitorowania **Wykonanie zasad workflow dla grupy wykonawczej „G”**.

Mechanizm ten jest użyteczny ponieważ:

- Pozwala definiować harmonogramy weryfikacji dla każdej grupy wykonawczej workflow.
- Różne grupy wykonawcze workflow mogą być monitorowane przez różne instancje Serwer Asset Center.

Po wykryciu grupy wykonawczej workflow, Serwer Asset Center monitoruje i wykonuje właściwe zasady workflow (monitorowanie alarmów, przetwarzanie zdarzeń typu **Okresowo**, aktywacja przejść, wykonywanie zadań, itp).

Śledzenie workflow

Jeśli tabela w AssetCenter zdefiniowana jest jako kontekst obiektu początkowego schematu workflow, w widoku szczegółów tej tabeli wyświetlana jest zakładka **Workflow**.

W tej zakładce **Workflow** widnieje stan instancji workflow w toku, używających danego rekordu jako obiektu początkowego.

Każda instancja workflow jest opisana w zakładce podrzędnej, która opisuje postęp instancji:

- Po lewej stronie zakładki podrzędnej podane są zdarzenia, które wystąpiły.
- Po prawej stronie zakładki podrzędnej wyświetlony jest schemat workflow. Czynności do wykonania pulsują. Następne etapy oznaczone są kolorem szarym.

Rozdział 15 - Importowanie danych

W niniejszym rozdziale wyjaśniono, w jaki sposób importować dane za pomocą AssetCenter.

Rozdział zawiera informacje na następujące tematy:

- Omówienie importowania danych
- Zalecenia
- Ograniczenia, których należy przestrzegać przy importowaniu danych do pól
- Importowanie plików tekstowych lub bazy danych ODBC
- Import bazy danych AssetCenter
- Zapisywanie i wykonywanie skryptu importu
- Uruchamianie skryptu importu z wiersza poleceń systemu 32-bitowego

Aby zaimportować pojedynczy plik tekstowy, użyj polecenia menu Plik/ Import, a następnie opcję Importuj plik tekstowy.

Aby zaimportować kilka plików tekstowych, użyj polecenia menu Plik/ Import i opcję Importuj bazę danych.



Wybór typu danych do importu

Omówienie importowania danych

Administrator AssetCenter może importować dane do bazy danych AssetCenter z:

Pojedynczego pliku tekstowego

Plik tekstowy jest mapowany do głównej tabeli w bazie danych AssetCenter.

Każde pole w pliku tekstowym jest mapowane do pola w bazie danych, które może się znajdować w tabeli głównej lub tabeli związanej.

Kilku plików tekstowych

Każdy plik tekstowy jest mapowany do tabeli w bazie danych AssetCenter.

Każde pole w każdym pliku tekstowym jest mapowane do pola w bazie danych, które może się znajdować w tabeli głównej lub tabeli związanej.

Kompletnej bazy danych

Można zaimportować bazę danych:

- AssetCenter
- ODBC w przypadku innych baz danych

Każda tabela źródłowej bazy danych jest mapowana do tabeli w docelowej bazie danych.

Każde pole w tabeli źródłowej jest mapowane do pola w tabeli docelowej bazy danych. Pole docelowe może się znajdować w tabeli, która jest mapowana bezpośrednio, lub w tabeli dołączonej.

Można dodawać lub modyfikować rekordy bazy danych AssetCenter. Nie można ich jednak usuwać.

Wszystkie informacje dotyczące transferu mogą być zapisane w postaci skryptu. Skrypt ten można zastosować, aby ponownie zaimportować dane bez konieczności ponownego definiowania ustawień importu.

Moduł importu oferuje kilka możliwości obsługi błędów oraz możliwość zarejestrowania wyników przeprowadzonych czynności w pliku dziennika.

Uwaga dotycząca ochrony baz danych: Tylko administrator AssetCenter lub użytkownik z uprawnieniami administratora może korzystać z polecenia menu **Plik/ Import** (nazwa logowania „Admin”). To polecenie menu jest wyłączane dla innych użytkowników. Ogranicza to dostęp do bazy danych.

Uwaga dotycząca importowania danych numerycznych: Pola numeryczne muszą mieć prawidłowy format, bez względu na opcje zdefiniowane w Panelu sterowania systemem Windows: na przykład tylko znaki od 0 do 9; znak „.” jest używany jako separator dziesiętny. Wartości numeryczne muszą być importowane do pól numerycznych. W ten sposób dane numeryczne mogą być importowane niezależnie od ustawień komputera lub komputerów używanych do wykonania importu.

Uwaga dotycząca importowania obrazów, formularzy, zapytań, raportów, planów kondygnacji, uprawnień dostępu, widoków, haseł i „nazw logowania”: elementy te można zaimportować jedynie importując bazę danych AssetCenter.

Zalecenia

Oto kilka zaleceń do rozważenia przed zaimportowaniem danych do bazy danych AssetCenter:

Wartości domyślne pól obowiązkowych

Moduł importu uwzględnia obowiązkowe właściwości pól. Jeśli rekord do zaimportowania zawiera puste pole i jeśli to pole jest zdefiniowane jako obowiązkowe w docelowej bazie danych, moduł importu odrzuca ten rekord. Aby uniknąć odrzucania rekordów z tego powodu, zalecamy przypisywanie wartości domyślnych do pól obowiązkowych w docelowej bazie danych. Jeśli w importowanych plikach podana jest wartość pola, przesłania ona wartość domyślną pola.

Unikaj stosowania pól identyfikacyjnych jako kluczy uzgodnienia

Nie zalecamy stosowania pól identyfikacyjnych tabel jako kluczy uzgodnienia przy ponownym importowaniu wyeksportowanych danych, ponieważ nie są one stałe i mogą ulec zmianie. Należy stosować klucze, których wartości są „niezmienne”, jak na przykład **KodWewn** w przypadku zasobów.

Wykonaj kopię zapasową bazy danych AssetCenter

Ponieważ import może wprowadzić globalne zmiany w bazie danych AssetCenter, zalecamy wykonanie kopii zapasowej przed uruchomieniem importu.

Unikaj równoczesnego dostępu do bazy danych AssetCenter

Nie zalecamy wykonywania równoczesnych importów za pomocą kilku stacji roboczych lub używania AssetCenter na innej stacji roboczej podczas wykonywania importu.

Ograniczenia, których należy przestrzegać przy importowaniu danych do pól

Przy importowaniu danych do bazy danych AssetCenter, moduł importu sprawdza, czy dane są zgodne ze strukturą bazy danych. Jeśli niektóre pola lub łącza w danych do zaimportowania są niezgodne ze strukturą bazy danych, wyniki mogą być następujące:

- Rekord zostanie odrzucony w całości,
- Wartość zostanie obcięta,
- Pole zostanie pozostawione bez zmian.

Dlatego lepiej jest upewnić się, czy dane przedstawione do importu są zgodne ze strukturą bazy danych, aby uniknąć odrzucenia danych.

Warto uwzględnić dwie możliwości:

- Importujesz strukturalną bazę danych:
 - ❖ Jeśli pola w źródłowej i docelowej bazie danych są tego samego typu (na przykład w źródle i w celu jest pole typu data) lub jeśli pola są zgodne (na przykład w źródle jest pole typu data+czas, a w celu pole typu data), nie ma szczególnych ograniczeń, których należałoby przestrzegać.
 - ❖ Jeśli źródłowe pole jest typu „tekst” a docelowe pole jest typu bardziej specyficznego, należy przestrzegać ograniczeń pola docelowego.
- Importujesz dane z pliku tekstowego:
 - ❖ Musisz przestrzegać ograniczeń pola docelowego.

Zgodnie z wartością właściwości „UserType” pola docelowego

Jeśli wartość jest:	Wtedy:
Domyślna	Musisz przestrzegać formatu zdefiniowanego we właściwości „Typ”.
Numeryczna lub pieniężna	<p>Pole źródłowe musi być numeryczne.</p> <p>Jeśli pole źródłowe pochodzi ze strukturalnej bazy danych, a typ pola jest „numeryczny” lub „pieniężny”, nie występują ograniczenia.</p> <p>Jeśli pole źródłowe jest typu „tekstowego”, musi ono zawierać wartość numeryczną, ze znakiem „,” jako separatorem dziesiętnym, bez separatora dla tysięcy.</p>
Tak/Nie	Pole źródłowe musi zawierać wartość „1” (na „Tak”) lub „0” (na „Nie”).
Data	<p>Jeśli pole źródłowe pochodzi ze strukturalnej bazy danych i jest ono typu „Data” lub „Data+Czas”, nie występują żadne ograniczenia.</p> <p>Jeśli pole źródłowe jest typu „tekstowego”, musisz przestrzegać następujących ograniczeń:</p> <ul style="list-style-type: none"> ✓ Format daty (kolejność roku, miesiąca i dnia) musi być taki sam dla wszystkich rekordów. Określ ten format przy importowaniu. ✓ Dzień, miesiąc i rok należy wprowadzać systematycznie. ✓ Stosuj ten sam separator (dowolny) do oddzielania dni, miesięcy i lat we wszystkich rekordach. Podaj ten separator przy importowaniu.
Listą pozycji systemu	<p>Należy podać tylko jedną wartość listy pozycji. W przeciwnym razie linia nie zostanie zaimportowana. Można zidentyfikować element listy pozycji na podstawie jego czytelnej wartości lub na podstawie numeru przechowywanego w bazie danych.</p> <p>Na przykład: Jeśli lista pozycji ma postać „Tak 1 Nie 0”, zaimportowanie „Tak” lub „1” da ten sam wynik.</p> <p>Jeśli pole nie będzie miało wartości, moduł importu przypisze do pola wartość „0”.</p> <p>Zalecamy przedstawianie składowanych wartości numerycznych, ponieważ są one bardziej stabilne niż ich odpowiedniki tekstowe przy aktualizacji do nowszych wersji programu AssetCenter oraz niezależne od używanej wersji językowej.</p>

Jeśli wartość jest:	Wtedy:
Niestandardową listą pozycji	Wartość pola jest wybierana z listy pozycji, która może być modyfikowana przez użytkownika. Modułowi importu można przedstawić jedną z wartości listy pozycji. Jeśli dodatkowo lista pozycji jest „otwarta”, można przedstawić dowolną wartość. Wartość ta zostanie dodana do listy pozycji.
Wartością procentową	Importowane wartości muszą być wartościami procentowymi z lub bez znaku „%” (na przykład: „10” lub „10%”).
Okresem czasu	Należy przestrzegać ograniczeń zdefiniowanych przez właściwości „UserType” i „UserTypeFormat” z typem użytkownika określonym jako „Okres czasu”.
Nazwą SQL tabeli lub pola	Można zaimportować dowolną wartość alfanumeryczną. Jeśli jednak wartość ta nie odpowiada poprawnej nazwie SQL pola lub tabeli, istnieje ryzyko uszkodzenia bazy danych.

Zgodnie z wartością właściwości „typ” pola docelowego

Należy przestrzegać ograniczeń związanych z tą właściwością, jeśli właściwość „UserType” jest ustawiona na wartość „Domyślny”.

Przypadek szczególny: Jeśli wartość ma postać:	Wtedy:
Data+Czas	<p>Jeśli pole źródłowe pochodzi ze strukturalnej bazy danych, a typem pola jest „Data” lub „Data+Czas”, nie występują żadne ograniczenia.</p> <p>Jeśli pole źródłowe jest typu „tekstowego”, należy przestrzegać ograniczeń zdefiniowanych przez właściwość „UserType”, jeśli właściwość UserType ma wartość „Data” lub „Data+Czas”.</p>

Zgodnie z wartością innych właściwości pola docelowego

Jeśli właściwość:	Jest ustawiona na:	Wtedy:
MandatoryType	Tak	Jeśli pole źródłowe jest puste, rekordy, które moduł importu powinien dodać lub zmodyfikować pozostają bez zmian.
Size	Jest wypełnione	Zbyt długie wartości z pola źródłowego są obcinane przy imporcie.
ReadOnly	Tak	Nie można importować wartości do pola o tej właściwości.

Pole „dtLastModif” (nazwa SQL)

Pole to jest aktualizowane przy modyfikacji lub tworzeniu rekordów w bazie danych AssetCenter za pomocą interfejsu użytkownika lub za pomocą importu. Podana w nim jest data modyfikacji lub utworzenia rekordu.

Jeśli do tego pola zostanie zaimportowana wartość, rzeczywista data importu zostanie zastąpiona.

Importowanie plików tekstowych lub bazy danych ODBC

W niniejszej sekcji podano informacje o tym, w jaki sposób zaimportować jeden lub więcej plików danych lub bazę danych ODBC .

Omówiono następujące tematy:

- Przed importem plików tekstowych
- Step 1- Select the text files or ODBC databaseKrok 1: Wybór plików tekstowych lub bazy danych ODBC do zaimportowania
- Krok 2: Definiowanie analizy składniowej plików lub tabel do zaimportowania
- Krok 3: Opis importowanych pól
- Krok 4: Mapowanie pól źródłowych do pól docelowych w bazie danych AssetCenter

- Krok 4: Mapowanie każdego pliku tekstowego lub tabeli źródłowej do tabeli docelowej
- Krok 4: Mapowanie importowanych pól do pól w bazie danych AssetCenter
- Krok 4: Dodawanie dodatkowych pól wyliczanych w pliku źródłowym
- Krok 4: Przypadki szczególne
- Krok 4: Przykłady używania kluczy
- Krok 5: Konfiguracja transferu
- Krok 6: Transfer danych

Aby zaimportować pojedynczy plik tekstowy, użyj polecenia menu Plik/ Import, opcja Importuj plik tekstowy.

Aby zaimportować zestaw plików tekstowych, użyj polecenia menu Plik/ Import , opcja Importuj bazę danych, zakładka Tekst.

Aby zaimportować bazę danych ODBC, użyj polecenia menu Plik/ Import , opcja Importuj bazę danych, zakładkę ODBC.

Przed importem plików tekstowych

Rozpocznij od przygotowania plików tekstowych zawierających dane do zaimportowania. Muszą one spełniać następujące warunki:

- Każda kolumna oznacza pole.
- Każda linia oznacza rekord.
- Na początku pliku można opcjonalnie umieścić nazwy pól. Jeśli nie umieścisz ich w pliku, będzie można je zdefiniować przy imporcie danych.
- Na początku pliku, przed opcjonalną linią zawierającą nazwy pól, można umieścić maksymalnie 100 linii zawierających komentarze, które nie zostaną zaimportowane. Komentarze te nie muszą się zaczynać od jakiegoś specjalnego znaku. Liczbę linii podaje się programowi AssetCenter przy importowaniu pliku.
- Plik może korzystać z zestawu znaków OEM DOS (DOS) lub ANSI (Windows).
- Kolumny mogą być o stałej lub zmiennej szerokości. W tym przypadku wybierasz znak, który jest separatorem.

- Treść pola może być rozgraniczona dowolnym wybranym znakiem.
- Pola mogą być typu numerycznego, znakowego lub daty.
- Importowane wartości muszą spełniać ograniczenia związane ze strukturą bazy danych AssetCenter (format wprowadzania danych, typ pola, typ łącza, typ indeksu.)
- Dopuszczalne są wszystkie znaki za wyjątkiem tych, które zostały wybrane jako separator. Nie można włączyć znaku separatora do ciągu znaków tekstowych.
- Dane typu data, data+czas i czas trwania podlegają tym samym ograniczeniom, jak gdyby były wprowadzane w programie.
- Zalecamy utworzenie jednego pliku tekstowego dla każdej głównej tabeli docelowej.

Jeśli twoje dane źródłowe znajdują się w bazie danych, której motor nie jest obsługiwany przez moduł importu, trzeba wydobyć dane do plików tekstowych i zaimportować je. Jeśli AssetCenter rozpoznaje bazę danych, można zaimportować informacje bezpośrednio z bazy danych za pomocą polecenia menu **Plik/ Import**, opcji **Importuj bazę danych**.

Krok 1: Wybór plików tekstowych lub bazy danych ODBC do zaimportowania


Import pojedynczego pliku tekstowego

- Uruchom polecenie menu **Plik/ Import**.
- Wybierz opcję **Importuj plik tekstowy**.
- AssetCenter prosi o wskazanie lokalizacji strukturalnego pliku tekstowego zawierającego dane do zaimportowania.

Import zestawu plików tekstowych

Aby wybrać przygotowane pliki tekstowe:

↳ Uruchom polecenie menu **Plik/ Import**, opcję **Importuj bazę danych**, zakładkę **Tekst**.

↳ Kliknij .


- ↳ Po uruchomieniu modułu importu, wybierz polecenie menu **Plik/ Dodaj plik**. Wskaż nazwy plików tekstowych do dodania.


Importowanie bazy danych ODBC

Wybierz polecenie menu **Plik/ Import** i opcję **Importuj plik tekstowy**, aby wybrać źródłową bazę danych.

Za pomocą zakładki **ODBC** można zaimportować bazę danych ODBC.

- ↳ Określ źródło danych, użytkownika i hasło.

Ikona  umożliwia bezpośrednie utworzenie źródła danych ODBC, bez konieczności otwierania Panelu sterowania systemem Windows, a następnie aplikacji Administrator źródeł danych ODBC.

- ↳ Kliknij .

Krok 2: Definiowanie analizy składniowej plików lub tabel do zaimportowania


Ważna uwaga: Jeśli plik tekstowy zawiera pola o stałej szerokości, w tym kroku wyświetlane jest tylko jedno okno. Jeśli między polami stosowane są separatory, wyświetlane są dwa okna.

Import pojedynczego pliku tekstowego

Po wybraniu pliku tekstowego w kroku 1, AssetCenter pyta automatycznie, jak powinien być on analizowany.

Import zestawu plików tekstowych

Po wybraniu pliku tekstowego w kroku 1, AssetCenter pyta, jak powinien być on analizowany. Czynność tę można wykonać później, klikając

, lub wpisać informacje od razu.

Są dwa sposoby przywołania tego okna z głównego ekranu modułu importu:

- ❖ Kliknij dwukrotnie plik źródłowy lub
- ❖ Użyj polecenia menu **Edycja/ Właściwości** po wybraniu pliku źródłowego.

Listę plików tekstowych można znaleźć w kolumnie „Tabele źródłowe” głównego ekranu modułu importu.

Import bazy danych ODBC

Po otwarciu bazy danych w kroku 1, możesz wyświetlić opis każdej tabeli z głównego ekranu modułu importu:

- ❖ Kliknij dwukrotnie tabelę źródłową lub
- ❖ Użyj polecenia menu **Edycja/ Właściwości** po wybraniu tabeli źródłowej.

Listę tabeli źródłowych można znaleźć w kolumnie „Tabele źródłowe” głównego ekranu modułu importu.

Pierwszy ekran

Zestaw znaków

Podaj, czy tekst korzysta z zestawu znaków ANSI (Windows), czy ASCII (DOS).

Pierwsza linia importu

Wprowadź numer linii zawierającej pierwszą linię danych do zaimportowania. Moduł importu pomija linie przed tym numerem.

Jeśli dokument zawiera linię z nazwami pól, a linia ta znajduje się bezpośrednio przed pierwszą linią danych, wpisz numer linii z nazwami pól.

AssetCenter może pominąć maksymalnie 100 linii na początku pliku.

Pierwsza importowana linia zawiera nazwy kolumn

Zaznacz to pole, jeśli pierwsza importowana linia zawiera nazwy pól (nazwy kolumn.) Dzięki temu nie trzeba będzie ich wprowadzać ręcznie.

Jeśli plik nie zawiera nazw kolumn, można je zdefiniować w kolejnych krokach.

Według separatorów

Zaznacz to pole, jeśli wartości pól są rozdzielone ustalonym znakiem.

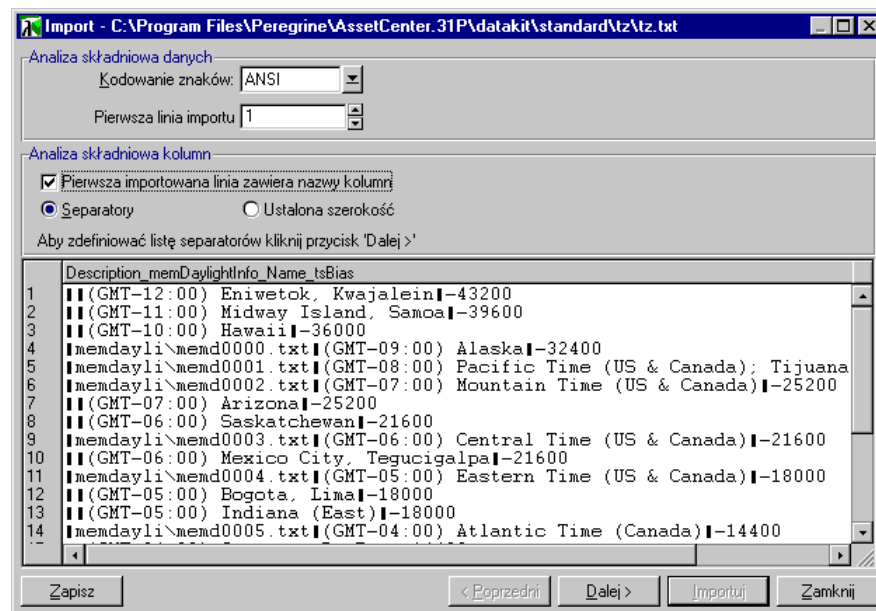
Podaj ten znak w następnym ekranie.

Stała szerokość

Zaznacz to pole, jeśli wszystkie wartości każdego pola mają tę samą długość.

AssetCenter automatycznie wyświetla granice kolumn.

- ❖ Aby przesunąć granicę kolumny, wybierz granicę w obszarze danych za pomocą myszy i przeciągnij ją w żądane miejsce. Nie można wybrać granicy w obszarze tytułu.
- ❖ Aby usunąć granicę kolumny, wybierz ją za pomocą myszy i przeciągnij ją poza tabelę.
- ❖ Aby dodać nową granicę kolumny, kliknij myszą w obszarze danych w miejscu, gdzie powinien być wstawiony separator.



Definiowanie analizy składniowej plików tekstowych (ekran 1)

Uwaga: W dolnej części ekranu wyświetlany jest podgląd importowanego pliku. Maksymalnie pokazane jest 25 linii.

Opcjonalny drugi ekran

Drugi ekran jest wyświetlany, jeśli na pierwszym ekranie:

↳ Określono, że wartości pól rozdzielone są znakami separatora.

↳ Kliknięto .

Separatory kolumn

Podaj znak używany jako separator dwóch kolejnych wartości pól. Zalecamy stosowanie znaku „;” jako separatora.

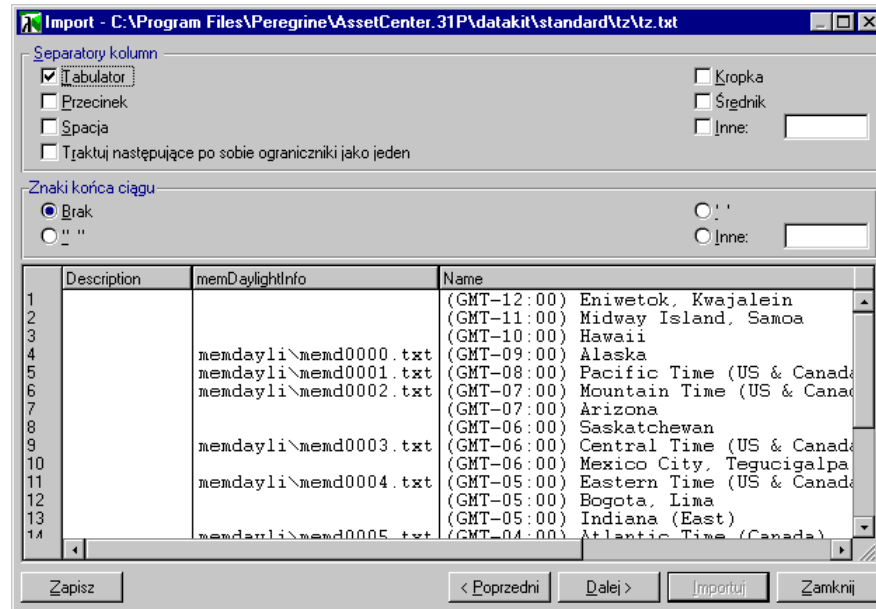
Jeśli zaznaczono opcję **Traktuj następujące po sobie ograniczniki jako jeden**, AssetCenter będzie traktować identyczne kolejne separatory jako jeden i nie będzie tworzyć pustych kolumn. Jeśli chcesz utworzyć pustą kolumnę, usuń zaznaczenie z tego pola, aby umożliwić używanie dwóch kolejnych separatorów w pliku tekstowym.

Ograniczniki ciągów znakowych

Jeśli stosowane są znaki obejmujące tekst, podaj stosowany znak. AssetCenter usunie dodatkowe znaki, jeśli będą występować, przed przeniesieniem pola do bazy danych.

Jeśli pomiędzy dwoma znakami ograniczenia AssetCenter napotka separator kolumny, zostanie on potraktowany jako tekst. Podanie znaku ogranicznika ciągów znakowych nie wprowadza wymogu systematycznego stosowania go wokół wszystkich wartości. Z drugiej strony, jeśli ogranicznik zostanie umieszczony na początku ciągu znakowego, należy umieścić go również na końcu tego ciągu.

Nie można zaimportować ogranicznika ciągu znaków jako wartości.




Definiowanie analizy składniowej plików tekstowych (ekran 2)


Uwaga: W dolnej części ekranu wyświetlany jest podgląd importowanego pliku. Maksymalnie pokazane jest 25 linii.

Krok 3: Opis importowanych pól


Import pojedynczego pliku tekstowego

Po przeanalizowaniu pliku tekstowego w kroku 2, kliknij przycisk , aby wyświetlić ekran opisu pól znajdujących się w pliku.

Import zestawu plików tekstowych

Po przeanalizowaniu pliku tekstowego w kroku 2, kliknij przycisk , aby wyświetlić ekran opisu pól znajdujących się w pliku.

Ekran ten można wyświetlić z głównego okna modułu importu klikając podwójnie plik źródłowy lub za pomocą polecenia menu **Edycja/**

Właściwości po wybraniu pliku i kliknięciu przycisku  raz lub dwa razy, zgodnie z „analizą kolumn” wybraną w poprzednim kroku.

Import bazy danych ODBC

Po wybraniu bazy danych w kroku 1, możesz wyświetlić opis tabel z głównego ekranu modułu importu. Kliknij dwukrotnie tabelę źródłową lub wybierz polecenie menu **Edycja/ Właściwości** po wybraniu tabeli źródłowej (kolumna „Tabele źródłowe” w głównym ekranie modułu importu podaje listę tabel źródłowych.)

Wybierz skonfigurowaną kolumnę klikając w dowolnym miejscu.

Numer

Tu wyświetlany jest numer wybranej kolumny.

Jeśli nie wybrano opcji **Pierwsza linia zawiera nazwy kolumn** w kroku 2, możesz bezpośrednio wybrać numer kolumny, zamiast klikać tabelę.

Nazwa

Tu wyświetlana jest nazwa kolumny (lub nazwa pola).

- Jeśli wybrano opcję **Pierwsza linia zawiera nazwy kolumn** w kroku 2, nie można zmodyfikować nazwy kolumny.
- Jeśli nie wybrano tej opcji, pozostaw domyślną nazwę lub zmodyfikuj ją. Nazwa ułatwia identyfikację kolumn w późniejszych etapach.

Typ

Wpisz typ importowanego pola. AssetCenter daje wybór spośród następujących możliwości:

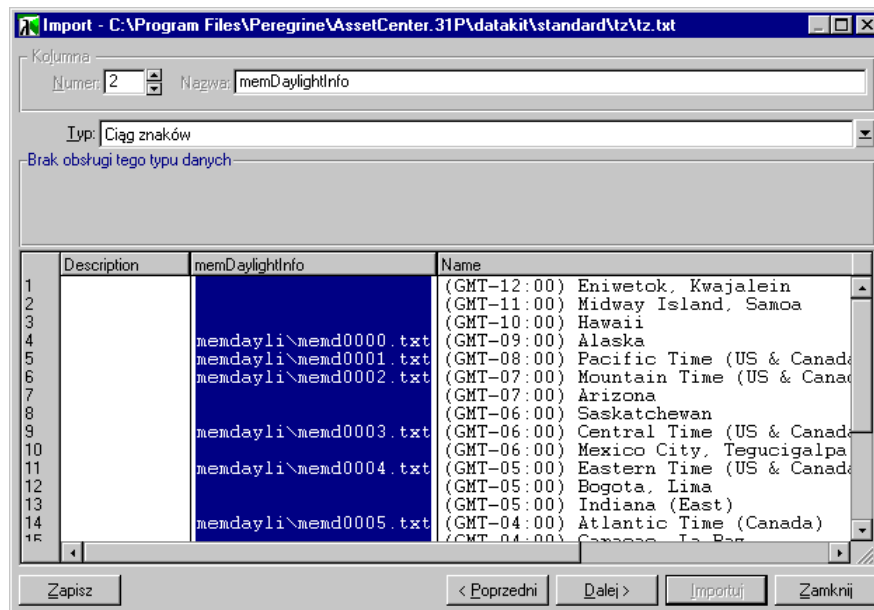
- Numeryczne: Wszystkie znaki muszą być liczbami. Jeśli znalezione zostaną znaki innego typu, wartość tego pola ustawiana jest na „0”.
- Ciąg znakowy: Dopuszczalne są wszystkie znaki za wyjątkiem separatorów.

- Data: Dopuszczalne są tylko formaty daty zdefiniowane w obszarze „Format daty”. Jeśli podczas importu zostaną znalezione inne formaty, wartość pola jest zerowana.

Format daty

Jeśli podasz, że pole zawiera „Datę”, AssetCenter pyta o znak separatora oddzielającego dzień, miesiąc i rok, jak również o kolejność tych trzech elementów.

Oprócz tych dwóch parametrów, w przypadku dat można korzystać z wszystkich dostępnych możliwości przy wprowadzaniu dat w AssetCenter.




Opis pól w źródłowym pliku tekstowym

Uwaga: Tabela wyświetla maksymalnie 25 linii importowanego pliku tekstowego.


Krok 4: Mapowanie pól źródłowych do pól docelowych w bazie danych AssetCenter

Import pojedynczego pliku tekstowego


Po wybraniu pól źródłowych w kroku 3, kliknij przycisk , aby wyświetlić ekran mapowania pól w pliku tekstowym do pól bazy danych AssetCenter.

- ↳ Rozpocznij od zmapowania pliku tekstowego do tabeli w bazie danych AssetCenter za pomocą pola „Tabela docelowa”.
- ↳ Następnie zmapuj pola importowanego pliku tekstowego (wyświetlonego w tabeli **Tabela docelowa**) do pól w bazie danych AssetCenter. (Pola w tabeli docelowej i jej tabelach związanych są wyświetlone na liście po prawej stronie pod polem **Tabela docelowa**.)

Importowanie zestawu plików tekstowych

- ↳ Po wybraniu pól źródłowych w kroku 3, kliknij przycisk , aby wrócić do głównego okna modułu importu.
- ↳ Zmapuj każdy plik tekstowy do tabeli docelowej.
- ↳ Następnie, dla każdej pary (plik tekstowy, tabela docelowa) w tabeli po prawej stronie, zmapuj każde pole tekstowe do tabeli AssetCenter: kliknij podwójnie daną parę lub wybierz ją i użyj polecenia menu **Edycja/ Właściwości**.

Import bazy danych ODBC


- ↳ Po wybraniu pól źródłowych w kroku 3, kliknij przycisk , aby powrócić do głównego okna modułu importu.
- ↳ Zmapuj każdy plik tekstowy do tabeli docelowej.
- ↳ Następnie, dla każdej pary (plik tekstowy, tabela docelowa) w tabeli po prawej stronie, zmapuj pola tekstowe do pól bazy danych AssetCenter: Kliknij podwójnie daną parę, lub wybierz ją i użyj polecenia menu **Edycja/ Właściwości**.


Krok 4: Mapowanie każdego pliku tekstowego lub tabeli źródłowej do tabeli docelowej

Uwaga: Ta sekcja dotyczy importowania zestawu plików tekstowych lub bazy danych ODBC .

Mapowanie każdego pliku tekstowego lub tabeli źródłowej do tabeli docelowej

Kliknij plik lub tabelę źródłową (kolumna „Tabele źródłowe”) i właściwą tabelę docelową (kolumna „Tabele docelowe”). Następnie wykonaj jedną z poniższych czynności:

- Użyj polecenia menu **Edycja/ Mapuj** lub
- Kliknij ikonę , albo
- Użyj polecenia menu **Edycja/ Mapuj** według nazwy: AssetCenter automatycznie mapuje pliki lub tabele o dokładnie tej samej nazwie. Osiągnięte jest to na podstawie technicznej nazwy pola.

Użyj polecenia menu **Edycja/ Usuń mapowanie** lub ikonę , aby usunąć mapowanie pliku lub tabeli źródłowej z tabeli docelowej.

Wyświetlanie struktury bazy danych AssetCenter

Kliknij podwójnie tabelę docelową w kolumnie „Tabele docelowe” lub wybierz ją za pomocą myszy i użyj polecenia menu **Edycja/ Właściwości**. AssetCenter wyświetli listę pól, ich typy i długości.

Krok 4: Mapowanie importowanych pól do pól w bazie danych AssetCenter

Pola źródłowe

W tej części ekranu wyświetlane są nazwy przypisane do kolumn w kroku 3 (w przypadku plików tekstowych) lub krótki opis pola (w przypadku bazy danych AssetCenter.)

Tabela docelowa

Import pojedynczego pliku tekstowego




Wybierz tabelę docelową, w której zostaną umieszczone dane. AssetCenter wyświetli strukturę tabeli (pola w tabeli lub tabelach dołączonych.)

Import zestawu plików tekstowych lub bazy danych ODBC

AssetCenter wyświetla strukturę tabeli docelowej, do której mapowany jest plik lub tabelę źródłową (pola w tabeli lub tabelach związanych).




Mapowanie pól źródłowych do pól docelowych

Jest kilka dostępnych możliwości:

- Za pomocą myszy przeciągnij „pole źródłowe” do „pola docelowego”, aby je zmapować.
- Możesz również wybrać „pole źródłowe”, wybrać „pole docelowe”, a następnie kliknąć ikonę , aby je zmapować.
- Ikona  pozwala również usunąć mapowanie pola źródłowego z pola docelowego po kliknięciu pary (pole źródłowe, pole docelowe).
- Ikona  pozwala automatycznie zmapować pola źródłowe i pola docelowe, które mają dokładnie te same nazwy. Jest to realizowane za pomocą technicznej nazwy pola.



Aby dodać dodatkowe pola wyliczane w pliku źródłowym

AssetCenter pozwala dodać dodatkowe pola do pliku źródłowego. Pola te nie są zapisywane; są one przechowywane w pamięci.

Użyj ikonę ,  i  aby dodawać, usuwać lub wyświetlać te dodatkowe pola.

Wybór kluczy

Możesz wybrać jedno lub więcej pól docelowych, aby utworzyć klucze identyfikacyjne rekordów. Klucz identyfikacyjny pozwala zidentyfikować rekord w tabeli. Jeśli wybierzesz kilka kluczy, wszystkie klucze będą pozwalały identyfikować rekordy.

Wybierz odpowiednie pary (pole źródłowe, pole docelowe) i kliknij przycisk , aby zadeklarować je jako „klucze”. Jeśli przycisk jest aktywny, przypomina on wciśnięty przycisk i jest jaśniejszy; a mała ikona po lewej stronie pola docelowego lub łącza wygląda również jak .




AssetCenter importuje linie z pliku źródłowego pojedynczo, w następujący sposób:

- ↪ Jeśli istnieje rekord w bazie danych, którego klucze mają dokładnie te same wartości, AssetCenter modyfikuje rekord zgodnie z danymi zawartymi w pliku tekstowym.
- ↪ Jeśli istnieje kilka rekordów z tym samym zestawem kluczy, program zatrzymuje się na pierwszym znalezionym rekordzie i ignoruje pozostałe. Dlatego należy odpowiednio wybrać właściwe klucze.
- ↪ Jeśli nie istnieją rekordy pasujące do kluczy, AssetCenter tworzy nowy rekord w bazie danych.

Uwaga: Nie zalecamy stosowania pól identyfikacyjnych tabel jako kluczy uzgodnienia przy ponownym importowaniu wyeksportowanych danych, ponieważ nie są one stałe i mogą ulec zmianie. Należy stosować klucze, których wartości są „niezmiennie”, jak na przykład **KodWewn** w przypadku zasobów.

Konfiguracja tworzenia rekordów związanych


Przy importowaniu danych z pliku do kilku różnych tabel (na przykład, plik zawierający pracowników i ich lokalizacje), wybierz główną tabelę docelową (w naszym przykładzie tabelę pracowników) i wskaż za pomocą łączy, gdzie powinny zostać zaimportowane dane do innych tabel (w naszym przykładzie tabela lokalizacji.)

AssetCenter pozwala skonfigurować sposób tworzenia rekordów w tabeli dołączonej, jeśli przed importem rekord nie istniał. Użyj ikony , aby wykonać tę konfigurację. Ikony tej można używać tylko w odniesieniu do łącza (a nie pól tabel związanych). Łącza oznaczane są ikonami  i .

Aby wyświetlić ekran konfiguracji:

↳ Zmapuj importowane pola do pól w tabeli związanej.

↳ Kliknij właściwe łącze.

↳ Kliknij ikonę .

Jeśli rekord zostanie znaleziony za pomocą podanych kluczy identyfikacyjnych, program modyfikuje dane w rekordzie, jeśli są one zgodne.

Utwórz rekord

Rekord jest tworzony, jeśli nie znaleziono go w bazie danych za pomocą podanych kluczy identyfikacyjnych.


Nie twórz rekordu

Rekordy nie są tworzone, nawet jeśli nie zostaną znalezione w bazie danych za pomocą podanych kluczy identyfikacyjnych.

Sygnalizuj błąd – sytuacja anormalna

AssetCenter generuje komunikat o błędzie, jeśli nie znaleziono rekordu w bazie danych za pomocą podanych kluczy identyfikacyjnych.

Uwzględniaj tylko już dołączone rekordy (-)

Moduł importu uwzględni tylko te rekordy, które już zostały dołączone do głównego rekordu, jeśli dołączysz ikonę  (pinezka) do łącza.

Przykład: Importujesz listę pracowników i zasobów, z których korzystają. Tabela wydziałów i pracowników jest tabelą docelową. Dołączasz pinezkę do łącza z tabelą zasobów. Dla każdego zasobu związanego z pracownikiem w pliku źródłowym, moduł importu uwzględni tylko te zasoby, które są już używane (zakładka **Wykorzystywane zasoby** w szczegółach pracownika).

W przypadku łącza typu „Własne”, pinezka umieszczana jest automatycznie i nie można jej usunąć. Łącza typu „Własne” są to łącza, dla których rekordy związane są automatycznie usuwane przy usunięciu głównego rekordu. Do tego typu należy na przykład łącze pracownicy-szkolenie: jeśli zostanie usunięty pracownik, zostaną również usunięte wszystkie pozycje szkolenia związane z danym pracownikiem.

Wynik zastosowania pinezki zależy od typu łącza:

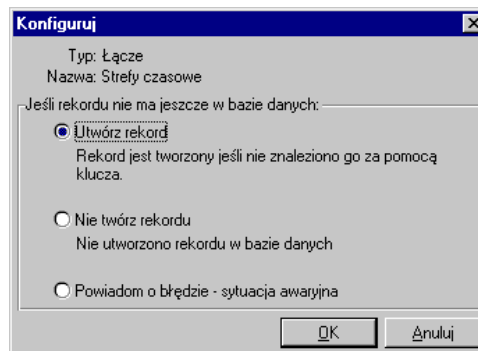
- Jeśli tabela docelowa jest tabelą zasobów, a pinezkę dołączasz do łącza „użytkownik”, moduł importu szuka jedynie użytkowników związanych z zasobami. Ponieważ dla danego zasobu jest tylko jeden użytkownik, można w ten sposób zmodyfikować lub utworzyć użytkownika zasobu bez identyfikacji użytkownika za pomocą klucza. Jest to również bardzo przydatne przy modyfikacji wartości cechy danego zasobu.
- Jeśli tabela docelowa jest tabelą wydziałów i pracowników, a pinezkę dołączasz do łącza „zasoby”, moduł importu szuka tylko zasobów związanych z pracownikiem. W tym przypadku, aby zmodyfikować lub utworzyć zasoby użytkownika, musisz mieć odpowiednie klucze identyfikacyjne, jednak nie muszą one być tak charakterystyczne, jak w przypadku bez używania pinezki.

Uwaga: W AssetCenter stosowane są trzy rodzaje łączy pomiędzy rekordami:

łącza n: Na przykład, zasób może być związany tylko z jedną lokalizacją, jednak lokalizacja może być związana z kilkoma zasobami.


łącza 1: Na przykład, zasób może być związany z jednym komentarzem, a komentarz może być związany z jednym zasobem.

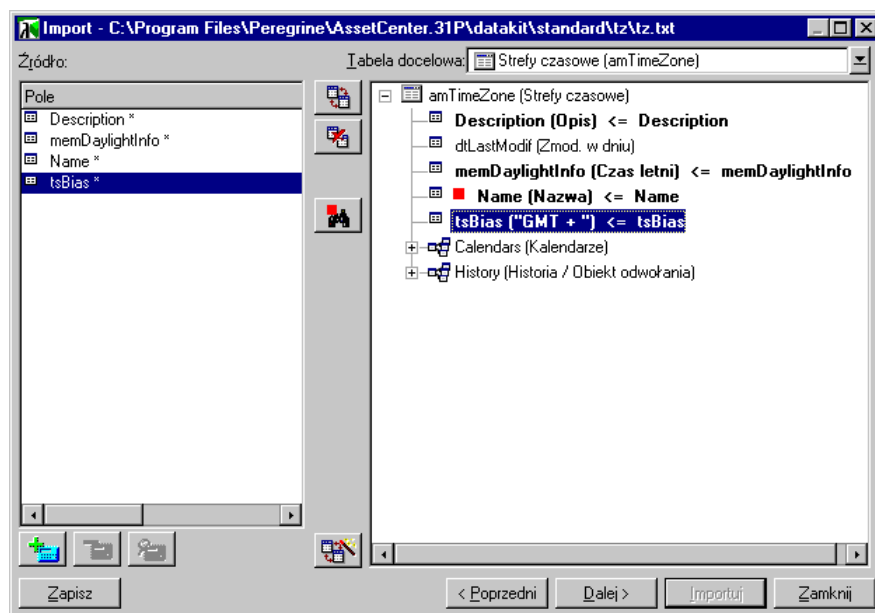
łącza n-n: Na przykład, dostawca może być związany z kilkoma produktami; a produkt może być związany z kilkoma dostawcami.



Konfiguracja tworzenia rekordów w tabelach związanych

Symbole używane w strukturze drzewa tabel docelowych

- ☐ Wskazuje, że podana nazwa jest tabelą (duży symbol) lub polem (mały symbol).
- ☐☐ Wskazuje, że podana nazwa jest tabelą związaną z tabelą zwierzchnią. Możesz wybrać tylko jeden rekord w tabeli związanej ze szczegółów rekordu tabeli zwierzchniej. Ten typ łącza przedstawia pole, którego wartość można wprowadzić za pomocą „okna wyboru” lub „listy rozwijanej”.
- ☐☒ Wskazuje, że podana nazwa jest tabelą związaną z obiektem zwierzchnim. Możesz wybrać kilka rekordów w tabeli związanej ze szczegółów rekordu tabeli zwierzchniej. Ten typ łącza przedstawia listę rekordów znajdującą się w zakładce okna szczegółów tabeli zwierzchniej.
- ☐☒ Ten symbol to „pinezka”. Można ją dołączać do łącza z tabelą zawierającą pole, do którego importowane pole jest zmapowane. Po ustawieniu pinezki, moduł importu przeszukuje tylko rekordy dołączone do rekordu, do którego jest ona „przypięta”. Obecność pinezki jest określana przez opcje wyświetlane za pomocą ikony .



Mapowanie pól źródłowych do pól tabeli docelowej

Krok 4: Dodawanie dodatkowych pól wyliczanych w pliku źródłowym

AssetCenter pozwala wstawiać dodatkowe pola do pliku źródłowego. Pola te nie są zapisywane; są one przechowywane w pamięci.

Dodatkowe pola można dodawać, usuwać i wyświetlać za pomocą ikon



Nazwa

Podaj nazwę nowego pola.

Pole typu wzór

Wpisz sposób wyliczania nowego pola. Aspekt ekranu zmienia się zgodnie z wybranym typem.

Łączenie

Ten tryb pozwala łączyć kilka pól w pliku źródłowym. Wybierz kolejno pola. Można je oddzielić dowolnym znakiem. Znaki należy ująć w cudzysłów.

Przykład: Pole1" i "Pole2.

Pobieranie określonej liczby znaków

Tryb ten pozwala wydzielić część pola tekstowego:

- ❖ Wybierz pole źródłowe (zwane „Polem głównym”).
- ❖ Wypełnij pole **Liczba znaków do zignorowania**. AssetCenter pominie te znaki.
- ❖ Wypełnij pole **Liczba znaków do wykorzystania**: AssetCenter zachowa te znaki po pominięciu **Znaków do zignorowania**.
- ❖ Pole **Rozpocznij pobieranie od końca pola** informuje AssetCenter, że **Liczbę znaków do zignorowania** należy liczyć od końca pola i zachować „Liczbę znaków do wykorzystania” rozpoczynając po pominiętych znakach i licząc wstecz.

Przykład:

- ❖ „Liczba znaków do zignorowania”: 3
- ❖ „Liczba znaków do wykorzystania”: 5
- ❖ Wartość pola w pliku źródłowym: „REFIMP05A18500”
- ❖ Wartość zaimportowana do bazy danych: „IMP05”, jeśli nie zaznaczono pola **Rozpocznij pobieranie od końca pola**, lub „05A18”, jeśli je zaznaczono.

Pobieranie według separatorów

Ten tryb pozwala uzyskać część pola z pliku źródłowego:

- ❖ Wybierz pole źródłowe (zwane „Polem głównym”).
- ❖ Podaj **Separator** używane w wartościach głównego pola.
- ❖ Wypełnij pole **Liczba separatorów do zignorowania**. AssetCenter zachowuje wszystkie dane następujące po tych separatorach.
- ❖ Wypełnij pole **Liczba separatorów do uwzględnienia**. AssetCenter zachowuje wszystkie dane między początkiem odczytywanego tekstu i separatorem następującym po ostatnim separatorze, który ma być odczytany.
- ❖ Zaznacz pole wyboru **Rozpocznij pobieranie od końca pola**, jeśli chcesz, aby AssetCenter liczył **Liczbę separatorów do zignorowania** i **Liczbę separatorów do uwzględnienia** od końca.

Przykład:

- ❖ **Separator:** /
- ❖ **Liczba znaków do zignorowania:** 2
- ❖ **Liczba separatorów do uwzględnienia:** 3
- ❖ Wartość pola w pliku źródłowym: „1/2/3/4/5/6/7/8/9”
- ❖ Wartość importowana do bazy danych: „3/4/5/6”, jeśli nie zaznaczono pola **Rozpocznij pobieranie od końca pola** i „4/5/6/7”, jeśli je zaznaczono.

Wartość ustalona

Ten tryb pozwala uwzględnić połączenie:

- ❖ Ciągów znakowych objętych znakiem".
- ❖ Zmiennych. Są to pewne zmienne wynikające z funkcji używanych w domyślnych wartościach pól, takich jak AmLoginName(), AmDate(), AmCounter().

Tryb drzewa

Ten tryb pozwala utworzyć strukturę drzewa z pojedynczego pola w pliku źródłowym.

- ↳ Wybierz pole źródłowe (zwane „Polem głównym”).
- ↳ Podaj „Separatory” używane pomiędzy wartościami podrzędnymi pola.

AssetCenter dzieli pole źródłowe na wartości podrzędne. Liczba wartości podrzędnych jest równa liczbie ciągów znakowych oddzielonych znakami separatorów. Następnie AssetCenter tworzy rekord dla każdej wartości podrzędnej i porządkuje je hierarchicznie.

Przykład:

- ↳ Utwórz plik tekstowy zawierający kolumnę o nazwie „Nazwa”. Wartość jednej linii w pliku jest następująca „/Oddział we Francji/Główne biuro sprzedaży/Wydział marketingu”.
- ↳ Skonfiguruj moduł importu tworząc pole ze wzorem typu „Tryb drzewa” (separatorem jest znak „/”). Pole to nosi nazwę „PoleWzoru”. Utwórz pole ze wzorem typu „WartośćStała” (wartość = „1”) i zmapuj je do pola **Wydział** (nazwa SQL: bDepartment) (aby utworzyć wydziały, ale nie pracowników).
- ↳ Zmapuj „PoleWzoru” do pola **Nazwisko** (nazwa SQL: Name) w tabeli pracowników.
- ↳ Uruchom import pliku
- ↳ Wynik: Utworzone zostały 3 wydziały połączone hierarchicznie: „Oddział we Francji”, „Główne biuro sprzedaży” i „Wydział marketingu”.

Plik

Ten tryb pozwala zaimportować plik do bazy danych. Jest to użyteczne przy importowaniu obrazów lub długich fragmentów tekstu.

Pliki mogą być importowane do następujących typów pól:

- ❖ Memo,
- ❖ Blob.


Obsługiwane są następujące formaty plików:

- ❖ Tekst ANSI,
- ❖ Obrazy (można importować wszystkie formaty grafiki obsługiwane przez AssetCenter).

We wzorze wyliczenia pola wskaż plik źródłowy podając pełną ścieżkę importowanego pliku (ścieżkę, nazwę i rozszerzenie). Domyślnie jako ścieżka stosowany jest bieżący folder.

Skrypt

Ten tryb pozwala wyliczyć wartość za pomocą skryptu BASIC. Skrypt może odwoływać się do importowanych pól źródłowych.

Aby utworzyć skrypt obliczeniowy, wpisz kod bezpośrednio lub kliknij przycisk  aby skorzystać z kreatora wyrażeń.

Skrypt nie może odwoływać się do pól w bazie danych.

Test

Uwaga: To pole jest wyświetlane wyłącznie, jeśli typ wzoru pola jest ustawiony jako „Stała długość” lub „Pobieranie pól rozgraniczonych”.

Wpisz wybraną wartość pola.

Wynik

Uwaga: To pole jest wyświetlane wyłącznie, jeśli typ wzoru pola jest ustawiony jako „Stała długość” lub „Pobieranie pól rozgraniczonych”.

W tym polu wyświetlana jest symulowana wartość importu danych próbnych

Krok 4: Przypadki szczególne

Importowanie wydziałów i pracowników

Przy importowaniu rekordów z tabeli wydziałów i pracowników, niekiedy trzeba określić, czy importowany rekord odnosi się do wydziału, czy do pracownika.

Istnieje pole, które to umożliwia: jest to pole **Wydział** (nazwa SQL: bDepartment). Jego wartość wynosi „1” dla wydziałów i „0” dla pracowników. Domyślnie moduł importu przyjmuje jego wartość jako „0”.

Zalecamy utworzenie pola ze wzorem typu „WartośćStała” przyjmującego wartość „1” i dołączenie tego wzoru do pola **Wydział**, kiedy importowany element jest wydziałem.

Uwaga: Moduł importu wykrywa, że rekord dotyczy wydziału, gdy rekord ten ma rekord podległy w tabeli wydziałów i pracowników; pracownicy nie mogą mieć rekordów podległych.

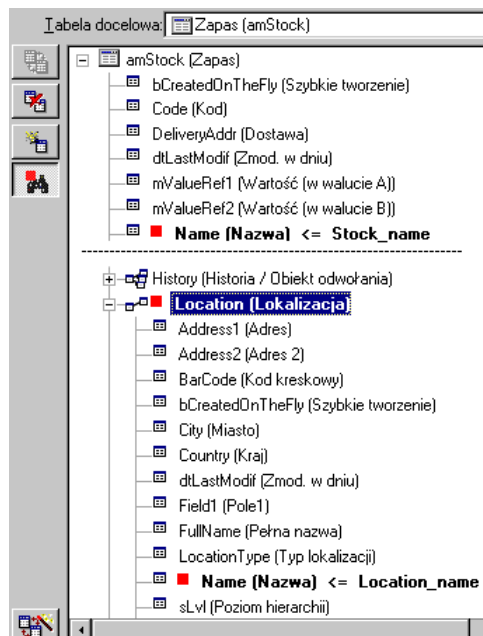
Importowanie dokumentów

Przy importowaniu dokumentów należy określić pole importowane w polu **Tabela** (nazwa SQL: DocObjTable) w tabeli dokumentów. Pole **Tabela** (nazwa SQL: DocObjTable) określa nazwę SQL tabeli, z którą związany jest dokument.

Krok 4: Przykłady używania kluczy

Oto jak AssetCenter interpretuje wybrane klucze:

Przykład 1: Używanie pola związanego jako klucza głównej tabeli



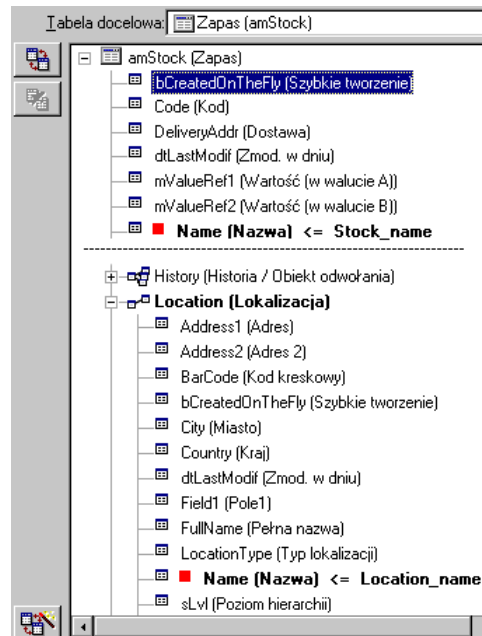
W tym przykładzie zapas jest identyfikowany za pomocą dwóch głównych kluczy:

- Nazwa.Zapasu: Ponieważ klucz „**Name (Nazwa) <= Stock_name**” jest zadeklarowany jako klucz identyfikacyjny i jest częścią głównej tabeli.
- Nazwa.Lokalizacji: Ponieważ pole związane „**Name (Nazwa) <= Location_name**” jest zadeklarowane jako klucz identyfikacyjny w tabeli lokalizacji, a łącze **Location (Lokalizacja)** jest zadeklarowane jako klucz identyfikacyjny.

W tym samym przykładzie, lokalizacja jest identyfikowana za pomocą głównego klucza:

- Nazwa.Lokalizacji: Ponieważ pole „**Name (Nazwa) <= Location_name**” jest zadeklarowane jako klucz identyfikacyjny w tabeli Lokalizacji.

Przykład 2: Zdefiniowanie pola jako klucza w tabeli związanej, nie będącego kluczem w głównej tabeli



W tym przykładzie zapas jest identyfikowany za pomocą pojedynczego klucza:

- Nazwa.Lokalizacji: Ponieważ pole „**Name (Nazwa) <= Stock_name**” jest zadeklarowane jako klucz i jest częścią głównej tabeli,
- A łącze **Location (Lokalizacja)** nie jest kluczem identyfikacyjnym.

W tym samym przykładzie, lokalizacja jest identyfikowana za pomocą klucza:


- Nazwa.Lokalizacji: Ponieważ klucz „**Name (Nazwa) <= Location_name**” jest zadeklarowany jako klucz główny.

Wniosek

- Można definiować klucze dla głównej tabeli i inne niezależne klucze dla tabel związanych z główną tabelą. Można zatem importować dane do kilku tabel z pojedynczego pliku tekstowego.

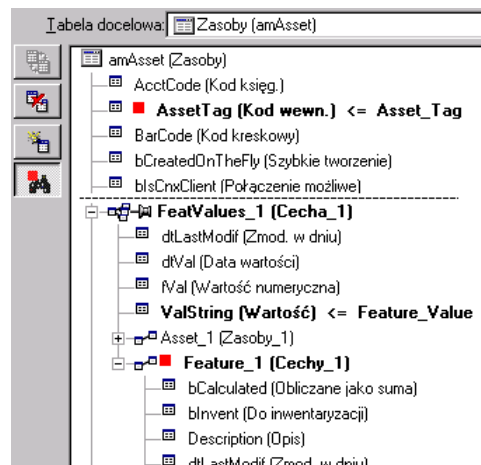
- Aby zadeklarować pole w tabeli dołączonej jako jeden z kluczy w głównej tabeli, trzeba zadeklarować pole w tabeli związanej ORAZ łącze jako klucze identyfikacyjne. Jeśli zostanie zaznaczone tylko łącze, klucz zostanie użyty tylko w tabeli związanej.


Przykład 3: Klucze aktualizujące wartości cech w rekordach bazy danych

Aby zaktualizować wartość cechy rekordu do określonej wartości, trzeba znaleźć parę (rekord, cecha) w bazie danych i przypisać jej nową wartość. Jeśli ta para nie istnieje, AssetCenter odwołuje się do opcji łącza zdefiniowanych za pomocą ikony , aby określić, czy należy utworzyć rekord związany.

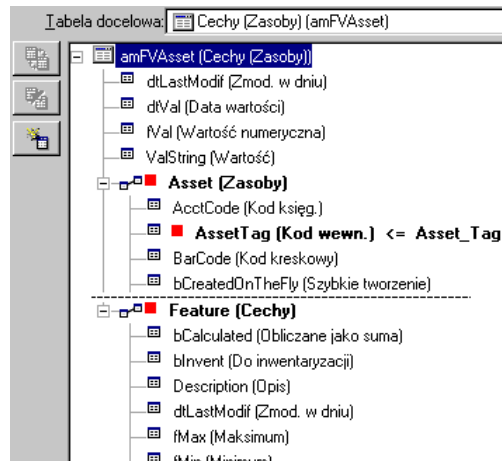
Można użyć dowolnej z poniższych dwóch metod:

Metoda pierwsza (przykład z tabeli zasobów):



- ❖ Klucz w **AssetTag (Kod wewn.)** określa zasoby do modyfikacji.
- ❖ Ikona  na łączu **FeatValues (Cecha)** określa, że chcemy znaleźć tylko cechy danego zasobu.
- ❖ Klucz w **Feature (Cechy)** wskazuje, że para (cecha, wartość) jest oznaczona za pomocą cechy.
- ❖ Klucz w **Name (Nazwa) <= Feature_Name** wskazuje, że cecha jest oznaczona za pomocą nazwy.
- ❖ Nowa wartość pojawia się w polu „**ValString (Wartość) <= Feature_Value**”.

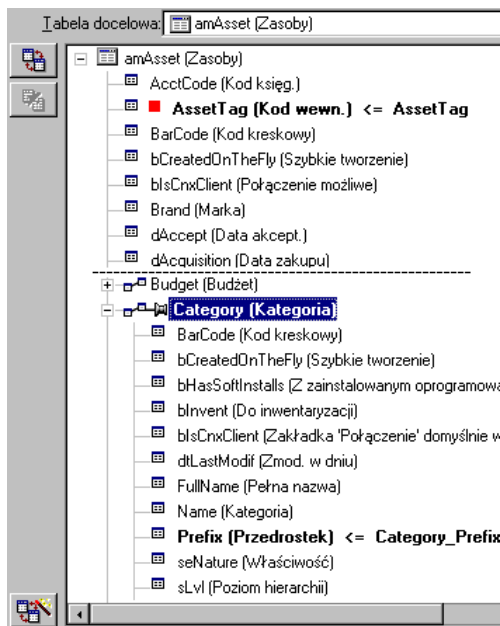
Metoda druga (przykład z tabeli wartości cech zasobów):




- ❖ Para (**Feature (Cechy)** , **Asset_1 (Zasoby_1)**) określa cechę związaną z zasobem.
- ❖ Klucz w „ **Name (Nazwa)** ” określa kluczowe pole łącza „ **Feature (Cechy)** ”.
- ❖ Klucz w „ **AssetTag (Kod wewn.)** ” określa kluczowe pole łącza „ **Asset_1 (Zasoby_1)** ”.
- ❖ Nowa wartość pojawia się w polu „ **ValString (Wartość)** <= **Feature_Value** ”.

Przykład 4: Modyfikacja zawartości rekordu związanego, dla którego nie ma klucza identyfikacyjnego

Przykład: Chcesz zmodyfikować przedrostek kategorii, która jest związana z danym zasobem. W importowanym pliku nie ma klucza identyfikującego kategorię. Wiadomo tylko, że kategoria jest związana z danym zasobem.




- Klucz w **AssetTag (Kod wewn.) <= AssetTag** określa zasób.
- Symbol  na **Category (Kategoria)** określa, że chcemy znaleźć tylko kategorie już związane z zasobem.
- **Prefix (Przedrostek) <= Category_Prefix** przypisuje nową wartość do pola **Przedrostek** (nazwa SQL: Prefix) kategorii.

Efektywność: Dla optymalnej efektywności, zalecamy wybranie kluczy spośród tych, które stanowią indeksy tabeli (Ostrzeżenie: Niektóre indeksy składają się z kilku pól.)

Krok 5: Konfiguracja transferu

Import pojedynczego pliku tekstowego

Po przypisaniu pól źródłowych do pól w bazie danych w kroku 4, kliknij przycisk , aby wyświetlić okno ustawień transferu.

Import zestawu plików tekstowych lub bazy danych ODBC

Użyj polecenia menu **Edycja/ Opcje** na głównym ekranie modułu importu.

Ramka „Obsługa błędów”

Wybierz preferowaną metodę obsługi błędów:

Wstrzymaj import w razie wystąpienia błędu

Proces importu zostaje przerwany natychmiast po napotkaniu błędu.

Wykonuj każdą zaimportowaną linię

Moduł importu zatwierdza każdą importowaną linię. Jeśli w linii zostaną stwierdzone jakieś błędy (w głównej tabeli, lub w tabeli związanej), cała linia nie zostanie zaimportowana. Program przywraca stan bazy danych sprzed importu linii.

Wykonuj według grupy linii

Moduł importu przetwarza import grupami linii; podaje się ilość linii w grupie. Jeśli w grupie zostaną stwierdzone jakieś błędy (w głównej tabeli, tabeli związanej lub dowolnej linii z grupy), nie zostanie zaimportowana żadna linia z całej grupy. Program przywraca stan bazy danych sprzed importu grupy. Gwarantuje to integralność bazy danych.

Ramka „Plik dziennika”

Wybierz operacje, które powinny być rejestrowane w pliku dziennika:

- Błędy
- Dołączenia i uaktualnienia

Wpisz nazwę i ścieżkę pliku dziennika. Jeśli plik nie istnieje, AssetCenter utworzy go. Dodaj dowolne rozszerzenie. Zalecamy stosowanie rozszerzenia **.log**.


Ostrzeżenie: Moduł importu nie może tworzyć katalogów.

Plik dziennika zawiera również następujące informacje:

- Czas uruchomienia zadania.
- Opis zadania.
- Napotkane błędy.

Plik dziennika jest uaktualniany dla każdej nowej procedury importu.

Krok 6: Transfer danych

W każdym z poprzednich kroków, można kliknąć przycisk , aby rozpocząć transfer danych, jeśli dostępna jest wystarczająca ilość informacji.

Zachowanie na poziomie rekordu

- AssetCenter importuje dane linia po linii w kolejności, w jakiej dane występują w pliku źródłowym.
- AssetCenter może zaimportować dane do kilku różnych tabel z pojedynczej linii.
- Jeśli nie można zaimportować części linii, AssetCenter tworzy to, co może utworzyć.
- AssetCenter przeszukuje bazę danych AssetCenter w poszukiwaniu rekordu, którego klucze identyfikacyjne mają dokładnie takie same wartości jak rekord źródłowy. Jeśli taki rekord zostanie znaleziony, AssetCenter modyfikuje go zgodnie z danymi zawartymi w pliku źródłowym.
- Jeśli nie znaleziono rekordu odpowiadającego kluczom identyfikacyjnym, AssetCenter tworzy nowy rekord w bazie danych.
- Jeśli nie ma zdefiniowanych kluczy identyfikacyjnych, moduł importu dodaje importowane rekordy, jeśli importowane wartości są zgodne z unikalnymi właściwościami podstawowych pól. Bez kluczy, moduł importu nie może aktualizować rekordów.

- Moduł importu działa w ten sam sposób w odniesieniu do elementów głównych i elementów związanych.

Zachowanie na poziomie pola

- Moduł importu nie sprawdza, czy pole jest obowiązkowe. Należy sprawdzić, czy pola obowiązkowe rzeczywiście znajdują się w importowanych danych.
- Jeśli moduł importu znajdzie nieznaną wartość znajdującą się na liście pozycji „systemu”, importowana linia jest odrzucana.
- Jeśli moduł importu znajdzie wartość, której nie ma na standardowej liście pozycji, importowana linia jest akceptowana, a nowa wartość jest dodawana do listy pozycji, jeśli jest to lista „otwarta”. Jeśli lista pozycji jest „zamknięta”, importowana linia jest odrzucana.
- Jeśli wartość pola przekracza maksymalną długość, koniec wartości zostanie obcięty.
- Pole o pustej wartości w pliku tekstowym wymazuje istniejącą wartość w odpowiadającym mu polu bazy danych.
- Gdy import linii danych powoduje utworzenie nowego rekordu, AssetCenter wstawia domyślną wartość w polach, które nie mają odpowiadających im kolumn w pliku tekstowym lub źródłowej tabeli.
Jeśli kolumna jest obecna, ale nie ma określonej wartości, AssetCenter wstawia puste pole zamiast domyślnej wartości.

Import bazy danych AssetCenter

Użyj zakładki AssetCenter w opcji Importuj bazę danych w menu Plik/ Import, aby zaimportować bazę danych AssetCenter.

Zapisywanie i wykonywanie skryptu importu

Skrypt jest grupą ustawień importu zapisanych pod podaną nazwą. Skrypt importu pozwala odtworzyć podobne warunki importu bez ponownego definiowania wszystkich parametrów. Pozwala to zaoszczędzić czas.


Skrypty są użyteczne do:

- ❖ Wykonywania tego samego importu kilka razy, aż będzie działał w oczekiwany sposób. (Przy każdym uruchomieniu można modyfikować pole źródłowe.)
- ❖ Wykonywania regularnych aktualizacji bazy danych (można aktualizować tabelę pracowników w oparciu o plik z Działu Kard.)

AssetCenter pozwala zapisywać skrypty i uruchamiać je później.

Aby zapisać skrypt

Import pojedynczego pliku tekstowego


- ❖ Użyj polecenia menu **Plik/ Import**, aby wybrać plik tekstowy.
- ❖ Zdefiniuj ustawienia importu (nazwa i lokalizacja pliku danych, jego struktura, itp.)
- ❖ Kliknięcie przycisku  zapisuje ustawienia jako skrypt.

Import bazy danych lub zestawu plików tekstowych

- ❖ Użyj polecenia menu **Plik/ Import**, aby wybrać bazę danych.
- ❖ Zdefiniuj ustawienia importu (lokalizacja bazy danych lub plików tekstowych, mapowania pomiędzy polami, itp.).
- ❖ Ustawienia te można zapisać w dowolnej chwili jako skrypt za pomocą polecenia menu **Plik/ Zapisz** lub **Plik/ Zapisz jako**.

Aby zmodyfikować skrypt

Import pojedynczego pliku tekstowego


- ❖ Użyj polecenia menu **Plik/ Import**.
- ❖ Wskaż import pliku tekstowego.
- ❖ W oknie dialogowym „Otwórz plik danych”, wybierz pliki typu „Skrypty importu *.scr” w polu „lista typów plików”.
- ❖ Otwórz skrypt.
- ❖ Zmodyfikuj ustawienia importu (nazwę i lokalizację pliku danych, jego strukturę, itp.)
- ❖ Kliknięcie przycisku  zapisuje nowe ustawienia.

Import bazy danych lub zestawu plików tekstowych

- ❖ Użyj polecenia menu **Plik/ Import**, aby zaimportować bazę danych.
- ❖ Wypełnij zakładkę **ODBC** lub zakładkę **Tekst**. Kliknij opcję **Otwórz**.
- ❖ Następnie otwórz skrypt za pomocą polecenia menu **Plik/ Otwórz skrypt**.
- ❖ Zmodyfikuj ustawienia importu (nazwę i lokalizację bazy danych, jej strukturę, itp.)
- ❖ Ustawienia te można zapisać w dowolnej chwili jako skrypt za pomocą polecenia menu **Plik/ Zapisz** lub **Plik/ Zapisz jako**.

Aby utworzyć nowy skrypt przy definiowaniu ustawień importu

Import pojedynczego pliku tekstowego


Kliknij przycisk . Postępuj tak jak przy tworzeniu nowego skryptu.

Import bazy danych lub zestawu plików tekstowych

Użyj polecenia menu **Plik/ Nowy skrypt**. AssetCenter zaproponuje zapisanie bieżących ustawień jako skrypt przed ich odrzuceniem.

Aby wykonać skrypt

Aby wykonać istniejący skrypt:

- Użyj polecenia menu **Plik/ Import**.
- Kliknij ikonę .
- Podaj nazwę pliku skryptu.
- Rozpocznij procedurę importu.

Można również postąpić tak, jak przy modyfikacji skryptu, a następnie uruchomić procedurę importu z okien dialogowych importu.

Uruchamianie skryptu importu z wiersza poleceń systemu 32-bitowego

Ostrzeżenie: Ta operacja jest możliwa jedynie z wiersza poleceń (okno DOS) w 32-bitowej wersji systemu Windows (na przykład Windows NT, 95 lub 98)

Jak to działa

Aby uruchomić program DOS w trybie „on line”, trzeba mieć gotowy skrypt utworzony wcześniej za pomocą modułu importu w środowisku Windows.

Można uruchomić go ręcznie lub automatycznie (na przykład za pomocą pliku wsadowego). Jest to program o nazwie **AmImpL32.exe** (znajdujący się w folderze **Bin32** foldera programu AssetCenter.) Program ten działa wyłącznie w środowiskach 32-bitowych.

Składnia

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

-verbose: wyświetla komunikaty podczas importu. Opcja ta jest włączona domyślnie.

-.?, -h lub -H: wyświetla komunikaty pomocy.

-src: odpowiednio do sytuacji, parametr ten wyświetla:

- ❖ Ścieżkę i nazwę wykonywanego skryptu importu,
- ❖ Nazwę połączenia z bazą danych AssetCenter dla całkowitego importu (podana w poleceniu menu **Plik/ Zarządzaj połączeniami**, pole **Nazwa**.)
- ❖ Nazwę bazy danych AssetCenter bez połączenia (dla której nie zdefiniowano połączenia w pliku .ini:
[<EngineName>;<DatabaseLocation>;<User>;<Password>]
Dla tego przypadku, w poniższej tabeli podano, w jaki sposób wypełnić poszczególne pola wymienione powyżej pomiędzy znakami <>:

	Oracle	MS SQL Server	Sybase SQL Anywhere	Sybase SQL Server
EngineName	Oracle	ODBC	ODBC	Sybase
DatabaseLocation	Nazwa serwera	Nazwa źródła danych	Nazwa źródła danych	Nazwa serwera:Nazwa bazy danych
User	Nazwa konta	Nazwa użytkownika MS SQL Server	Nazwa użytkownika Sybase SQL Anywhere	Nazwa konta
Password	Hasło konta	Hasło użytkownika MS SQL Server	Hasło użytkownika Sybase SQL Anywhere	Hasło konta

-srcpass: hasło związane ze źródłową importowaną bazą danych. W przypadku bazy danych AssetCenter, jest to hasło konta „Admin”.

-dst: nazwa połączenia bazy danych AssetCenter, do której importowane są dane (podana w poleceniu menu **Plik/ Zarządzaj połączeniami**, pole **Nazwa**.)

-dstlogin: nazwa logowania konta AssetCenter, które otrzyma importowane dane (konto „Admin” lub użytkownik AssetCenter z uprawnieniami administratora).

-dstpass: hasło związane z opcją „dstlogin”.

-log: pełna ścieżka pliku dziennika importu.

Uwaga: Jeśli ciąg znaków w nawiasach ostrych <> zawiera spacje, należy ująć je w cudzysłów (,).

Przykład:

```
amimpl32 -verbose -src:employee.scr -srcpass:PassWord -dst:MainDBase -  
dstlogin:Gerald -dstpass:PassWord -log:"My Log File.txt"
```

Rozdział 16 - Eksport danych i zarządzanie widokami SQL

W niniejszym rozdziale wyjaśniono, w jaki sposób eksportować dane z AssetCenter i zarządzać widokami SQL bazy danych.

Rozdział zawiera informacje na następujące tematy:

- Definicje
- Eksport danych z bazy danych AssetCenter
- Zarządzanie widokami SQL w bazie danych AssetCenter
- Zalecenia
- Definiowanie skryptu eksportu
- Wykonywanie skryptu eksportu

Definicje

Skrypty eksportu

Skrypty eksportu pozwalają eksportować dane, tworzyć, odtwarzać lub usuwać widoki SQL za pomocą opcji AssetCenter Export lub programu **amexpl32.exe**. Skrypty eksportu można zapisywać do późniejszego wykorzystania.

Skrypt eksportu działa:

- w „Trybie eksportu” dla eksportu danych.

- lub w „Trybie widoku”, aby tworzyć, odtwarzać lub usuwać widoki SQL z bazy danych.

Zawiera on:

- Zapytania eksportowe, które definiują, co powinno być pobrane z bazy danych.
- Opcje formatowania, dla eksportu danych.
- Działania do wykonania, dotyczące tworzenia/usuwania widoków SQL.

Zapytania eksportowe

Zapytania eksportowe definiuje się za pomocą modułu AssetCenter Export.

Zapytanie eksportowe jest definiowane przez:

- Nazwę.
- Wynikowy plik eksportu (w „Trybie eksportu”).
- Komentarz (który nie jest eksportowany).
- Tabelę początkową.
- Listę pobieranych kolumn (pola, łącza, cechy i pola wyliczane z tabeli początkowej) i związanych z nią kryteriów sortowania.
- Filtr zawierający warunek WHERE, definiujący warunki pobierania.
- Filtr zawierający warunek HAVING definiujący warunki pobierania.
- Tekst zapytania (zgodny z **Filtrem (warunek WHERE)** i warunek „HAVING”).
- Zakładkę podglądu.

Eksport danych z bazy danych AssetCenter

Dane z bazy danych AssetCenter można eksportować do plików tekstowych:

- Za pomocą skryptu eksportu.
- Za pomocą polecenia menu **Plik/ Eksportuj listę**. To polecenie menu jest wyświetlane, gdy wyświetlana jest przynajmniej jedna lista lub lista zakładek. Umożliwia ono eksport aktywnej listy.


Eksport danych za pomocą skryptu eksportu

- ↳ Uruchom AssetCenter Export za pomocą menu **Start** lub z grupy programu AssetCenter.
- ↳ Zdefiniuj skrypt eksportu z trybem ustawionym na „Tryb eksportu”:
 - ❖ W zakładce **Zapytania** wpisz zapytania definiujące eksportowane dane.
 - ❖ W zakładce **Formatowanie** określ format plików tekstowych, do których zostaną wyeksportowane dane.
 - ❖ Użyj polecenia menu **Plik/ Zapisz skrypt** lub **Plik/ Zapisz skrypt jako**, aby zapisać skrypt.
- ↳ Wykonaj skrypt eksportu.
 - ❖ Bezpośrednio w module AssetCenter Export lub
 - ❖ Uruchamiając program **amexpl32.exe** z wiersza poleceń.
Uwaga: Ten program jest dostępny tylko dla środowisk 32-bitowych.

Uwaga: Aby zachować integralność ograniczeń dostępu zdefiniowanych w AssetCenter, tylko administratorzy mogą uruchamiać moduł AssetCenter Export lub program **amexpl32.exe** (nazwa logowania „Admin” lub użytkownik z uprawnieniami administratora)

Eksport danych za pomocą polecenia menu **Plik/ Eksportuj listę**

Polecenie menu **Plik/ Eksportuj listę** jest dostępne dla wszystkich użytkowników AssetCenter; pozwala ono użytkownikom eksportować dane, które mają prawo przeglądać.

- ↳ Wyświetl listę, którą chcesz wyeksportować (listę główną lub listę zakładek). Jeśli wyświetlonych jest kilka list, upewnij się, że eksportujesz właściwą listę.
- ↳ Wybierz polecenie menu **Plik/ Eksportuj listę**.
- ↳ Wypełnij kolejne okno, a następnie kliknij przycisk .

Dalsze informacje o poleceniu menu **Plik/ Eksportuj listę** znajdują się w podręczniku pt. „Opis funkcji systemu AssetCenter: Wprowadzenie”, rozdział „Korzystanie z AssetCenter”, sekcja „Listy rekordów”, punkt „Eksportowanie listy”.

Zarządzanie widokami SQL w bazie danych AssetCenter

Moduł AssetCenter Export pozwala tworzyć, odtwarzać lub usuwać widoki SQL w bazie danych AssetCenter. Narzędzia zewnętrzne mogą korzystać z tych widoków zamiast z plików tekstowych.

Ostrzeżenie: Widoki SQL, które można tworzyć/modyfikować/usuwać za pomocą skryptów, różnią się od widoków w sensie znaczenia terminu w AssetCenter. Widok SQL jest odpowiednikiem instrukcji SQL „CREATE VIEW”.

Aby utworzyć, odtworzyć lub usunąć widok SQL w bazie danych AssetCenter:

- ↳ Uruchom moduł AssetCenter Export.
- ↳ Zdefiniuj skrypt eksportu z trybem ustawionym na „Tryb widoku”:
 - ❖ W zakładce **Zapytania** wpisz zapytania definiujące pobierane dane.

- ❖ W zakładce **Widoki** określ działania do wykonania: utworzenie, modyfikacja lub usunięcie widoków, wykonanie skryptu SQL bezpośrednio lub zapisanie go do pliku.
- ❖ Zapisz skrypt eksportu.
- ↪ Wykonaj skrypt eksportu:
 - ❖ bezpośrednio w module AssetCenter Export
 - ❖ lub uruchamiając program **amexpl32.exe**. Ostrzeżenie: Ten program jest dostępny tylko dla środowisk 32-bitowych.

Zalecenia

Nie zalecamy stosowania pola „Identyfikator” w tabelach jako klucza uzgodnienia, jeśli eksportowane dane mają być później ponownie importowane. Może to spowodować, że numery identyfikacyjne nie będą stałe i mogą zostać zmodyfikowane. Zamiast tego zalecamy stosowanie klucza o niezmiennej wartości, np. kod wewnętrzny zasobu.

Definiowanie skryptu eksportu

Aby eksportować dane lub generować widoki SQL dla bazy danych, należy zdefiniować skrypty eksportu i zapytania eksportowe w nich zawarte. Aby to zrobić, użyj modułu AssetCenter Export.

Skrypt eksportu działa:

- w „Trybie eksportu” dla eksportu danych.
- lub w „Trybie widoku”, aby tworzyć, odtwarzać lub usuwać widoki SQL z bazy danych.

Zawiera on:

- Zapytania eksportowe, które definiują, co powinno być pobrane z bazy danych.
- Opcje formatowania, dla eksportu danych.

- Działania do wykonania, dotyczące tworzenia/usuwania widoków SQL.

W tej sekcji opisano tworzenie skryptów eksportu:

- Metodologia
- Definiowanie zapytań eksportowych
- Format wyjściowy skryptu eksportu
- Działania dotyczące widoków SQL



Metodologia

Aby utworzyć lub zmodyfikować skrypt eksportu:


- ↪ Uruchom moduł AssetCenter Export.
- ↪ Otwórz właściwą bazę danych. Ostrzeżenie: Połączenie można nawiązać tylko za pomocą nazwy logowania „Admin” lub nazwy logowania użytkownika o uprawnieniach administratora.
- ↪ Utwórz nowy skrypt za pomocą polecenia menu **Plik/ Nowy skrypt** lub otwórz skrypt do edycji za pomocą polecenia menu **Plik/ Otwórz skrypt**.
- ↪ U góry ekranu modułu AssetCenter Export zdefiniuj, czy chcesz eksportować dane (Tryb eksportu), czy zarządzać widokami SQL (Tryb widoku) bazy danych.
- ↪ Wpisz zapytania skryptu eksportowego w zakładce **Zapytania**.
- ↪ Jeśli eksportujesz dane, podaj wyjściowy format eksportowanych danych w zakładce **Formatowanie**.
- ↪ Jeśli chcesz zarządzać widokami SQL, określ co chcesz zrobić w zakładce **Widoki**.
- ↪ Zapisz skrypt używając polecenia menu **Plik/ Zapisz skrypt** lub **Plik/ Zapisz skrypt jako**.

Definiowanie zapytań eksportowych

Zapytania skryptu eksportowego można zdefiniować w zakładce **Zapytania** modułu AssetCenter Export.

- Kliknij przycisk , aby dodać zapytanie eksportowe.
- Kliknij przycisk , aby usunąć wybrane zapytanie eksportowe.


Utwórz zapytanie w skrypcie eksportowym

- Kliknij przycisk  w zakładce **Zapytania**.
- Podaj nazwę zapytania. Nazwa ta jest używana w raporcie wykonania w zakładce **Komunikaty** szczegółów skryptu eksportowego.
- Możesz wpisać komentarze (nie są one eksportowane).
- Zdefiniuj dane do pobrania w polu **Zapytanie**.
- Jeśli raczej chcesz eksportować dane niż tworzyć/modyfikować/usuwać widoki, w polu **Plik** podaj ścieżkę i nazwę wynikowego pliku tekstowego, do którego zostaną zapisane dane wybrane za pomocą zapytania. Skrypt eksportu zawierający kilka pytań eksportowych wygeneruje kilka plików tekstowych.

Uwaga: Pole **Plik** nie jest wyświetlane, jeśli wybrano opcję **Utwórz/Usuń widoki SQL**.

Dane do pobrania

Aby wskazać dane do pobrania, wypełnij pole **Zapytanie** w szczegółach zapytania skryptu eksportowego. Zapytanie odnosi się do tabeli w bazie danych AssetCenter.

Możesz wpisać zapytanie bezpośrednio, lub kliknąć przycisk , aby uzyskać dostęp do okna ułatwiającego zdefiniowanie zapytania:

Eksportowane kolumny i kolejność sortowania

W zakładce **Kolumny i sortowanie** definiuje się listę pól, łączy, cech i pól wyliczanych do wyeksportowania, jak również związane z nimi kryteria sortowania.

Wybierz po kolei wszystkie pola, łączy, cechy i pola wyliczane do wyeksportowania z listy znajdującej się po lewej stronie i kliknij strzałkę, aby dodać je do listy po prawej stronie.

Dla każdej kolumny na liście po prawej stronie:

- ❖ Zaznacz pole wyboru **Widoczność**, aby wyeksportować kolumnę. Jeśli pole **Widoczność** nie będzie zaznaczone, kolumna nie zostanie wyeksportowana (może jednak zostać użyta do sortowania eksportowanych danych, itp).
- ❖ Zaznacz pole wyboru **Grupuj według**, aby grupować dane według pola zgodnego z kolumną. Jest to odpowiednik instrukcji „GROUP BY <Field name>” w zapytaniu SQL.

Przykład:

```
SELECT Brand, Count(lAstId) FROM amAsset GROUP BY Brand ORDER BY Brand
```

Ostrzeżenie: Jeśli zaznaczysz pole **Grupuj według**, instrukcja „GROUP BY” zostanie dołączona do zapytania, jednak aby zapytanie było poprawne, trzeba również dodać odpowiednie funkcje zebrane w instrukcji SELECT.

Zdefiniuj kolejność sortowania eksportowanych danych:

- ❖ Aby zdefiniować indeks sortowania, wybierz indeks w polu **Sortuj według indeksu**.
- ❖ W przeciwnym razie, zaznacz odpowiednie pole **Sortuj** dla żądanej kolejności sortowania.

Uwaga: Za wyjątkiem SQL Anywhere, można zaznaczyć opcję **Wymuś indeksy**, aby wymusić stosowanie indeksów w odniesieniu do zapytania. Więcej informacji znajduje się w podręczniku pt. „Opis funkcji systemu AssetCenter: **Administracja i zaawansowane zastosowania AssetCenter**”, rozdział „Pisanie zapytań w AQL”, sekcja „Sortowania i indeksy”.

Jeśli zaznaczysz opcję **Tylko unikalne rekordy**, dokładnie takie same linie będą eksportowane tylko raz. Jest to odpowiednik dodania instrukcji „DISTINCT” do zapytania SQL.

Przykład bez zaznaczenia pola **Tylko unikalne rekordy**:

```
SELECT Brand FROM Asset
```

Przykład z zaznaczeniem pola **Tylko unikalne rekordy**:

```
SELECT DISTINCT Brand FROM Asset
```

Filtry

Możesz zdefiniować dwa typy filtrów służących do określania pobieranych danych:


- ❖ Zapytanie AQL z użyciem klauzuli WHERE w zakładce **Filtr (warunek WHERE)**.
- ❖ Zapytanie AQL z użyciem klauzuli HAVING w zakładce **Warunek HAVING**.

Wyświetlanie zapytania

Zapytanie AQL zdefiniowane w zakładkach **Kolumny i sortowanie**, **Filtr (warunek WHERE)** i **Warunek HAVING** zostaje wyświetlone w zakładce **Zapytania**.

Podgląd wyników zapytania

W zakładce **Podgląd** można testować zapytanie i oglądać jego składnię w języku SQL.

Kliknij przycisk , aby uzyskać podgląd wyników, jako listę rekordów. Należy pamiętać, że AssetCenter wyświetla liczbę rekordów zgodnych z zapytaniem, u dołu okna po prawej stronie.

Format wyjściowy skryptu eksportu

Jeśli wybierzesz opcję **Eksportuj dane**, możesz zdefiniować format wynikowych plików tekstowych w zakładce **Formatowanie**. Format ten jest stosowany do wszystkich zapytań eksportowych.

Uwaga: Zakładka **Formatowanie** nie jest wyświetlana, jeśli wybierzesz opcję usuwania/ tworzenia/ odtwarzania widoków.

Tytuł kolumny

Wybierz odpowiednią wartość, jeśli pierwszą linią pliku eksportowanego ma zawierać:

- Aliasy kolumn podanych w zapytaniu eksportowym.
- „Nazwę SQL” pól lub łącz zgodnych z kolumnami.
- „Opis” pól lub łącz zgodnych z kolumnami.

Separator kolumn

Ten separator jest wstawiany pomiędzy danymi w każdej kolumnie.

Identyfikator tekstu

Identyfikator obejmuje tekstowe ciągi znaków. Jeśli zastosujesz znak ', wszystkie eksportowane znaki ' zostaną odtworzone jako „i odwrotnie, w przypadku znaku”.

Zestaw znaków

Opcja ta pozwala wybrać zestaw znaków ANSI (Windows) lub OEM (DOS).

Separator dziesiętny

Znak ten jest używany do oddzielania ułamka dziesiętnego eksportowanych liczb.

Separator daty

Znak ten jest wstawiany między dniem, miesiącem i rokiem eksportowanych dat.

Format daty

Format daty określa kolejność eksportowania dni (DD), miesięcy (MM) i lat (RR).

Format roku

Określa, czy lata są eksportowane w formacie 2- czy 4-cyfrowym.

Separator czasu

Znak ten jest wstawiany pomiędzy godzinami, minutami i sekundami.

Wyświetlaj sekundy

Wskaż, czy sekundy mają być uwzględniane w eksportowanych danych dotyczących czasu.

Działania dotyczące widoków SQL


Jeśli chcesz usuwać, tworzyć lub odtwarzać widoki SQL zgodne z zapytaniami eksportowymi, użyj zakładki **Widoki**, aby zdefiniować działania do wykonania.

Uwaga: Zakładka **Widoki** nie jest wyświetlona, jeśli została wybrana opcja „Eksportuj dane”.

Wybierz jedno z działań do wykonania w ramce „Działania”:

- Tworzenie lub odtwarzanie widoków.
- Usuwanie widoków.

W sekcji „Widok SQL - skrypt sterujący” określ, jak ma być przetwarzane zapytanie (zakładka **Zapytania**, ramka „Działania”):

- Aby tworzyć lub odtwarzać widoki SQL bezpośrednio przy uruchamianiu skryptu eksportowego, wybierz opcję **Uruchom SQL bezpośrednio**.
- Aby wygenerować skrypt widoku SQL tworzący widok (instrukcja „CREATE VIEW”) lub usuwający widok (instrukcja „DROP VIEW”), wybierz opcję **Zapisz kod SQL do pliku**, a następnie:
 - ❖ Kliknij przycisk , aby podać ścieżkę i nazwę pliku.
 - ❖ Podaj separator instrukcji SQL. Dostępne są opcje „;” (dla Oracle) i „GO” (dla wszystkich innych systemów DBMS).

Wykonywanie skryptu eksportu


Skrypty eksportu mogą służyć do eksportowania danych lub zarządzania widokami SQL.

W tej sekcji opisano dwie metody wykonywania skryptów eksportu:

- Wykonywanie skryptu eksportu z modułu AssetCenter Export
- Wykonywanie skryptów eksportowych w systemie DOS

Wykonywanie skryptu eksportu z modułu AssetCenter Export





Aby wykonać skrypt eksportu z modułu AssetCenter Export:


- Uruchom moduł AssetCenter Export.
- Zdefiniuj skrypt eksportu i zapisz go.
- Następnie wykonaj skrypt:
 - ❖ za pomocą polecenia menu **Działania/ Wykonaj skrypt**,
 - ❖ lub naciskając F8, albo
 - ❖ klikając przycisk .

Informacje o postępie procesu eksportu są wyświetlane w zakładce **Komunikaty**.

Jeśli eksport kończy się powodzeniem, ostatnim wyświetlanym komunikatem jest: „Wykonanie skryptu powiodło się”. Jeśli wystąpi błąd, wyświetlany jest następujący komunikat: „Podczas wykonywania skryptu wystąpił błąd”.

Przed wszystkimi komunikatami wyświetlane są ikony:

-  Informacje ogólne.
-  Błąd.
-  Zakończony powodzeniem.
-  Ostrzeżenie.

Przycisk  pozwala anulować eksport będący w toku.

Wykonywanie skryptów eksportowych w systemie DOS

Zasada

Aby uruchamiać program systemu DOS w trybie „on-line”, trzeba najpierw utworzyć skrypt eksportu za pomocą modułu AssetCenter Export.

Następnie można ręcznie lub automatycznie (np. za pomocą pliku wsadowego) uruchomić polecenie eksportu z zastosowaniem programu **amexpl32.exe** znajdującego się w folderze podrzędnym **Bin32** foldera instalacyjnego AssetCenter.

Program ten jest dostępny tylko dla środowisk 32-bitowych.

Składnia

```
amexpl32 [-verbose] [-?|h|H] -script:<script> -cnx:<cnx> [-login:<login>] [-password:<password>]
```

-verbose: wyświetla komunikaty podczas eksportu.

-.?, -h lub -H: wyświetla komunikaty pomocy dotyczące programu.

-script: ścieżka i nazwa skryptu eksportu do uruchomienia.

-cnx: nazwa połączenia z bazą danych AssetCenter (podana w poleceniu menu **Plik/ Zarządzaj połączeniami**).

-login: nazwa logowania administratora bazy danych („Admin” lub użytkownik z uprawnieniami administratora).

-password: hasło związane z nazwą logowania.

Ciagi znaków ujęte w <> nie mogą zawierać spacji.

Przykład:

```
amexpl32 -verbose -script:ibmassets.scx -cnx:GeneralDatabse -login:Gerald -password:PAssword
```

Rozdział 17 - Używanie skryptów

W niniejszym rozdziale opisano używanie skryptów.

Rozdział zawiera informacje na następujące tematy:

Wprowadzenie

- Definicja skryptu
- Zastosowanie skryptów
- Wprowadzenie do funkcji
- Klasyfikacja funkcji języka Basic

Pisanie skryptu

- Pierwszy krok pisania skryptów
- Wskazówki i ostrzeżenia

Przykłady

- Drugi przykład

Definicja skryptu

Omówienie

Słowo „skrypt” ogólnie oznacza program napisany w języku wysokiego poziomu. W AssetCenter, pojęcie to obejmuje trzy typy skryptów:

- Skrypty proceduralne obejmujące:
 - ❖ Skrypty obliczeniowe napisane w języku BASIC, używane do wyliczania wartości pól, warunkowania właściwości obiektów w bazie danych AssetCenter, itp.
 - ❖ Skrypty w języku BASIC wykonujące zadania, w szczególności w ramach działań.

Programy w języku BASIC zawierają funkcje. W niniejszym rozdziale omówiono właśnie ten rodzaj skryptu.

- Skrypty deklaracyjne. Są to skrypty eksportu i importu, napisane w języku skryptowym innym niż BASIC. Ten typ skryptu omówiono szczegółowo w podręcznikach pt. „Opis funkcji systemu AssetCenter: Administracja i zaawansowane zastosowania AssetCenter”, rozdział „Importowanie danych” oraz „Opis funkcji systemu AssetCenter: Administracja i zaawansowane zastosowania AssetCenter”, rozdział „Eksport danych i zarządzanie widokami SQL”
- „Mieszane”, deklaracyjno-proceduralne. Ten typ skryptu używany jest w kreatorach AssetCenter.

Informacje o tej wersji języka BASIC

Wersja języka BASIC używana w programie AssetCenter jest podzbiorem, opracowanym przez firmę Cypress, zgodnym z językiem „Visual Basic for Applications[™]”. Dodatkowe informacje o tym języku, jego strukturze i składni można znaleźć w dokumentacji języka BASIC.

Obsługiwane są tylko niektóre funkcje języka „Visual Basic for Applications”, np.:

- Funkcje dostępu do plików nie są obsługiwane.
- Ograniczone są funkcje obsługi dat i czasu, szczególnie w systemie UNIX.
- Formanty języka Visual Basic for Applications nie są dostępne.

Notacja dostępu do danych

Składnia języka BASIC używana w AssetCenter jest podobna do składni standardowej, za wyjątkiem funkcji dostępu do danych z bieżącego rekordu; używany jest następujący format:

```
[Link.Link.Field]
```

Przykład: **[Category.FullName]** z tabeli produktów.

Za pomocą następującej składni można pobrać numer identyfikacyjny łącza: **[Link.Link]**

Aby odwołać się do łącza, można zastosować nazwę SQL łącza lub nazwę klucza łącza.

```
Example:RetVal=[Contact.Location] or RetVal=[Contact.lLocaId]
```

Obydwa przykłady zwracają ten sam wynik, identyfikator łącza.

Zastosowanie skryptów

AssetCenter pozwala kojarzyć następujące właściwości ze skrypcem języka Basic :

- Dotyczące konfiguracji pól (polecenie **Konfiguruj obiekt** z menu kontekstowego lub AssetCenter Database Administrator):
 - ❖ Wartość domyślna.
 - ❖ Wartość obowiązkowa.
 - ❖ Historia.
 - ❖ Właściwość ‘Tylko do odczytu’.
- Dotyczące parametrów cechy związanej z tabelą:

- ❖ **Wartość domyślna** (nazwa SQL: DefValScript).
- ❖ **Dostępne** (nazwa SQL: seAvailable).
- ❖ **Zawsze widoczne** (nazwa SQL: seForceDisplay).
- ❖ **Obowiązkowe** (nazwa SQL: seMandatory).
- ❖ **Zachowuj historię** (nazwa SQL: seKeepHistory).
- Dotyczące działań typu **Skrypt**:
 - ❖ **Skrypt działania** (nazwa SQL: Script) dla typu **Skrypt**.
- W kreatorach:
 - ❖ Rozpoczęcie i zakończenie skryptów kreatorów.
 - ❖ Skrypty służące do definiowania właściwości węzłów.
- W polach wyliczanych typu „BASIC”.
- W workflow:
 - ❖ Czynności workflow typu **Test / skrypt**.
 - ❖ Zdarzenia workflow typu **Baza danych**.
 - ❖ „Wyliczane” osoby przypisywane do workflow.

Wprowadzenie do funkcji

W niniejszej sekcji podano następujące informacje:

- Definicja funkcji
- Funkcje wbudowane i programowalne
- Typy funkcji i parametrów

Definicja funkcji

Funkcja jest programem wykonującym operacje i zwracającym wartość do użytkownika. Wartość ta jest nazywana „wartością zwracaną” lub „kodem powrotu”.

Funkcje mają następującą strukturę:

```
Function <Function name> (<Parameter> As <Parameter type>[, ...,  
<Parameter> As <Parameter type>]) As <Function type>  
  
<Program (script) executed by the function. This program must define  
the return value.>  
  
End Function
```

Struktura ta obowiązuje zarówno w przypadku funkcji wbudowanych jak i funkcji programowalnych.

Funkcje wbudowane i programowalne

Funkcje wbudowane i funkcje programowalne to dwie główne kategorie funkcji dostępne w AssetCenter.

Funkcje wbudowane

Funkcje wbudowane są podobne do elementów oprogramowania napisanych dla użytkownika. Te elementy oprogramowania wykonują wszelkie rodzaje zadań (wyliczenia, konwersje danych dostarczanych przez użytkownika) i zwracają wynik. Użytkownik po prostu wywołuje funkcję za pomocą jej nazwy i przedstawia dane wymagane do zwrócenia wartości. Elementy danych dostarczane przez użytkownika zwane są „parametrami”.

Na przykład, funkcja „AmConvertCurrency()” przelicza kwotę w walucie A na kwotę w walucie B, za pomocą kursu wymiany zdefiniowanego dla danej daty. W tym przykładzie:

- Funkcja nosi nazwę „AmConvertCurrency”
- Parametry dostarczane przez użytkownika to:
 - ❖ Waluta A
 - ❖ Waluta B
 - ❖ Kwota do przeliczenia
 - ❖ Data wykonania konwersji (używana do określenia stosowanego kursu wymiany).


Funkcja wykonuje konwersję i dostarcza zwracaną wartość zgodną z wynikiem konwersji.

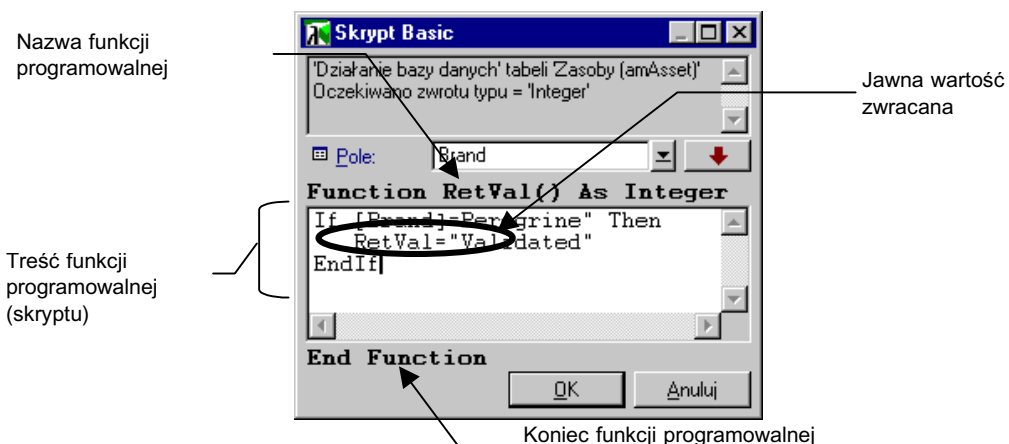
Funkcje programowalne

Funkcje programowalne są to elementy oprogramowania, które użytkownicy mogą pisać sami. Zadaniem użytkownika jest wyraźne zdefiniowanie wartości zwracanej w zmiennej „RetVal” (zwanej również „wartością zwracaną”) przez funkcję programowalną, w następującym formacie:

```
RetVal=<Expression>
```

AssetCenter nie skompiluje skryptu funkcji, dla której nie zdefiniowano zwracanej wartości.

Funkcje programowalne udostępniane są przez kompilator skryptu (po kliknięciu przycisku  w polu, do którego można zastosować skrypt). Kompilator skryptu ułatwia użytkownikom tworzenie elementów oprogramowania będących odpowiednikami funkcji. Kompilator skryptu zawiera szablon pisania funkcji programowalnych:



Kompilator skryptu

Opis funkcji programowalnej znajduje się w górnej części okna kompilatora skryptu. Identyfikuje on obiekt, którego dotyczy funkcja (na przykład, wartość domyślna pola **Kod kreskowy** (nazwa SQL: BarCode) w tabeli zasobów) jak również spodziewany typ zwracanej wartości (w powyższym przykładzie jest to: „String”).

Typy funkcji i parametrów

Typy funkcji

Typ funkcji wbudowanej jest typem wartości zwracanej przez funkcję. Zalecamy zwracać szczególną uwagę na ten punkt, ponieważ może on powodować błędy kompilacji i wykonania w skryptach języka BASIC.

Na przykład, nie można zastosować funkcji zwracającej wartość jednego typu przy definiowaniu wartości domyślnej pola innego typu. Na przykład, spróbuj przypisać następujący domyślny skrypt do jakiegokolwiek pola typu „Data” lub „Data+Czas”:

```
RetVal=AmLoginName()
```

Funkcja „AmLoginName()” zwraca nazwę połączonego użytkownika, w postaci ciągu znaków (typ String). Dlatego wartość zwracana jest w formacie niezgodnym z formatem pola „Data”, a AssetCenter wyświetla komunikat o błędzie.

Typy parametrów

Parametry używane w funkcjach wbudowanych również mają typ; należy więc przestrzegać tego typu, aby funkcja była wykonywana poprawnie. Jeśli wystąpi błąd związany z typem parametru, AssetCenter wyświetla komunikat o błędzie podczas wykonywania funkcji.

Lista typów

W poniższej tabeli podsumowano różne typy dostępne dla parametru lub funkcji:

Typ	Opis
Integer	Liczba całkowita od -32.768 do +32.767.
Long	Liczba całkowita od -2.147.483.647 do +2.147.483.646.
Double	8 bajtowa liczba zmiennoprzecinkowa.
String	Tekst w którym dozwolone są wszystkie znaki.
Date	Data lub Data+Czas.
Variant	Typ ogólny, który może reprezentować każdy typ.

Określanie typu zwracanej wartości funkcji programowalnej

Przed napisaniem skryptu, należy określić daną funkcję i jej typ. Informacja ta jest wyświetlana czcionką pogrubioną we wszystkich oknach „skryptu Basic” w następującym formacie:

```
Function <Function name>() As <Function type>
```

Trzy najczęściej występujące typy funkcji to „Boolean”, „Integer” i „String”:

- Funkcje typu „Boolean” zwracają wartość „TRUE” lub „FALSE”; każda inna wartość powoduje błąd podczas kompilacji skryptu Basic.
- Funkcje typu „Integer” zwracają tylko liczby całkowite (na przykład 0, 1, 8, 12).
- Funkcje typu „String” zwracają tylko ciągi znakowe (na przykład: „Budynek21”) ujęte w cudzysłów.

Nieprzestrzeganie typu funkcji może powodować błędy podczas kompilacji programu języka Basic. Należy zawsze zwracać uwagę na typ używanej funkcji.

Nazwa i typ funkcji pozwalają określić kod powrotu, który należy zastosować w skrypcie, w następującym formacie:

```
RetVal=<Expression respecting the function type>
```

Klasyfikacja funkcji języka Basic

Język BASIC używany w skryptach opiera się na różnych klasach funkcji:

- Tradycyjnych funkcjach języka Basic zgodnych ze standardem „Visual Basic for Applications™”
- Funkcjach ogólnych właściwych dla AssetCenter, które mogą być stosowane wszędzie tam, gdzie używane są skrypty.
- Funkcjach specyficznych, które mogą być używane w niektórych częściach AssetCenter.

Pierwszy krok pisania skryptów

W niniejszym rozdziale omówiono działanie skryptów i podano przykładowy scenariusz:

- Przykładowy scenariusz
- Krok 1: Utwórz cechę „Podręcznik”
- Krok 2: Otwórz okno edycji
- Krok 3: Analiza i definicja algorytmu
- Krok 4: Napisz skrypt w języku Basic
- Krok 5: Test skryptu BASIC

Przykładowy scenariusz


Cel

Zagwarantować, aby cecha „Podręcznik” była dostępna jedynie dla produktów w **Kategorii** „Komputer/ Płyta główna”.


Metoda


Dołączenie skryptu języka Basic do parametru **Dostępne** (nazwa SQL: seAvailable) cechy „Podręcznik”.

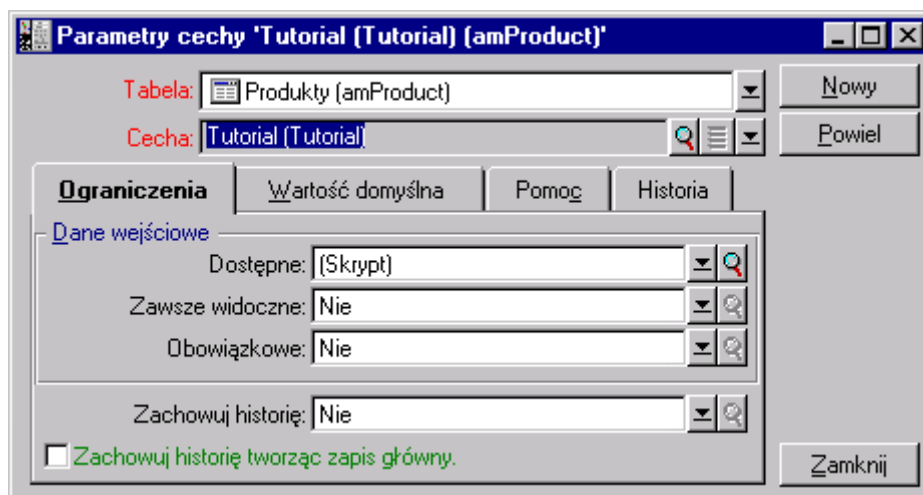
Krok 1: Utwórz cechę „Podręcznik”

Wybierz polecenie menu **Narzędzia/ Administracja/ Cechy**. Kliknij , aby utworzyć nową cechę. Wypełnij tę cechę zgodnie z poniższymi wskazówkami:

Nazwa pola	Wartość
Nazwa (nazwa SQL: TextLabel)	„Podręcznik”
Nazwa SQL (nazwa SQL: SQLName)	„Podręcznik” (Nazwa używana w tym przykładzie)
Typ wpisu (nazwa SQL: seDataType)	Numeryczne


Kliknij przycisk , aby utworzyć tę cechę.

Przejdź do opcji **Parametry** i kliknij przycisk , aby zmodyfikować parametry cechy „Podręcznik”. Wypełnij zakładkę **Ograniczenia** zgodnie z następną ilustracją.



Parametry cechy „Podręcznik”

Krok 2: Otwórz okno edycji

W zakładce **Ograniczenia**, ustaw wartość parametru **Dostępne** (nazwa SQL: seAvailable) jako „(Script)”. Kliknij przycisk z lupą . AssetCenter otworzy okno edycji skryptu:

Krok 3: Analiza i definicja algorytmu


Algorytm musi spełniać następujące zadania:

- Ustawiać wartość pola **Dostępne** (nazwa SQL: seAvailable) na **Tak**, jeśli produkt należy do kategorii „/Komputer/ Płyta główna”.
- Ustawiać wartość pola **Dostępne** (nazwa SQL: seAvailable) na **Nie** we wszystkich pozostałych przypadkach.


Nasz algorytm ma wobec tego następującą postać:

```
If the category of the product is "/Computer/Motherboard" then
The feature is available
Else
The feature is not available
```

Wartość pola **Kategoria** (nazwa SQL: Category) w tabeli produktów warunkuje zatem wartość pola **Dostępne** (nazwa SQL: seAvailable) cechy. Tylko to pole występuje w naszym algorytmie.

Kliknij przycisk z lupą  obok pola **Dostępne** (nazwa SQL: seAvailable), aby rozpocząć edycję skryptu Basic. Lista rozwijana w oknie edycji pozwala wybrać pole **Kategoria** (nazwa SQL: Category) z tabeli produktów.

Pole **Kategoria** (nazwa SQL: Category) jest polem typu „Łącze”.

Po wybraniu pola, przenieś je do pola edycji klikając przycisk .

Krok 4: Napisz skrypt w języku Basic

Teraz trzeba zapisać w języku BASIC algorytm przygotowany w kroku 2.






```
If [Category]= "/Computer/ Motherboard" Then
  RetVal=True
Else
  RetVal=False
End If
```

Skrypty nie rozróżniają dużych i małych liter.

Kliknij przycisk , aby zatwierdzić skrypt.

Krok 5: Test skryptu BASIC

W tym kroku sprawdzamy, czy skrypt działa prawidłowo.

- Otwórz tabelę produktów wybierając polecenie menu **Repozytorium/ Produkty**. Kliknij , aby utworzyć nowy produkt.
- Wypełnij tylko pola obowiązkowe.
 - ❖ Ustaw pole **Kategoria** (nazwa SQL: Category) jako „/Komputer/Płyta główna”.
 - ❖ Ustaw pole **Model** (nazwa SQL: Model) i **Kod kreskowy** (nazwa SQL: BarCode) na dowolną wartość.
- Kliknij przycisk , aby utworzyć nowy produkt.
- Wybierz zakładkę **Cechy** i kliknij przycisk , aby dodać cechę. Na ekranie wyboru wyświetlona jest nazwa cechy, dla której właśnie utworzono skrypt.
- Zmień wartość kategorii produktu na „/Komputer”. Kliknij , aby zatwierdzić zmianę.
- Wybierz zakładkę **Cechy** w szczegółach produktu i kliknij przycisk , aby dodać cechę. Ekran wyboru nie wyświetla już nazwy cechy, dla której właśnie utworzono skrypt.

Skrypt prawidłowo spełnia swe zadanie.

Wskazówki i ostrzeżenia

W niniejszej sekcji podano kilka wskazówek pomocnych przy pisaniu skryptów.

- Uwaga na funkcje programowalne
- Format stałych typu „Data+Czas” w skryptach

- Format stałych typu „Czas trwania” w skryptach
- Dostęp do listy pozycji systemu w trybie odczytu i zapisu
- Łącze wirtualne „CurrentUser”
- Komentarze w skrypcie Basic
- Inicjowanie komunikatu o błędzie

Uwaga na funkcje programowalne

Oto kilka środków ostrożności, o których należy pamiętać przy pisaniu własnych skryptów:

- ↳ Celem funkcji programowalnych, np. takich, które definiują wartość domyślną pola lub łącza, jest ustawianie wartości zwracanej przez funkcję. Dlatego usilnie odradzamy wykonywanie innych operacji w ramach funkcji programowalnej. W najlepszym przypadku, może nastąpić ogólny spadek wydajności, a w najgorszym przypadku, może dojść do uszkodzenia bazy danych.
- ↳ Funkcje programowalne są szeroko stosowane w AssetCenter. Wszędzie tam, gdzie to możliwe, staraj się optymalizować swe skrypty, aby utrzymać ogólną wydajność AssetCenter.

Format stałych typu „Data+Czas” w skryptach

Odwołania do dat w skryptach wyrażane są w formacie międzynarodowym, bez względu na opcje wyświetlania stosowane przez użytkownika:

rrrr/mm/dd gg:mm:ss

Przykład:

```
RetVal="1998/07/12 13:05:00"
```

Jako separator daty można również stosować łącznik („-”).

Data typu „Basic” i data typu „Unix”

W języku Basic i w systemie Unix daty wyrażane są odmiennie:

- W języku Basic, data może być wyrażona w formacie międzynarodowym, lub jako numeryczna wartość zmiennoprzecinkowa (typ „Double”). W tym przypadku, część wartości numerycznej będąca liczbą całkowitą oznacza liczbę dni od północy 30 grudnia 1899 roku, a część dziesiętna oznacza czas bieżącego dnia (liczba sekund od początku dnia podzielona przez 86400).
- W systemie Unix, daty wyrażane są jako liczby całkowite typu „long” oznaczające liczbę sekund od północy 1 stycznia 1870 niezależnie od strefy czasowej (czas UTC).

Format stałych typu „Czas trwania” w skryptach

W skryptach, czasy trwania są przechowywane i wyrażane w sekundach. Na przykład, aby ustawić wartość domyślną pola typu „czas trwania” na 3 dni, zastosuj następujący skrypt:

```
RetVal=259200
```

Podobnie, funkcje wyliczające czasy trwania, jak np. funkcja „AmWorkTimeSpanBetween()”, zwracają liczbę sekund.

Dla potrzeb konwersji, AssetCenter przyjmuje rok o długości 12 miesięcy, a miesiąc o długości 30 dni (zatem 1 rok = 360 dni).

Dostęp do listy pozycji systemu w trybie odczytu i zapisu

AssetCenter zarządza listami pozycji systemu przypisując liczbę całkowitą do każdej możliwej wartości na liście pozycji.

Weźmy przykład listy pozycji używanej do wypełniania pola **Przypisanie** (nazwa SQL: seAssignment) w zakładce **Ogólne** szczegółów zasobu.

W poniższej tabeli podano wartości używane na liście pozycji:

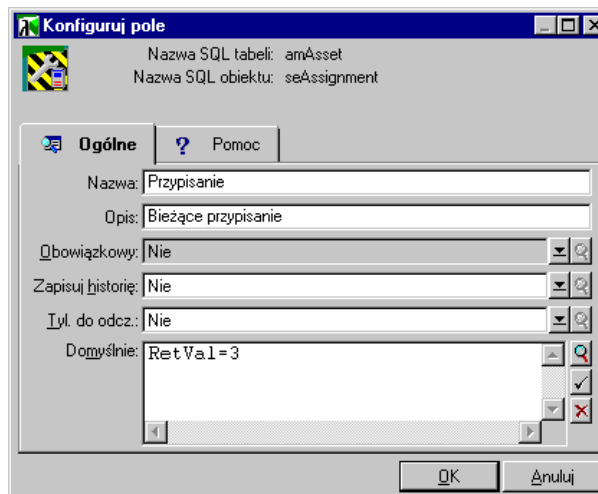
Wartość listy pozycji	Skojarzona liczba całkowita
W użyciu	0
W zapasie	1
Zasób wycofany	2
Oczekiwanie na dostawę	3

Zatem, aby zdefiniować domyślną wartość listy pozycji, należy:

- Ustalić, jaka liczba całkowita odpowiada właściwej wartości
- Poddać edycji następujący ciąg:

```
RetVal=<Integer associated with the appropriate value>
```

Na podstawie tego przykładu, jeśli chcemy ustawić domyślną wartość listy pozycji systemu używanej w polu **Przypisanie** na **Oczekiwanie na dostawę**, trzeba zmienić ciąg w następujący sposób:



Ustawianie domyślnej wartości listy systemu

Nie należy mylić listy pozycji systemu z zamkniętą listą pozycji zdefiniowaną przez użytkownika.

Pełny wykaz wartości listy pozycji systemu znajduje się w pliku **Database.txt**, zapisanym w folderze podrzędnym „infos” foldera instalacyjnego AssetCenter. Dwie kolumny, „Typ wyświetlania i wprowadzania danych” i „Dodatkowe informacje o typie wyświetlania i wprowadzania danych”, opisują typ listy pozycji i wartości przyjmowane przez listę pozycji.

Łącze wirtualne „CurrentUser”

Definicja

„CurrentUser” można rozpatrywać jako łącze zaczynające się we wszystkich tabelach i wskazujące na rekord w tabeli wydziałów i pracowników odpowiadający aktualnemu użytkownikowi.

- W formacie „CurrentUser” łącze to wskazuje na rekord odpowiadający aktualnemu użytkownikowi i zwraca numer identyfikacyjny aktualnego użytkownika.
- W formacie „CurrentUser.<nazwa SQL pola>” łącze to zwraca wartość pola aktualnego użytkownika.

Łącze wirtualne nie jest wyświetlane na liście pól i łączy; dlatego nie jest bezpośrednio dostępne w kompilatorze skryptów. Wyrażenie to należy wpisywać ręcznie.

Elementy równoważne

Funkcje „AmLoginName()” i „AmLoginId()”, które zwracają nazwę i numer identyfikacyjny aktualnego użytkownika, można uważać za funkcje pochodzące od „CurrentUser”. W efekcie, poniższe zapisy są równoważne:

- AmLoginName()=[CurrentUser.Name]
- AmLoginId()=[CurrentUser.lPersId]

Komentarze w skrypcie Basic

Często użyteczne jest dodawanie komentarzy do skryptu języka Basic, aby jasno określić, jakie zadanie wykonuje lub umożliwić użytkownikowi

zrozumienie i ułatwić modyfikację skryptu. AssetCenter daje możliwość komentowania treści skryptu za pomocą znaku apostrofu ('). Wszystkie znaki po pojedynczym znaku apostrofu w tej samej linii są ignorowane przez kompilator, który interpretuje je jako komentarze. Są również dwie możliwe sytuacje:

- Komentarz znajduje się w osobnej linii skryptu Basic, tak jak pokazano poniżej; wtedy nie są potrzebne żadne inne środki ostrożności

```
' Here we test the value of the BarCode field in the table of
assets
' If this value is PC1, the return code is set to TRUE
If [BarCode]=„PC1” Then
RetVal=True
End If
```

- Lub komentarz dodawany jest na końcu linii interpretowanej przez kompilator języka BASIC. W tym przypadku należy zastosować znak „:”, aby oddzielić skrypt od komentarza. Komentarz nadal poprzedzamy pojedynczym apostrofem.

```
If [BarCode]= "PC1" Then : ' Then BarCode is PC1
RetVal=TRUE : ' The return value is set to TRUE
End If : ' End of test
```

Inicjowanie komunikatu o błędzie

Można celowo zainicjować komunikat o błędzie za pomocą funkcji Err.Raise. Jej składnia jest następująca:

```
Err.Raise (<Error number>, <Error message>)
```

W przypadku, gdy tworzenie lub modyfikacja rekordu zostanie uznana za nieważną, ze względu na wartość pola „Ważność” danej tabeli, dobrze jest zainicjować komunikat o błędzie za pomocą funkcji Err.Raise, aby ostrzec użytkownika. Bez tego komunikatu o błędzie, użytkownik może nie zrozumieć, dlaczego nie można utworzyć lub zmodyfikować rekordu.

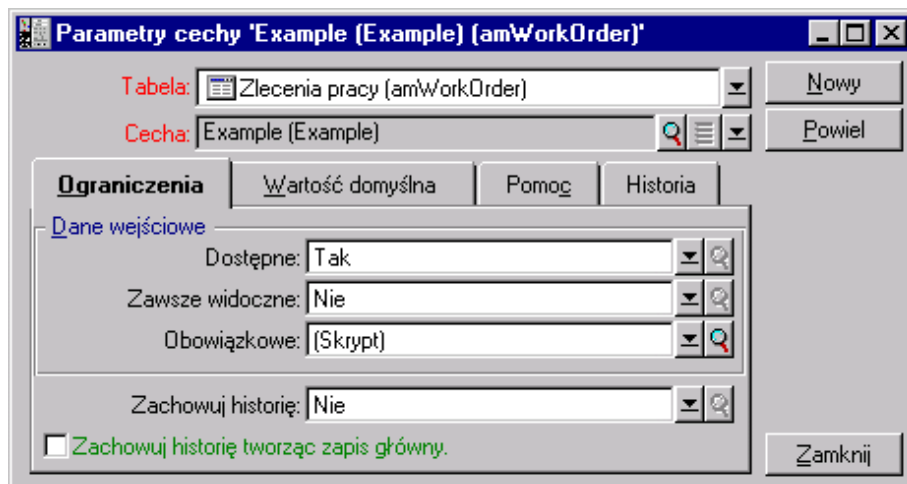
Pierwszy przykład

W tej sekcji omówiono hipotetyczny problem, który można rozwiązać za pomocą skryptu języka BASIC. Zalecamy samodzielną próbę rozwiązania problemu przed sprawdzeniem podanego rozwiązania.

- Opis problemu
- Krok 1: Analiza i definicja algorytmu
- Krok 2: Utworzenie skryptu BASIC
- Krok 3: Test skryptu BASIC

Opis problemu

Cechę o nazwie „Przykład1”, związaną z tabelą zleceń pracy, należy wypełniać przy zamykaniu zleceń pracy. Cecha ta jest opcjonalna dla niezamkniętych zleceń pracy. W dalszej części przykładu zakładamy, że cecha już została utworzona, ma dowolny typ wprowadzania, jest związana z tabelą zleceń pracy, jest dostępna i wyświetlana domyślnie (wyświetlanie wymuszone), tak jak pokazano na poniższej ilustracji:



Parametry cechy „Przykład1”

Krok 1: Analiza i definicja algorytmu

Algorytm musi realizować następujące zadania:

- Ustawiać wartość pola **Obowiązkowe** (nazwa SQL: seMandatory) na **Tak**, jeśli karta zgłoszenia została zamknięta.
- Ustawiać wartość pola **Obowiązkowe** (nazwa SQL: seMandatory) na **Nie** we wszystkich innych przypadkach.


Nasz algorytm jest zatem następujący

```
If the work order is closed then
Populating the feature is mandatory
Else
Populating the feature is not mandatory
```

Zlecenie pracy jest zamknięte, jeśli pole **Stan** (nazwa SQL: seStatus) ma wartość **Zamknięte**.

Zatem wartość pola **Stan** (nazwa SQL: seStatus) w tabeli zleceń pracy warunkuje wartość pola **Obowiązkowe** (nazwa SQL: seMandatory) cechy. Tylko to pole pojawia się w naszym algorytmie.

Za pomocą listy rozwijanej w oknie edycji można znaleźć pole **Stan** w tabeli zleceń pracy.

Po wybraniu pola, przenieś je do okna edycji, klikając przycisk .

To pole jest wypełniane na podstawie listy pozycji systemu.

Jest ona następująca:

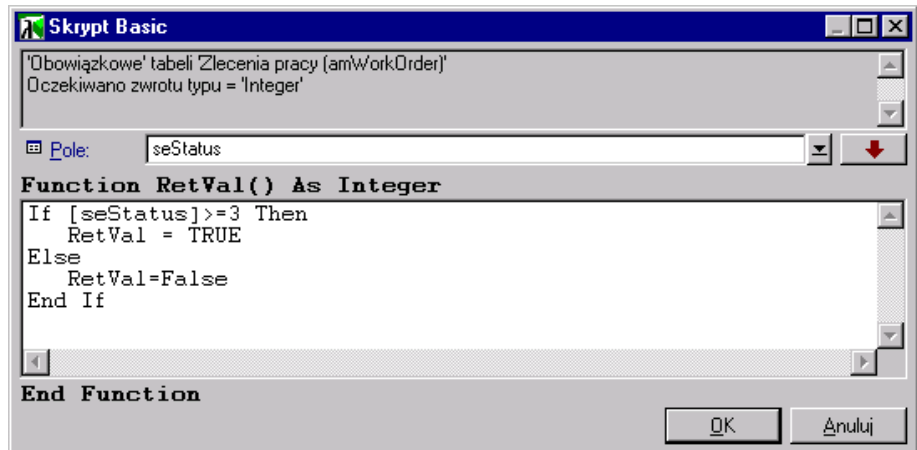
Wartość listy pozycji	Skojarzona liczba
Zgłoszone	
Zaplanowane	
W toku	
Zamknięte	3

Wartość z listy pozycji, która nas interesuje, to:


- **Zamknięte**, co odpowiada wartości „3”

Krok 2: Utworzenie skryptu BASIC

Teraz algorytm z kroku 1 trzeba wyrazić w języku BASIC.



Edycja programu

Kliknij opcję , aby zatwierdzić skrypt.

Krok 3: Test skryptu BASIC

Etap ten pozwala zagwarantować, że skrypt będzie działał prawidłowo.

- Otwórz tabelę zleceń pracy za pomocą polecenia menu **Obsługa/ Zlecenia pracy**. Wybierz kartę zgłoszenia, która jest **Zamknięte** (jeśli brak takiej karty, utwórz zlecenie pracy o takim stanie).
- Wybierz zakładkę **Cechy**. AssetCenter dodał cechę do zlecenia pracy, a cecha ta jest obowiązkowa.
- Teraz wybierz zlecenie pracy o statusie innym niż **Zamknięte**. Przejdź do zakładki **Cechy** tego zlecenia pracy. Cecha „Przykład1” jest wyświetlona, ale jest polem opcjonalnym.

Skrypt działa prawidłowo.

Drugi przykład

W niniejszej sekcji omówiono hipotetyczny problem, który można rozwiązać za pomocą skryptu BASIC. Zalecamy próbę samodzielnego rozwiązania problemu przed sprawdzeniem podanego rozwiązania.

- Opis problemu
- Krok 1: Analiza i definicja algorytmu
- Krok 2: Utworzenie skryptu BASIC
- Krok 3: Test skryptu BASIC

Opis problemu

Chcemy, aby domyślnie w polu **Pole1** (nazwa SQL: Field1) w szczegółach pracownika wyświetlone było nazwisko i imię pracownika, jeśli oba te elementy istnieją, lub nazwisko jeśli brak imienia.

Krok 1: Analiza i definicja algorytmu

Algorytm musi realizować następujące zadania:

- Wyświetlać domyślnie nazwisko i imię pracownika w polu **Pole1** (nazwa SQL: Field1) w szczegółach pracownika, jeśli istnieje zarówno nazwisko jak i imię pracownika.
- Wyświetlać domyślnie samo nazwisko pracownika w polu **Pole1** (nazwa SQL: Field1) w szczegółach pracownika, jeśli imię nie istnieje.

Nasz algorytm jest zatem następujący:

```
If the first name of the employee does not exist then
The default value of "Field1" is the name of the employee
Else
The default value of "Field1" is "Name, First"
```

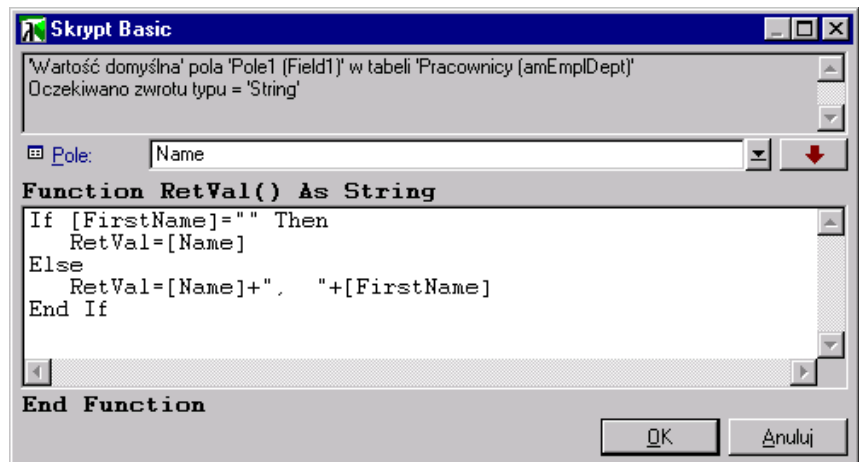
Wartości pól **Nazwisko** (nazwa SQL: Name) i **Imię** (nazwa SQL: FirstName) w tabeli pracowników warunkują wartość domyślną pola **Pole1** (nazwa SQL: Field1) w szczegółach pracownika. Tylko te dwa pola pojawiają się w naszym algorytmie.

Kliknij prawym przyciskiem myszy pole **Pole1** (nazwa SQL: Field1) w szczegółach pracownika i wybierz opcję **Konfiguruj obiekt**.

Kliknij przycisk z lupą  obok pola **Domyślne**, aby poddać edycji skrypt BASIC.

Krok 2: Utworzenie skryptu BASIC

Teraz należy wyrazić algorytm z kroku 1 w języku BASIC.




Edycja programu

Kliknij przycisk , aby zatwierdzić skrypt.

Krok 3: Test skryptu BASIC

Na tym etapie można upewnić się, czy skrypt działa prawidłowo.

- Otwórz tabelę pracowników wybierając polecenie menu **Repozytorium/ Wydziały i pracownicy** i utwórz nowego pracownika.
- Wypełnij pola **Nazwisko** (nazwa SQL: Name) i **Imię** (nazwa SQL: FirstName), następnie kliknij opcję , aby to zatwierdzić. AssetCenter wyświetla nazwisko i imię w polu **Pole1** (nazwa SQL: Field1).

Skrypt działa prawidłowo.

Rozdział 18 - Zarządzanie kalendarzami

Użyj polecenia menu Narzędzia/ Kalendarze, aby uzyskać dostęp do listy kalendarzy.

W niniejszym rozdziale wyjaśniono, jak opisywać i zarządzać kalendarzami. Rozdział zawiera informacje na następujące tematy:

Wprowadzenie

- Omówienie kalendarzy
- Wpływ kalendarzy na niektóre zakresy funkcjonalności

Tworzenie kalendarza

- Metodologia tworzenia kalendarza
- Opis tworzenia kalendarza

Omówienie kalendarzy

Szczegóły kalendarza

W szczegółach kalendarza wyświetlane są:

- Informacje ogólne, które umożliwiają unikalną identyfikację kalendarza:

- ❖ Pole **Nazwa** (nazwa SQL: Name) kalendarza.
- ❖ Pole **Strefa czasowa** (nazwa SQL: TimeZone), z którą związany jest kalendarz.
- Opis normalnych godzin pracy w zakładce **Harmonogramy**.
- Lista wyjątków od harmonogramu w zakładce **Wyjątki**.
- Podgląd godzin pracy za dany okres, uwzględniający reguły określone w zakładkach **Harmonogram** i **Wyjątki**.

Używanie kalendarzy

Kalendarze związane są z:

- Czynnościami workflow.

Umożliwiają one również określenie pór, kiedy alarmy zdefiniowane w czynnościach workflow mają być inicjowane. **Limit czasu** definiowany w tych alarmach jest zatem podawany w godzinach pracy.

Wpływ kalendarzy na niektóre zakresy funkcjonalności

Kalendarze mają wpływ na niektóre zakresy funkcjonalności AssetCenter. Modyfikacja kalendarza powoduje bezpośrednie i pośrednie zmiany w niektórych rekordach bazy danych. Kalendarze odgrywają rolę w:

- Czasach wykonania zadań workflow.
- Alarmach związanych z czynnościami workflow.

Metodologia tworzenia kalendarza

Kalendarze umożliwiają precyzyjne określenie godzin pracy.

Aby utworzyć kalendarz wykonaj następujące kroki:

- Rozpocznij od identyfikacji kalendarza nadając mu **Nazwa** (nazwa SQL: Name).
- Jeśli jest taka potrzeba, skojarz kalendarz ze strefą czasową wypełniając pole **Strefa czasowa** (nazwa SQL: TimeZone).
- Zdefiniuj codzienne godziny pracy w zakładce **Harmonogramy** szczegółów kalendarza.
- Następnie zdefiniuj wszystkie wyjątki od godzin pracy w zakładce **Wyjątki** szczegółów kalendarza.
- Na koniec możesz zweryfikować działanie kalendarza w zakładce **Podgląd**.

Opis tworzenia kalendarza

Kalendarz tworzony jest w następujących krokach:

- Wprowadzenie informacji ogólnych
- Wypełnianie zakładki Harmonogram
- Wypełnianie zakładki Wyjątki
- Sprawdzanie kalendarza

Wprowadzenie informacji ogólnych

Przed wprowadzeniem aktualnych godzin pracy i wyjątków, należy oznaczyć kalendarz wypełniając pole **Nazwa** (nazwa SQL: Name) na ekranie szczegółów.

Można również skojarzyć kalendarz ze strefą czasową wypełniając pole **Strefa czasowa** (nazwa SQL: TimeZone).

Wypełnianie zakładki *Harmonogram*

Zakładka **Harmonogram** w szczegółach kalendarza określa tygodniowe harmonogramy skojarzone z tym kalendarzem. Okresy godzin pracy zdefiniowane w tej zakładce określają tygodniowe harmonogramy związane z tym kalendarzem. Okresy te opisują ogólną zasadę. Święta państwowe, itp. są wyjątkami definiowanymi w zakładce **Wyjątki**.

Definicje harmonogramu tygodniowego:	
	0 3 6 9 12 15 18 21 24
Poniedziałek:	8:00-12:00;14:00-18:00
Wtorek:	8:00-12:00;14:00-18:00
Środa:	8:00-12:00;14:00-18:00
Czwartek:	8:00-12:00;14:00-18:00
Piątek:	8:00-12:00;14:00-18:00
Sobota:	
Niedziela:	

Zakładka *Harmonogramy* w szczegółach kalendarza

Dla każdego dnia tygodnia można zdefiniować jeden lub więcej okresów harmonogramu oznaczających godziny pracy. Można je zdefiniować za pomocą dwóch metod:

- Graficznie, za pomocą suwaków z podziałką oznaczającą dni tygodnia.
 - ❖ Kliknij formant na początku okresu harmonogramu.
 - ❖ Rozszerz obszar zaznaczenia przeciągając myszą do końca okresu harmonogramu. AssetCenter automatycznie wypełnia pole tekstowe po prawej stronie formantu z podziałką.
 - ❖ Powtarzaj czynności w miarę potrzeb.
- „Ręcznie”, za pomocą pola tekstowego. W polu tym obowiązuje następująca składnia:

```
<Start time of business period>-<End time of business period>;<Start time of business period>-<End time of business period>;...
```

Dla czasu stosowany jest następujący format:

```
<hh:mm [ {AM|PM} ] >
```

Jeśli opcjonalny parametr [AM | PM] nie jest zdefiniowany, AssetCenter uznaje, że stosowany jest format 24-godzinny. AssetCenter automatycznie wypełnia suwak z podziałką znajdujący się po lewej stronie pola tekstowego.

Formant graficzny działa z dokładnością do pół godziny. Formant tekstowy daje dokładność co do minuty.

Wypełnianie zakładki **Wyjątki**

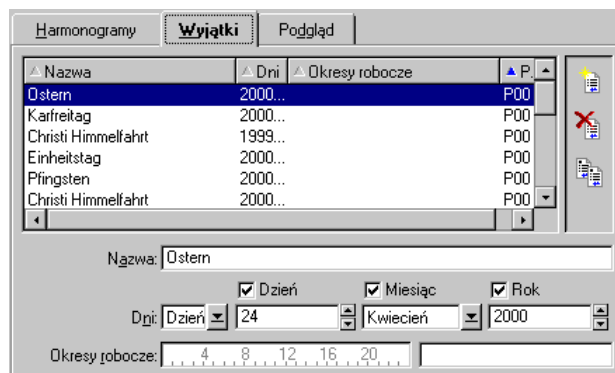
Zakładka **Wyjątki** w szczegółach kalendarza definiuje wyjątki w tygodniowych okresach pracy określonych w zakładce **Harmonogramy**.

Metodologia

Wyjątek tworzy się w następujący sposób:

- ↪ Określ **Nazwę** wyjątku.
- ↪ Zdefiniuj obszar zastosowania wyjątku wypełniając pole **Dzień**. Wyjątki można definiować dla danego dnia, miesiąca lub roku.
- ↪ Można również zdefiniować okres pracy objęty wyjątkiem za pomocą pola **Okresy pracy**. Za pomocą tego pola można zdefiniować bardziej precyzyjne wyjątki, np.: „W ostatni piątek każdego miesiąca, zespół pracuje od 8:30 do 10:30 i od 17:30 do 18:30”.




Zakładka Wyjątki



Zakładka Wyjątki w szczegółach kalendarza

Zakładka ta dzieli się na dwie części.

- W pierwszej części podana jest lista wyjątków, które można tworzyć, powielać, usuwać, modyfikować i anulować za pomocą przycisków na pasku narzędzi:

- ❖ : Kliknij ten przycisk, aby utworzyć nowy wyjątek.
- ❖ : Kliknij ten przycisk, aby usunąć wyjątek.
- ❖ : Kliknij ten przycisk, aby powielić wyjątek.

Kolumna „Priorytet” pozwala sortować wyjątki według priorytetu: Określa ona, który wyjątek jest ważniejszy w przypadku sprzeczności. AssetCenter automatycznie przydziela priorytet (od „P00” do „P15”) do wyjątku. Im niższa liczba, tym wyższy priorytet wyjątku. Zatem wyjątek o priorytecie „P06” jest ważniejszy niż wyjątek o priorytecie „P10”.

- W drugiej części podaje się szczegóły wyjątku.
Wartości przyjmowane przez pole **Dzień** określają kontekst zastosowania wyjątku:

Wartość pola <i>Dzień</i>	Kontekst zastosowania wyjątku
„Codziennie”	Wyjątek odnosi się do wszystkich dni roku bez wyjątku.
„Dzień roku:”	Wyjątek odnosi się do danego dnia lub okresu dni, zdefiniowanego za pomocą pól Dzień , Miesiąc i Rok .
„Pierwszy „	Reguła znajduje zastosowanie do dnia tygodnia zdefiniowanego w polu Dzień miesiąca(-y) i roku(lat) zdefiniowanych w polach Miesiąc i Rok . Przykład: „Pierwszy” piątek każdego miesiąca.
„Drugi”	Reguła znajduje zastosowanie do dnia tygodnia zdefiniowanego w polu Dzień miesiąca(-y) i roku(lat) zdefiniowanych w polach Miesiąc i Rok . Przykład: „Drugi” poniedziałek września.
„Przedostatni”	Reguła znajduje zastosowanie do dnia tygodnia zdefiniowanego w polu Dzień miesiąca(-y) i roku(lat) zdefiniowanych w polach Miesiąc i Rok . Przykład: „Przedostatnia” środa listopada.
„Ostatni”	Reguła znajduje zastosowanie do dnia tygodnia zdefiniowanego w polu Dzień miesiąca(-y) i roku(lat) zdefiniowanych w polach Miesiąc i Rok . Przykład: „Ostatni” każdego miesiąca w roku 1998.

Przykład

Pracownicy w firmie Taltek mają następujące dni wolne od pracy:

- ↳ Pierwszy piątek każdego miesiąca jest dniem wolnym od pracy.
- ↳ W sierpniu, pracownicy firmy Taltek pracują tylko rano od 8:30 do 12:30.

Reguła nr 1: Pierwszy piątek każdego miesiąca jest dniem wolnym.

❖ Kliknij .

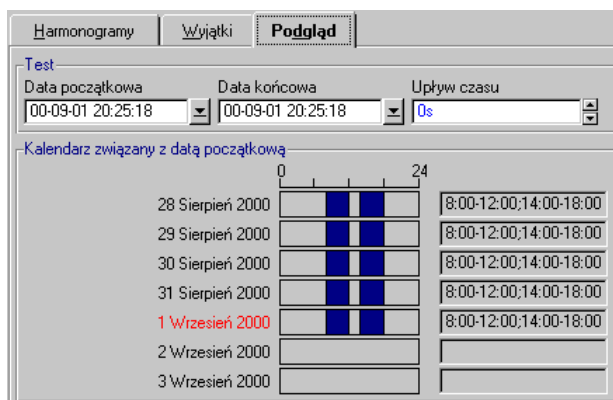
- ❖ Wyjątek odnosi się do pierwszego piątku każdego miesiąca wszystkich lat. Pola wyboru **Miesiąc** i **Rok** pozostają zatem niezaznaczone, co oznacza, że wyjątek stosowany jest niezależnie od miesiąca, czy roku. Pole wyboru **Dzień** jest zaznaczone, ponieważ wyjątek odnosi się tylko do piątków.
- ❖ Na koniec ustaw wartość pola **Dni** na: „Pierwszy”.

Reguła nr 2: W sierpniu pracownicy firmy Taltek pracują tylko rano od godziny 8:30 do 12:30.

- ❖ Kliknij .
- ❖ Wyjątek ten zależy tylko od miesiąca (odnosi się tylko do sierpnia). Pola wyboru **Dzień** i **Rok** pozostają zatem niezaznaczone, pole wyboru **Miesiąc** jest zaznaczone, a wartość jest ustawiona na „sierpień”.
- ❖ Ponieważ wyjątek odnosi się tylko do wybranych dni (wszystkich dni w sierpniu), pole **Dni** trzeba ustawić na „Dzień roku”.
- ❖ W tym okresie pracownicy pracują od godziny 8:30 do 12:30. Na zakończenie wprowadzania wyjątku wprowadź okres od 8:30 do 12:30 w polu **Okresy pracy**.

Sprawdzanie kalendarza

Zakładka **Podgląd** pozwala zastosować reguły zdefiniowane w zakładkach **Harmonogramy** i **Wyjątki** do okresu wybranego za pomocą pól **Data początkowa** i **Data końcowa**.



Zakładka **Podgląd** w szczegółach kalendarza

- Ramka **Kalendarz związany z datą początkową** domyślnie daje podgląd godzin pracy w ciągu tygodnia zawierającego wybraną **Datę początkową**.
- W polu **Uływ czasu** podana jest łączna liczba godzin pracy w wybranym okresie.

W polu **Uływ czasu** można wprowadzić czas trwania, aby wymusić ponowne wyliczenie pola **Data końcowa** zgodnie z podaną **Datą początkową**.

Rozdział 19 - Zarządzanie strefami czasowymi

Niniejszy rozdział wyjaśnia, w jaki sposób używać stref czasowych. Rozdział zawiera informacje na następujące tematy:

Wprowadzenie

- Dlaczego zarządzać strefami czasowymi?

Metodologia

- Wprowadzanie stref czasowych
- Tworzenie stref czasowych
- Zarządzanie strefą czasową
- Zarządzanie strefami czasowymi w AssetCenter Server

Wpływ

- Skutki dla różnych operacji

Dlaczego zarządzać strefami czasowymi?

Ponieważ niektóre stacje robocze i serwer bazy danych mogą się znajdować w różnych lokalizacjach geograficznych, AssetCenter zarządza strefami i różnicami czasowymi w odniesieniu do czasu uniwersalnego Greenwich (GMT). AssetCenter przestrzega następujących reguł:

- Wszystkie pola typu „Data+Czas” są wyświetlane przez stację roboczą z zachowaniem strefy czasowej tej stacji.
- Wszystkie pola typu „Data+Czas” są przechowywane na serwerze w odniesieniu do zdefiniowanej strefy czasowej.
- Wszystkie obliczenia związane z datami i czasem uwzględniają różnice związane ze strefami czasowymi.

Przykład

Weźmy przykład serwera znajdującego się w Nowym Jorku, na którym dane są zaindeksowane zgodnie ze strefą czasową Paryż (Francja) i dwie stacje robocze znajdujące się w Londynie i w Paryżu. Rozpocznijmy od zdefiniowania strefy czasowej dla każdej stacji roboczej zgodnie z czasem GMT:

- Strefa czasowa serwera = GMT-5
- Strefa czasowa stacji roboczej w Paryżu = GMT+1
- Strefa czasowa stacji roboczej w Londynie = GMT
- Strefa czasowa danych = GMT+1

Wszystkie wartości typu „Data+Czas” są zatem przechowywane na serwerze w formacie GMT+1, na stacji roboczej w Paryżu są wyświetlane jako GMT+1, zaś na stacji roboczej w Londynie jako GMT. Na przykład, przyjmując zlecenie pracy na stacji roboczej w Londynie, jeśli termin ostateczny rozwiązania zlecenia pracy jest ustawiony na 15 maja 1998 na godzinę 17:30, na innych stacjach roboczych wyświetlane jest to następująco:

- Na serwerze: 15 maja 1998 o 12:30
- Na stacji roboczej w Paryżu: 15 maja 1998 o 18:30
- Na stacji roboczej w Londynie; 15 maja 1998 o 17:30.

Wprowadzanie stref czasowych

Aby strefy czasowe były obsługiwane prawidłowo przez AssetCenter należy stosować następującą procedurę:

- Zdefiniuj strefy czasowe przy tworzeniu bazy danych w programie AssetCenter Database Administrator za pomocą opcji **Użyj stref czasowych**.
- Utwórz strefy czasowe (np. importując informacje dotyczące stref czasowych).
- Zdefiniuj strefę czasową twojej stacji roboczej za pomocą polecenia menu **Narzędzia/ Opcje**.
- Zdefiniuj kalendarze zgodnie ze strefami czasowymi.

Tworzenie stref czasowych




Systemy Windows 95 i Windows NT mogą obsługiwać strefy czasowe i automatycznie przełączają czas zimowy i letni. AssetCenter zarządza tymi informacjami jeszcze dokładniej. Uwzględnia on zmiany w regułach zmian czasu letniego i zimowego w poszczególnych latach, co pozwala wyświetlać czas lokalny obowiązujący w przeszłości z większą dokładnością. Sposób, w jaki AssetCenter korzysta z informacji o strefach czasowych pozwala na:

- Wyświetlanie lokalnych dat i czasu z uwzględnieniem czasu letniego i zimowego.
- Symulację przebywania w innej lokalizacji.
- Funkcjonalność ta nie jest używana w lokalizacjach, które nie zmieniają czasu zimowego na letni i odwrotnie.

16-bitowe wersje systemu Windows nie obsługują stref czasowych.

Aby uniknąć konieczności ręcznego definiowania stref czasowych, AssetCenter dostarczany jest z plikiem opisu zawierającym główne strefy czasowe. Plik ten można zaimportować w następujący sposób:

- Wybierz polecenie menu **Plik/ Import**. AssetCenter otwiera okno wyboru importu.

- Wybierz „Wykonaj skrypt” klikając . AssetCenter otwiera okno aktualizacji bazy danych. Kliknij opcję , aby wybrać skrypt do wykonania, w tym przypadku „tz.scr” w folderze podrzędnym **//datakit/standard** foldera instalacyjnego AssetCenter.
- Kliknij . AssetCenter wykonuje import zgodnie ze skrypcem.

Zarządzanie strefą czasową

W tej sekcji bardziej szczegółowo omówione zostało pole **Czas letni** (nazwa SQL: memDaylightInfo) :

- Format pola ‘Czas letni’
- Wartości argumentu <Year>
- Wartości argumentu <DaylightInfo>
- Przykład

Format pola ‘Czas letni’

Pole **Czas letni** (nazwa SQL: memDaylightInfo) ma następującą strukturę:

```
<Year>=<DaylightInfo>|<Year>=<DaylightInfo>|<Year>=<DaylightInfo>|...
```

W pozostałej części tej sekcji stosowane są następujące konwencje:

- <Year>=<DaylightInfo> razem zwane są „parametrem”
- <Year> i <DaylightInfo> osobno zwane są „argumentami”

W poniższej tabeli omówiono zmiany czasu letniego następujące zgodnie z wartościami argumentów <Year> i <DaylightInfo>.

	Argument <DaylightInfo> jest pusty	Argument <DaylightInfo> ma wartość
Argument <Rok> jest pusty („<Year>=” nie pojawia się)	Nie ma zmiany czasu letniego w całej strefie czasowej.	Informacja o czasie letnim jest ważna dla wszystkich lat, za wyjątkiem tych, które są zdefiniowane przez parametry z argumentem <Year>.
Argument <Year> ma wartość	Nie dotyczy	Informacja o czasie letnim dla tej strefy czasowej obowiązuje dla każdego roku, począwszy od roku określonego przez argument <Year> do następnego argumentu <Year>.

Wartości argumentu <Year>

Argument <Year> określający rok, od którego obowiązują informacje o zmianie czasu letniego zdefiniowane w argumencie <DaylightInfo>, może przybrać dowolną wartość będącą czterocyfrowym oznaczeniem roku (na przykład 1990, 1997, 1998, 2012).

Wartości argumentu <DaylightInfo>

Pełny format argumentu <DaylightInfo> jest następujący:

```
<StdShift>,<DltShift>,<SDay>,<SMonth>,<SDayPos>,<SHour>,<DDay>,<DMonth>,<DDayPos>,<DHour>
```

Argument ten składa się z szeregu argumentów podrzędnych omówionych poniżej:

Argument podrzędny	Opis	Możliwa wartość
<StdShift>	<p>Wyrażony w minutach, argument ten opisuje różnicę czasu pomiędzy standardowym czasem w obrębie strefy czasowej, a czasem w danej strefie czasowej.</p> <p>Na przykład, dla Paryża (strefa czasowa GMT+1), jeśli <StdShift> jest ustawiony na 30 (minut), standardowy czas w tej strefie czasowej to GMT+1h30min, a nie GMT+1h.</p>	<p>Domyślnie ten argument podrzędny jest pusty, jednak może być ustawiony na dowolną wartość numeryczną. Użytkownik musi sprawdzać spójność tego argumentu podrzędnego.</p>
<DltShift>	<p>Wyrażony w minutach, argument ten opisuje różnicę czasu pomiędzy czasem letnim, a czasem danej strefy czasowej.</p>	<p>Domyślnie, wartość tego argumentu podrzędnego jest ustawiona na 60 (co odpowiada 1 godzinie różnicy pomiędzy czasem letnim, a czasem odniesienia „GMT +”), jednak może być ustawiony na dowolną wartość numeryczną. Użytkownik musi sprawdzać spójność tego argumentu podrzędnego.</p>
<SDay>	<p>Dzień zmiany z czasu letniego na czas standardowy</p>	<p>„Poniedziałek” „Wtorek” „Środa” „Czwartek” „Piątek” „Sobota” „Niedziela”</p> <p>Puste (w tym przypadku, należy ustawić wartość argumentu <SdayPos> na wartość pomiędzy 1 i 31, aby określić dzień zmiany z czasu standardowego na czas letni)</p>
<SMonth>	<p>Miesiąc zmiany z czasu letniego na czas standardowy.</p>	<p>„Styczeń” „Luty” „Marzec”</p>

Argument podrzędny	Opis	Możliwa wartość
		„Kwiecień” ... „Listopad” „Grudzień”
<SDayPos>	Pozycja w miesiącu dnia zmiany z czasu letniego na czas standardowy.	„Pierwszy” „Drugi” „Trzeci” „Czwarty” „Ostatni” „Przedostatni” Wartość między 1 a 31, gdy argument <SDay> jest pusty
<SHour>	Godzina zmiany z czasu letniego na czas standardowy (wyrażona w czasie letnim).	Dowolna wartość wyrażona w formacie 24-godzinnym (GG:MM:SS).
<DDay>	Dzień zmiany z czasu standardowego na czas letni.	„Poniedziałek” „Wtorek” „Środa” „Czwartek” „Piątek” „Sobota” „Niedziela” Puste (w tym przypadku, należy ustawić wartość argumentu <DdayPos> na wartość pomiędzy 1 i 31, aby określić dzień zmiany z czasu letniego na czas na czas standardowy)
<DMonth>	Miesiąc zmiany z czasu standardowego na czas letni.	„Styczeń” „Luty” „Marzec” „Kwiecień” itd. „Listopad” „Grudzień”
<DDayPos>	Pozycja w miesiącu dnia	„Pierwszy”

Argument podrzędny	Opis	Możliwa wartość
	zmiany z czasu standardowego na czas letni.	„Drugi” „Trzeci” „Czwarty” „Ostatni” „Przedostatni” Wartość między 1 a 31, gdy argument <DDay> jest pusty
<DHour>	Godzina zmiany z czasu standardowego na czas letni (wyrażona w czasie standardowym).	Dowolna wartość wyrażona w formacie 24-godzinnym (GG:MM:SS).

Przykład

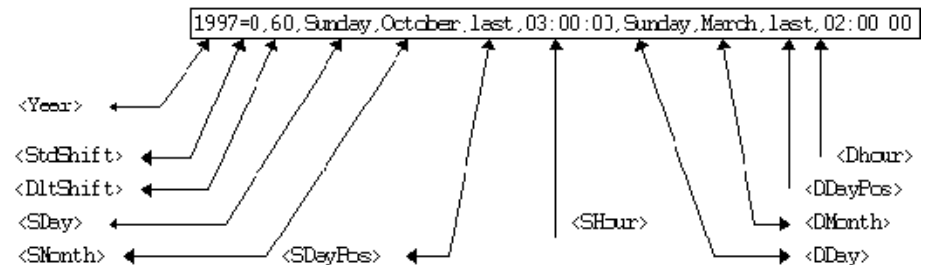
Jako przykład, weźmy informacje dotyczące czasu letniego „(GMT+01:00) w strefie czasowej Paryża, Madrytu, Amsterdamu”.

```
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
```

Zidentyfikujmy zastosowane parametry. Parametry oddzielone są znakiem „|”. W tym przypadku, zastosowane są dwa parametry:

```
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
```

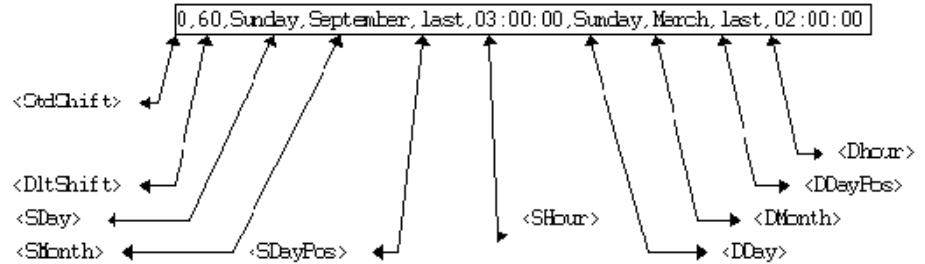
Pierwszy parametr



- <Year> = 1997 oznacza, że następujące parametry <DaylightInfo> znajdują zastosowanie w roku 1997 i następnych latach.

- <StdShift> = 0 oznacza, że nie ma różnicy pomiędzy strefą czasową i standardowym czasem w tej strefie czasowej.
- <DltShift> = 60 oznacza, że różnica czasu między czasem standardowym a czasem letnim wynosi 60 minut, to znaczy 1 godzinę. Czas letni jest zatem równy czasowi strefy czasowej plus jedna godzina.
- <SDay> = Niedziela oznacza, że przejście na czas standardowy ma miejsce w niedzielę.
- <SMonth> = Październik oznacza, że przejście na czas standardowy ma miejsce w październiku.
- <SDayPos> = Ostatni oznacza pozycję dnia w miesiącu. Tutaj przejście na czas standardowy ma miejsce w ostatnią niedzielę października.
- <SHour> = 03:00:00 oznacza, że przejście na czas standardowy ma miejsce o godzinie 03:00.
- <DDay> = Niedziela oznacza, że przejście na czas letni następuje w niedzielę.
- <DMonth> = Marzec oznacza, że przejście na czas letni następuje w marcu.
- <DDayPos> = Ostatni oznacza pozycję dnia w miesiącu. W tym przypadku, zmiana na czas letni ma miejsce w ostatnią niedzielę marca.
- <DHour> = 02:00:00 oznacza, że przejście na czas letni następuje o godzinie 02:00.

Drugi parametr



- Ponieważ nie ma argumentu `<Year>`, parametr ten określa, że obowiązuje tylko dla tych lat, które nie zostały opisane w poprzednim parametrze.
- `<StdShift> = 0` oznacza, że nie ma różnicy pomiędzy strefą czasową, a czasem standardowym w tej strefie czasowej. Czas zimowy jest tak sam, jak czas strefy czasowej.
- `<DltShift> = 60` oznacza, że różnica czasu pomiędzy czasem standardowym i czasem letnim wynosi 60 minut, to znaczy 1 godzinę. Zatem czas letni to czas strefy czasowej plus jedna godzina.
- `<SDay> = Niedziela` oznacza, że przejście na czas standardowy ma miejsce w niedzielę.
- `<SMonth> = Wrzesień` oznacza, że przejście na czas standardowy ma miejsce we wrześniu.
- `<SDayPos> = Ostatni` oznacza pozycję dnia w miesiącu. Tutaj przejście na czas standardowy ma miejsce w ostatnią niedzielę września.
- `<SHour> = 03:00:00` oznacza, że przejście na czas standardowy następuje o godzinie 03:00.
- `<DDay> = Niedziela` oznacza, że przejście na czas letni ma miejsce w niedzielę.
- `<DMonth> = Marzec` oznacza, że przejście na czas letni ma miejsce w marcu.
- `<DDayPos> = Ostatni` oznacza pozycję dnia w miesiącu. W tym przypadku przejście na czas letni ma miejsce w ostatnią niedzielę marca.
- `<DHour> = 02:00:00` oznacza, że przejście na czas letni następuje o godzinie 02:00.

W rezultacie:

Od roku 1997, zmiana na czas standardowy następuje w ostatnią niedzielę października o godzinie 03:00:00 (zegary cofane są na godzinę 02:00:00), a zmiana na czas letni następuje w ostatnią niedzielę marca o godzinie 02:00:00 (wskazówki zegarów przesuwane są do przodu, na godzinę 03:00:00).

Dla wszystkich lat przed 1997, zmiana na czas zimowy następowała w ostatnią niedzielę września o godzinie 03:00:00, a zmiana na czas letni następowała w ostatnią niedzielę marca o godzinie 02:00:00.

Zarządzanie strefami czasowymi w Serwer Asset Center

Serwer Asset Center pozwala konfigurować testy dotyczące stref czasowych. Wybierz polecenie menu **Opcje/ Konfiguruj**.

Testy do wykonania

W zakładce **Ogólne** ekranu konfiguracji, konfigurujesz typy testu strefy czasowej, które mają być wykonywane:

- **Weryfikuj strefę czasową serwera bazy danych.**
- **Weryfikuj czas lokalny w porównaniu z czasem serwera.**

Te dwa testy porównują czas serwera bazy danych z czasem komputera, na którym zainstalowany jest Serwer Asset Center. Różnica czasu jest wyrażona jako $[(n * 30\text{minut}) + m]$, gdzie m jest między -15 minut a +15 minut.

W obu przypadkach

Jeśli różnica czasu przekracza 5 minut, Serwer Asset Center oferuje aktualizację czasu lokalnego na komputerze, na którym jest zainstalowany.

Jeśli odrzucisz tę aktualizację (jeśli uważasz na przykład, że to czas serwera wymaga zmiany), połączenie zostaje odrzucone. Możesz połączyć

się ponownie, kiedy różnica między tymi dwoma czasami nie będzie już przekraczała 5 minut (w wyniku modyfikacji czasu serwera bazy danych oraz/lub komputera, na którym zainstalowany jest Serwer Asset Center).

Specyficzne aspekty opcji Weryfikuj strefę czasową serwera bazy danych

Jeśli jest to niezbędne, informacje dotyczące strefy czasowej serwera w tabeli opcji AssetCenter są aktualizowane (jeśli liczba (n * 30 minut) nie odpowiada strefie czasowej serwera).

Aby to zrobić, komputer, na którym uruchomiony jest Serwer Asset Center musi mieć prawidłowy czas oraz prawidłowe informacje dotyczące zmian czasu letniego.

Specyficzne aspekty opcji Weryfikuj czas lokalny w porównaniu z czasem serwera

Strefa czasowa serwera, niezbędna dla wewnętrznych operacji AssetCenter, zostaje odzyskana.

Częstotliwość testu

Test jest wykonywany:

- Po raz pierwszy, podczas gdy Serwer Asset Center nawiązuje połączenie z bazą danych.
- Następnie regularnie, zgodnie z harmonogramem zdefiniowanym w zakładce **Moduły** ekranu konfiguracji Serwer Asset Center.

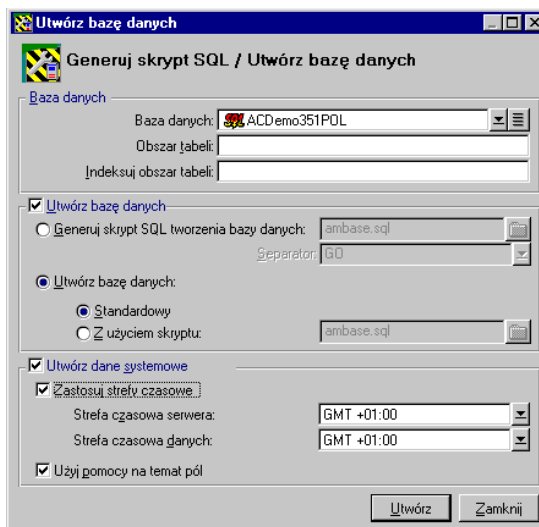
Skutki dla różnych operacji

Strefy czasowe mają wpływ na szereg operacji:

- Tworzenie bazy danych
- Połączenie z bazą danych
- Import i eksport

Tworzenie bazy danych

Przy tworzeniu bazy danych aplikacja AssetCenter Database Administrator pozwala zdefiniować opcje strefy czasowej. Wybierz polecenie menu **Działanie/ Utwórz bazę danych** w aplikacji AssetCenter Database Administrator. Ramka **Utwórz dane systemowe** zawiera opcje dotyczące stref czasowych.



Opcje tworzenia bazy danych

Pole wyboru **Używaj stref czasowych** określa, czy podczas tworzenia bazy danych uwzględniane są strefy czasowe.

- Jeśli pole to jest zaznaczone, przy tworzeniu bazy danych używane są strefy czasowe.
- Jeśli to pole nie jest zaznaczone, strefy czasowe są ignorowane podczas tworzenia stref czasowych.

Pola **Strefa czasowa serwera** i **Strefa czasowa danych** określają obowiązującą strefę czasową serwera w momencie tworzenia bazy danych, oraz strefę czasową, zgodnie z którą przechowywane będą dane.

Opcja ta jest dostępna tylko podczas tworzenia bazy danych. Pozwala ona zdefiniować referencję czasową serwera i danych. Jeśli zmienisz te wartości, wartości „Data+Czas” już znajdujące się w bazie danych nie będą miały sensu.

Połączenie z bazą danych

Przy nawiązywaniu połączenia z bazą danych, AssetCenter szuka w pliku „aam.ini” (znajdującym się w głównym folderze Windows) pozycji „LocalTimeZone”, która określa strefę czasową stacji roboczej.

Jeśli nie można znaleźć tej informacji, AssetCenter używa strefy czasowej systemu (zdefiniowanej w systemie Windows NT lub Windows 95).

Następnie AssetCenter szuka w bazie danych strefy czasowej zgodnej z pozycją „LocalTimeZone” w pliku „aam.ini” lub strefą czasową zdefiniowaną w systemie Windows.

W następniej tabeli zestawiono różne możliwe sytuacje:

Czy w pliku „aam.ini” jest „LocalTimeZone”?	Czy znaleziono zgodną strefę w tabeli stref czasowych?	Informacja zapisana w pliku „aam.ini” (pozycja „LocalTimeZone”)
Tak	Tak	Strefa czasowa bazy danych
	Nie	Bez zmian
Nie	Tak	Strefa czasowa bazy danych
	Nie	Strefa czasowa systemu

Dopasowanie czasu w stacji roboczej

Przy nawiązywaniu połączenia z odległą bazą danych, AssetCenter sprawdza poprawność czasu podanego przez zegar stacji roboczej w odniesieniu do zegara serwera.

AssetCenter wylicza strefę czasową zegara stacji roboczej i określa różnicę zegarową pomiędzy dwiema maszynami. Obliczenie to jest wykonywane w następujący sposób:

Różnica zegarowa to różnica synchronizacji, której nie należy mylić z różnicą czasową, która jest transpozycją w odniesieniu do różnicy stref czasowych.

```
Clock difference = Modulus((Difference in minutes between times of both machines concerned) / 30)
```

Moduł jest resztą z dzielenia.

Rozważmy następujące stacje robocze:

- Stacja robocza A jest w strefie czasowej GMT i wskazuje godzinę 18:02
- Stacja robocza B jest w strefie czasowej GMT+1 i wskazuje 18:19 (tzn. 17:19 dla stacji roboczej A, 43 minuty różnicy względem stacji roboczej A)

```
Clock differential = Modulus (43/30) = 13 minutes
```

Jeżeli różnica przekracza pięć minut (wartość stała), AssetCenter proponuje regulację zegara na poziomie stacji roboczej.

Jeżeli użytkownik odmówi, połączenie zostaje zakończone.

AssetCenter wykonuje to sprawdzenie okresowo, oraz gdy w stacji roboczej zostaje zmieniony czas. Domyślnie, sprawdzenie to jest wykonywane co 60 minut, ale można to zmienić modyfikując opcję „g_lTimeZoneCheckInMns” w pliku „aam.ini” (znajdującym się w głównym folderze systemu Windows), w sekcji [option].

```
[option]
g_lTimeZoneCheckInMns = 30
```

Częstotliwość sprawdzania różnicy zegarowej została ustawiona na 30 minut.

Częstotliwość tę można również skonfigurować za pomocą opcji **Weryfikuj strefę czasową serwera bazy danych** w AssetCenter.

Weryfikacja ta funkcjonuje wyłącznie, jeśli baza danych została utworzona z uwzględnieniem strefy czasowej.

Import i eksport

W przypadku tych dwóch funkcji, konwersja wykonywana jest z założeniem, że wszystkie pola „Data+Czas” są zgodne ze strefą czasową stacji roboczej wykonującej import lub eksport.

Rozdział 20 - Używanie AssetCenter jako serwera DDE

Zadaniem tej sekcji jest szczegółowy opis wywołań DDE rozpoznawanych przez AssetCenter, używanego jako serwer DDE.

W tej sekcji, po każdym opisie teoretycznym podano praktyczny przykład:

Uwagi ogólne na temat usług DDE

- Definicja serwera DDE
- Mechanizmy wywołań DDE

Polecenia DDE

- Wprowadzenie do poleceń DDE
- Polecenia globalne
- Polecenia związane z tabelą
- Polecenia związane z tabelą i polem lub łączem

Przykłady wywołań DDE

- Wprowadzenie do przykładów wywołań DDE
- Scenariusz 1: Wewnętrzne wywołania DDE programu AssetCenter
- Scenariusz 2: Wywołania DDE z aplikacji Excel 97
- Scenariusz 3: Tworzenie wywołań DDE za pomocą aplikacji Visual Basic 5.0

Definicja serwera DDE

DDE oznacza dynamiczną wymianę danych. Skrót ten jest nazwą mechanizmu wymiany danych pomiędzy aplikacjami systemu Windows. W opisanym przypadku, mechanizm DDE jest używany do uruchamiania poleceń AssetCenter z innej aplikacji

Mechanizmy wywołań DDE

Mechanizmy DDE oparte są na „usługach”. Aby uruchomić mechanizm DDE, trzeba zdefiniować „temat”, aby podać wymagany kontekst, w którym mają być uruchamiane „polecenia”. Ze względu na spójność, przy każdej zmianie kontekstu trzeba zakończyć poprzedni kontekst.

Niniejszy rozdział zawiera informacje na następujące tematy:

- Usługa DDE
- Temat DDE
- Polecenie DDE

Usługa DDE

W większości przypadków, „usługa” jest nazwą wykonywalnego obiektu załadowanego do pamięci. W tym przypadku, tzn. używając AssetCenter jako serwera DDE, usługa nosi nazwę „aam”.

Aby działały mechanizmy DDE, musi być uruchomiony program działający jako serwer DDE. W tym przypadku, do pamięci musi być wprowadzony program AssetCenter.

Temat DDE

Temat pozwala zdefiniować kontekst, w którym ma być wykonane działanie. W przypadku AssetCenter, tematem jest „AssetCenter”.

Polecenie DDE

Polecenia te wysyłane są do AssetCenter w celu wykonania. Można je podzielić na kilka grup:

- Polecenia globalne, dla których nie jest wymagana nazwa tabeli, ani nazwa pola.
- Polecenia związane z tabelą, wymagające nazwy SQL jako parametru, aby możliwe było ich wykonanie.
- Polecenia związane z tabelą i polem lub łączem, wymagające nazwy SQL tabeli i pola lub łącza jako parametru, aby możliwe było ich wykonanie.

Polecenia należące do tych grup mogą być dwóch rodzajów:

- „Execute”, pozwalające wykonywać zadania w AssetCenter.
- „Request”, pozwalające wysyłać do AssetCenter zapytania o informacje.

Jak znaleźć nazwę SQL tabeli, pola lub łącza

Po kliknięciu prawym przyciskiem myszy dowolnego pola tabeli, AssetCenter wyświetla menu kontekstowe. Po wybraniu polecenia menu **Konfiguruj obiekt**, okno wyświetlane przez AssetCenter podaje nazwę SQL i nazwę SQL obiektu (łącza lub pola), które zostało kliknięte.

Wprowadzenie do poleceń DDE

Kroki do wykonania

Aby polecenie DDE zostało wykonane prawidłowo, trzeba wykonać trzy kolejne kroki:

- Jasno zdefiniuj kontekst wykonania polecenia określając „Usługę” i „Temat”, który zostanie zastosowany. Przy korzystaniu z AssetCenter w roli serwera DDE, „tematem” zawsze jest „AssetCenter”.

Po zdefiniowaniu kontekstu, jest on używany we wszystkich następnych poleceniach DDE, aż do zdefiniowania nowego kontekstu.

- Uruchom polecenie. Dostępne są dwa typy poleceń.
 - ❖ **Execute:** <command>(<arguments>)
 - ❖ **Request:** <command>(<arguments>)
- Zamknij poprzednio zdefiniowany kontekst.

Szczegóły

Każda aplikacja systemu Windows ma swój własny sposób wysyłania poleceń DDE. W kolejnych sekcjach niniejszego podręcznika podano:

- Wyczerpującą listę poleceń DDE, które może przyjmować AssetCenter. Szczegółowo opisano składnię każdego z tych poleceń.
- Przykłady sterowania programem AssetCenter za pomocą mechanizmów DDE, ilustrujące zastosowanie tych poleceń. W każdym z tych przykładów zastosowano inny język programowania.

Polecenia globalne

Polecenia „globalne” są niezależne od danej tabeli, czy pola. W szczególności, nie jest wymagana nazwa SQL tabeli lub pola jako argument.

W tym rozdziale podano informacje o następujących poleceniach:

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

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

Typ działania

Execute (Wykonaj)

Opis

Nawiązuje połączenie z bazą danych za pomocą następujących parametrów:

<Cnx>

Argument ten może zawierać:

- Nazwę połączenia zdefiniowaną w AssetCenter (którą można znaleźć w pliku **AmDb.ini**).
- Pełną definicję połączenia z bazą danych zgodnie ze składnią opisaną poniżej:

```
[<Database engine>;<Location of database>;<Database engine login>;<Password of database engine>]
```

<User>

Argument ten zawiera nazwę użytkownika AssetCenter używaną do łączenia z bazą danych.

<Password>

Argument ten zawiera hasło związane z nazwą logowania (wartością argumentu „<User>”).

Przykłady

Następujące polecenie pozwala nawiązać połączenie z bazą danych Oracle, dla której już zostało zdefiniowane połączenie w AssetCenter. Połączenie to nosi nazwę „TDemo”. Używana jest w związku z nim nazwa logowania „Admin”. Hasło brzmi „password”.

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

Następujące polecenie wykonuje to samo połączenie bez korzystania z definicji w AssetCenter. Baza danych „TDemo” znajduje się na serwerze „Joshua”. Hasło połączenia Oracle brzmi „Root”.

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

Disconnect()

Typ działania

Execute (Wykonaj)

Opis

Odłącza AssetCenter od bieżącej bazy danych.

Przykład

Następujące polecenie kończy połączenie z bazą danych AssetCenter:

```
Execute:Disconnect()
```

ExecuteAction(<ActionName>)

Typ działania

Execute (Wykonaj)

Opis

Inicjuje działanie o nazwie „<ActionName>”.

<ActionName>

Argument ten zawiera nazwę działania zdefiniowaną w AssetCenter w polu **Nazwa** (nazwa SQL: Name) szczegółów działania.

Przykład

Następujące polecenie inicjuje działanie „Przypomnienie: zlecenie pracy nie zrealizowane”:

```
Execute:ExecuteAction(Reminder: work order not completed)
```

ListAllTables([Mask])

Typ działania

Request (Żądanie)

Opis

Podaje listę wszystkich tabel w bazie danych. Lista ta, składająca się z nazw SQL tabel, może być filtrowana za pomocą argumentu „<Mask>”.

<Mask>

Argument ten pozwala filtrować dane za pomocą następujących wzorców:

- Znak zapytania („?”) może być używany jako symbol zastępujący pojedynczy znak.
- Gwiazdka („*”) może być używana jako symbol zastępujący dowolny znak lub grupę znaków.

Przykłady

Następujące polecenie podaje listę nazw SQL wszystkich tabel w bieżącej bazie danych:

```
Request>ListAllTables()
```

Następujące polecenie podaje listę nazw SQL wszystkich tabel w bieżącej bazie danych, których nazwy SQL rozpoczynają się od „amA”:

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

Następujące polecenie podaje listę nazw SQL wszystkich tabel w bieżącej bazie danych zawierających literę „v”:

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

Następujące polecenie podaje listę nazw SQL wszystkich tabel w bieżącej bazie danych rozpoczynających się od „am”, których czwarta litera to „t”:

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

ListOpenTables([Mask])

Typ działania

Request (Żądanie)

Opis

Podaje nazwy SQL wszystkich otwartych tabel bazy danych. Listę tę można filtrować używając argumentu „<Mask>”.

<Mask>

Argument ten pozwala filtrować dane za pomocą następujących wzorców:

- Znak zapytania („?”) może być używany jako symbol zastępujący pojedynczy znak.
- Gwiazdka („*”) może być używana jako symbol zastępujący dowolny znak lub grupę znaków.

Przykłady

Następujące polecenie podaje nazwy SQL wszystkich otwartych tabel w bieżącej bazie danych:

```
Request:ListOpenTables()
```

Następujące polecenie podaje nazwy SQL wszystkich otwartych tabel w bieżącej bazie danych, których nazwa SQL rozpoczyna się od „amA”:

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

Załóżmy, że w AssetCenter otwarte są trzy następujące tabele: „amAsset”, „amAction”, „amProduct”. Poprzednie polecenie zwraca nazwę SQL dwóch tabel zaczynającą się od „amA”, tzn.: „amAsset” i „amAction”.

OpenView(<ViewName>)

Typ działania

Execute (Wykonaj)

Opis

Otwiera widok zdefiniowany w AssetCenter.

<ViewName>

Argument ten zawiera nazwę SQL widoku zdefiniowanego w AssetCenter.

Przykład

Następujące polecenie otwiera widok o nazwie „Zasoby wzięte w leasing”:

```
Execute:OpenView(Leased_assets)
```

Polecenia związane z tabelą

Polecenia te zależą od tabeli. Do ich wykonania wymagana jest nazwa SQL tabeli jako argument.

W niniejszym rozdziale podano informacje o następujących poleceniach:

- `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>)

Typ działania

Execute (Wykonaj)

Opis

Otwiera tabelę o nazwie SQL „<Table>”.

<Table>

Argument ten zawiera nazwę SQL otwieranej tabeli.

Przykład

Następujące polecenie otwiera tabelę zasobów (nazwa SQL: amAsset):

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

CloseTable(<Table>)

Typ działania

Execute (Wykonaj)

Opis

Zamyka tabelę poprzednio otwartą przez AssetCenter.

<Table>

Argument ten zawiera nazwę SQL zamykanej tabeli.

Przykład

Następujące polecenie zamyka tabelę zasobów (nazwa SQL: amAsset):

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

<Table>.GetRecordCount()

Typ działania

Request (Żądanie)

Opis

Zwraca liczbę rekordów w tabeli o nazwie SQL „<Table>”. Tabela, której dotyczy to polecenie musi być najpierw otwarta, aby to polecenie działało.

<Table>

Ten argument zawiera nazwę SQL tabeli, której liczbę rekordów chcemy poznać.

Przykład

Następujące polecenie zwraca liczbę rekordów w tabeli zasobów (nazwa SQL: amAsset):

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

<Table>.SetViewMode(<Mode>)

Typ działania

Execute (Wykonaj)

Opis

Definiuje tryb widoku otwartej tabeli.

<Table>

Argument ten zawiera nazwę SQL tabeli.

<Mode>

Dostępne są następujące wartości tego argumentu:

- „Arbo”: Rekordy w tabeli „<Table>” są wyświetlane w widoku drzewa.
- „List”: Rekordy w tabeli „<Table>” są wyświetlane w widoku listy.
- „ListOnly”: Wyświetlane są tylko rekordy w tabeli „<Table>”.
- „DetailOnly”: Wyświetlane są tylko szczegóły rekordu wybranego w tabeli „<Table>”.

- „ListDetail”: Wyświetla zarówno listę rekordów w tabeli „<Table>” oraz szczegóły rekordu wybranego z listy.

Przykład

Następujące polecenie przełącza tabelę zasobów (nazwa SQL: amAsset) na widok drzewa:

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

<Table>.SetRecordMode(<Mode>)

Typ działania

Execute (Wykonaj)

Opis



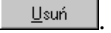
Definiuje tryb interakcji z rekordami z otwartej tabeli.






<Table>

Argument ten zawiera nazwę SQL tabeli.

<Mode>

Dostępne są następujące wartości tego argumentu:

- „New”: Rozpoczyna tworzenie nowego rekordu w tabeli „<Table>”. Polecenie to jest odpowiednikiem kliknięcia przycisku .
- „Duplicate”: Powiela wybrany rekord w tabeli „<Table>”. Polecenie to jest odpowiednikiem kliknięcia przycisku .
- „Delete”: Usuwa wybrany rekord z tabeli „<Table>”. Polecenie to jest odpowiednikiem kliknięcia przycisku .

- „Modify”: Potwierdza modyfikacje wprowadzone do wybranych rekordów w tabeli „<Table>”. Polecenie to jest odpowiednikiem kliknięcia przycisku .
- „Create”: Potwierdza utworzenie nowego rekordu w tabeli „<Table>”. Polecenie to jest odpowiednikiem kliknięcia przycisku .
- „CreateContinue”: Łączy tworzenie z powieleniem. Polecenie to jest odpowiednikiem kliknięcia przycisku .
- „Cancel”: Anuluje tworzenie nowego rekordu lub modyfikacje wprowadzone w wybranym rekordzie. Polecenie to jest odpowiednikiem kliknięcia przycisku .
- „Close”: Zamyka poprzednio otwartą tabelę „<Table>”. Polecenie to jest odpowiednikiem kliknięcia przycisku .

Przykład

Następujące polecenie otwiera tabelę zasobów (nazwa SQL: amAsset), rozpoczyna tworzenie nowego rekordu, następnie anuluje tworzenie:

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

<Table>.ListAllFields([Mask])

Typ działania

Request (Żądanie)

Opis

Zwraca nazwy SQL wszystkich pól w poprzednio otwartej tabeli „<Table>”.

<Table>

Argument ten zawiera nazwę SQL tabeli.

<Mask>

Argument ten pozwala filtrować dane według następujących wzorców:

- Znak zapytania („?”) zastępuje dowolny pojedynczy znak.
- Gwiazdka („*”) zastępuje znak lub grupę znaków.

Przykład

Następujące polecenie zwraca nazwy SQL wszystkich pól w tabeli zasobów:

```
Request:amAsset.ListAllFields
```

Następujące polecenie zwraca nazwy SQL wszystkich pól w tabeli zasobów (nazwa SQL: amAsset), których nazwa SQL rozpoczyna się od „se”:

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

<Table>.ListAllLinks([Mask])

Typ działania

Request (Żądanie)

Opis

Zwraca nazwy SQL wszystkich łączy w poprzednio otwartej tabeli „<Table>”.

<Table>

Argument ten zawiera nazwę SQL tabeli.

<Mask>

Argument ten pozwala filtrować dane według następujących wzorców:

- Znak zapytania („?”) zastępuje dowolny pojedynczy znak.
- Gwiazdka („*”) zastępuje znak lub grupę znaków.

Przykład

Następujące polecenie zwraca listę nazw SQL wszystkich łączy tabeli zasobów (nazwa SQL: amAsset):

```
Request:amAsset.ListAllLinks
```

Następujące polecenie zwraca listę nazw SQL wszystkich łączy tabeli zasobów (nazwa SQL: amAsset), której nazwa SQL zaczyna się od „se”:

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

<Table>.SetFilter(<Condition>)

Typ działania

Execute (Wykonaj)

Opis

Filtruje tabelę „<Table>” zgodnie z argumentem „<Condition>”.

<Table>

Argument zawiera nazwę SQL tabeli.

<Condition>

Argument ten zawiera warunek odnoszący się do polecenia. Ma on następującą ogólną postać:

```
(<Object><Relational operator><Value>)<Logical operator>(<Object><Relational operator><Value>)...
```

Przykład

Następujące polecenie filtruje tabelę zasobów (nazwa SQL: amAsset). Filtr ten wyświetla tylko te rekordy, które zostały zmodyfikowane przed 28/8/98 o godzinie 15:00:00:

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

<Table>.SetSelection(<Condition>)

Typ działania

Execute (Wykonaj)

Opis

Wybiera jeden lub więcej rekordów z uprzednio otwartej tabeli „<Table>”, zgodnie z argumentem „<Condition>”.

<Table>

Argument ten zawiera nazwę SQL tabeli.

<Condition>

Argument ten zawiera warunek odnoszący się do polecenia. Ma on następującą ogólną postać:

```
(<Object><Relational operator><Value>)<Logical operator>(<Object><Relational operator><Value>)...
```

Przykład

Następujące polecenie wybiera zasoby, których kod wewnętrzny jest większy lub równy „7”:

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

<Table>.GetSelectionId()

Typ działania

Request (Żądanie)

Opis

Zwraca listę identyfikatorów wybranych rekordów w tabeli „<Table>”.

<Table>

Argument ten zawiera nazwę SQL tabeli.

Przykład

Następujący przykład wybiera te rekordy w tabeli zasobów (nazwa SQL: amAsset), których kod wewnętrzny jest większy lub równy „7”, następnie zwraca listę identyfikatorów wybranych rekordów:

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

Polecenia związane z tabelą i polem lub łączem

Polecenia te są zależne od danego pola w tabeli. Aby zostały wykonane, wymagana jest nazwa SQL tabeli i nazwa SQL pola lub łącza w tej tabeli jako argument.

W niniejszym rozdziale omówiono następujące polecenia:

- <Table>:<Object>.GetValue()
- <Table>:<Object>.Highlight()
- <Table>:<Object>.RemoveLink()
- <Table>:<Object>.SetFocus()
- <Table>:<Object>.SetValue(<Value>)


- <Table>:<Link>.SetValueWhere(<Condition>)
- <Table>:<Object>.Show()

<Table>:<Object>.AddLink()

Typ działania

Execute (Wykonaj)

Opis

Symuluje kliknięcie przycisku  obszaru listy. Polecenie to pozwala dodać łącze do rekordu w tabeli.

<Table>

Parametr ten zawiera nazwę SQL tabeli, której dotyczy operacja.

<Object>

Parametr ten zawiera nazwę SQL obiektu, którego dotyczy operacja.

Przykład

Następujące polecenie dodaje wartość do listy pozycji.

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

<Table>:<Object>.GetValue()

Typ działania

Request (Żądanie)

Opis

Zwraca wartość obiektu „<Object>” (pola lub łącza) tabeli „<Table>” wybranego rekordu.

<Table>

Argument ten zawiera nazwę SQL tabeli.

<Object>

Argument ten zawiera nazwę SQL pola lub łącza tabeli „<Table>”, dla którego chcemy pobrać wartość.

Przykład

Następujące polecenie zwraca wartości pola **Pole1** (nazwa SQL: Field1) w tabeli zasobów (nazwa SQL: amAsset):

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

Następujące polecenie zwraca wartość łącza **Kategoria** (nazwa SQL: Category) tabeli zasobów (nazwa SQL: amAsset):

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

Następujące polecenie zwraca wartość łącza **Uwagi** (nazwa SQL: Comment) tabeli zasobów (nazwa SQL: amAsset):

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

<Table>:<Object>.Hilight()

Typ działania

Execute (Wykonaj)

Opis

Uaktywnia pole i zaznacza je.

<Table>

Parametr ten zawiera nazwę SQL tabeli, której dotyczy operacja.

<Object>

Parametr ten zawiera nazwę SQL pola, którego dotyczy operacja. Polecenie to nie działa w przypadku łączy.

Przykład

Następujące polecenie zaznacza pole **Kod kreskowy** (nazwa SQL: Barcode) w polu zasobów (nazwa SQL: amAsset):


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

<Table>:<Object>.RemoveLink()

Typ działania

Execute (Wykonaj)

Opis

Symuluje kliknięcie przycisku  obszaru listy. Polecenie to pozwala usunąć łącze do rekordu w tabeli.

<Table>

Parametr ten zawiera nazwę SQL tabeli, której dotyczy operacja.

<Object>

Parametr ten zawiera nazwę SQL obiektu, którego dotyczy operacja.

Przykład

Następujące polecenie usuwa wybraną wartość z listy pozycji.

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

<Table>:<Object>.SetFocus()

Typ działania

Execute (Wykonaj)

Opis

Uaktywia pole lub łącze „<Object>„ tabeli „<Table>” dla wybranego rekordu.

<Table>

Argument ten zawiera nazwę SQL tabeli zawierającej pole lub łącze, które chcemy uaktywnić.

<Object>

Argument ten zawiera nazwę SQL pola lub łącza tabeli „<Table>”, którą chcemy uaktywnić.

Przykład

Następujące polecenie uaktywia łącze **Kategoria** (nazwa SQL: Category) tabeli zasobów (nazwa SQL: amAsset):

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


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

Typ działania

Execute (Wykonaj)

Opis

Wypełnia pole „<Field>” w tabeli „<Table>”, dla wybranego rekordu, wartością „<Value>”.

<Table>

Argument ten zawiera nazwę SQL tabeli zawierającej pole, które chcemy wypełnić.

<Field>

Argument ten zawiera nazwę SQL pola w tabeli „<Table>”, które chcemy wypełnić.

<Value>

Argument ten zawiera wartość, którą chcemy przypisać do pola „<Field>” w tabeli „<Table>” wybranego rekordu. W przypadku pola typu „Data” lub „Data+Czas”, argument ten musi być wyrażony w międzynarodowym formacie daty (rr/mm/dd gg:mm:ss).

Przykład

Następujące polecenie przypisuje wartość „Test” do pola **Pole1** (nazwa SQL: Field1) w tabeli zasobów (nazwa SQL: amAsset) wybranego rekordu:

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

Następujące polecenie przypisuje wartość „28/08/98” do pola **Data instal.** (nazwa SQL: dInstall) w tabeli zasobów (nazwa SQL: amAsset) wybranego rekordu:

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

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

Typ działania

Execute (Wykonaj)

Opis

Wypełnia łączy „<Link>” w tabeli „<Table>” dla wybranego rekordu, zgodnie z warunkiem „<Condition>”.

<Table>

Argument ten zawiera nazwę SQL tabeli zawierające łączy, które chcemy wypełnić.

<Link>

Argument ten zawiera nazwę SQL łączy w tabeli „<Table>”, które chcemy wypełnić.

<Condition>

Argument ten pozwala określić rekord docelowy łączy. Ma on następującą ogólną postać:

```
(<Object><Relational operator><Value>)<Logical operator>(<Object><Relational operator><Value>)...
```

Przykład

Następujący przykład przypisuje wartość „Test” do łączy **Kategoria** (nazwa SQL: Category) tabeli zasobów (nazwa SQL: amAsset) wybranego rekordu. Kategoria „Test” musi istnieć, aby polecenie DDE zostało prawidłowo wykonane.

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

<Table>:<Object>.Show()

Typ działania

Execute (Wykonaj)

Opis

Uaktywnia pole lub łącze, które nie jest widoczne na ekranie. Tabela zawierająca pole lub łącze musi być otwarta.

<Table>

Ten parametr zawiera nazwę SQL tabeli, której dotyczy operacja.

<Object>

Parametr ten zawiera nazwę SQL obiektu, którego dotyczy operacja.

Przykład

Następujące polecenie uaktywnia pole **Kod księg.** (nazwa SQL: AcctCode) w tabeli zasobów (nazwa SQL: amAsset):

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

Wprowadzenie do przykładów wywołań DDE

Aby zilustrować ten zakres funkcjonalności, przygotowaliśmy kilka przykładowych scenariuszy.

- W pierwszym scenariuszu podano teoretyczny opis wywołań DDE.

- Drugi scenariusz jest praktycznym przykładem wywołań DDE w programie Excel. Przykład związany z tym scenariuszem napisano w języku Visual Basic for Applications.
- W trzecim scenariuszu przedstawiono aplikację napisaną w języku Visual Basic 5.0, która pozwoli zdobyć nieco praktycznego doświadczenia z wywołaniami DDE.

AssetCenter musi być uruchomiony, aby skorzystać z tych przykładów.

Scenariusz 1: Wewnętrzne wywołania DDE programu AssetCenter

Celem tego scenariusza jest określenie właściwych poleceń i argumentów DDE dla każdego wykonywanego działania. W scenariuszu tym omówiono wszystkie teoretyczne aspekty używania mechanizmów DDE; aspekty praktyczne poruszono w podanych przykładach.

W tym przykładzie utworzymy cechę typu „numerycznego” o nazwie „RAM”. Należy wykonać następujące działania:

- Otworzyć tabelę cech
- Wprowadzić nazwę cechy
- Wpisać tryb wprowadzania danych dla cechy
- Wprowadzić jednostkę
- Utworzyć cechę

Odbiór wywołania

Najpierw należy określić usługę i temat, konieczne do wykonywania poleceń DDE.

Jesteśmy w kontekście ogólnym AssetCenter, zatem:

- Usługa = „aam”
- Temat: „AssetCenter”

Teraz trzeba wykonać polecenie otwierające tabelę cech:

- Polecenie: `OpenTable()`

- Parametr: Nazwa SQL tabeli; w tym przypadku „amFeature”

Polecenie to ma następującą postać:

```
OpenTable (amFeature)
```

AssetCenter otwiera tabelę cech. Musimy teraz zainicjować tworzenie nowego rekordu w tej tabeli:

- Polecenie : SetRecordMode()
- Przedrostek polecenia: Nazwa SQL tabeli, tzn. „amFeature”
- Parametr: Tryb wprowadzania danych; w tym przypadku „New”

Polecenie to ma następującą postać:

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

Wprowadzanie danych

Następnie musimy podać programowi AssetCenter właściwe polecenia dla pól, które nas interesują:

- Pole **Nazwa** (nazwa SQL: TextLabel). Polecenie i jego argumenty są następujące:
 - ❖ Polecenie : **<Tabela>:<Obiekt>.SetValue (<Wartość>)**
 - * Argument „<Tabela>: Nazwa SQL tabeli, tzn. „amFeature”
 - * Argument „<Obiekt>”: Nazwa SQL pola, tzn. „TextLabel”
 - * Argument „<Wartość>”: Wartość pola, tzn. „RAM”

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

- Pole **Typ wpisu** (nazwa SQL: seDataType). Polecenie i jego argumenty są następujące:
 - ❖ Polecenie : **<Tabela>:<Obiekt>.SetValue (<Wartość>)**
 - * Argument „<Tabela>: Nazwa SQL tabeli, tzn. „amFeature”
 - * Argument „<Obiekt>”: Nazwa SQL pola, tzn. „seDataType”
 - * Argument „<Wartość>”: Wartość pola, tzn. „Numerical”

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

- Pole **Jednostka** (nazwa SQL: Unit). Polecenie i jego argumenty są następujące:
 - ❖ Polecenie: **<Tabela>:<Obiekt>.SetValue (<Wartość>)**

- * Argument „<Tabela>”: Nazwa SQL tabeli, tzn. „amFeature”
- * Argument „<Obiekt>”: Nazwa SQL pola, tzn. „Unit”
- * Argument „<Wartość>”: Wartość pola, tzn. „MB”

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

Tworzenie cechy

Aby utworzyć cechę, trzeba po prostu utworzyć rekord w tabeli cech:

- Polecenie : <Tabela>.SetRecordMode(<Tryb>)
 - ❖ Argument „<Tabela>”: Nazwa SQL tabeli, tzn. „amFeature”
 - ❖ Argument „<Tryb>”: Tryb tworzenia rekordu, tzn. „Create”

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

Scenariusz 2: Wywołania DDE z aplikacji Excel 97

Przykład ten bazuje na arkuszu Excel o nazwie „TestDDE.xls” który można znaleźć w folderze „amdemo” AssetCenter. Plik ten zawiera makra potrzebne do skorzystania z tego przykładu.

Przykład ten jest podobny do poprzedniego, z tą różnicą, że wywołania DDE mają tu inną postać, zgodną z językiem VBA (Visual Basic for Applications).

AssetCenter musi być uruchomiony, aby mogły zostać odebrane polecenia DDE wysłane przez Excel 97.

W niniejszej sekcji podano następujące informacje:

- Opis makra
- Kod źródłowy makra

Opis makra

Przejdź do arkusza „Data_entry”. Zawiera on tabelę z trzema kolumnami o następujących nazwach **Nazwa**, **Typ wpisu** i **Jednostka**, oraz przycisk (oznaczony „Utwórz cechę”), do którego przypisano makro o nazwie „Create”:

Trzeba tylko wprowadzić właściwe wartości w polach **Nazwa** (w tym przykładzie jest to komórka B6), **Typ wpisu** (komórka C6), **Jednostka** (komórka D6) i kliknąć przycisk „Utwórz cechę”, aby aplikacja Excel wysłała polecenie do AssetCenter, powodujące wykonanie następujących czynności:

- Otworzenie tabeli cech
- Wprowadzenie nazwy cechy
- Wpisanie typu wprowadzania danych dla cechy
- Wprowadzenie jednostki
- Utworzenie cechy

Kod źródłowy makra

Poniżej znajduje się kod źródłowy makra wykonującego wywołania DDE. Więcej informacji o wywołaniach DDE w aplikacji Excel lub Word i składni języka VBA znajduje się w pomocy ekranowej tych aplikacji.

```
Sub CreateFeature()  
Set Label = Worksheets("Date_entry").Range("B6")  
Set Type = Worksheets("Date_entry").Range("C6")  
Set Unit = Worksheets("Date_entry").Range("D6")  
Contexte = Application.DDEInitiate(app:= "aam", topic:= "AssetCenter")  
Application.DDEExecute Context, "OpenTable(amFeature)"  
Application.DDEExecute Context, "amFeature.SetRecordMode(New)"  
Application.DDEExecute Context,  
„amFeature:TextLabel.SetValue("+Label+")"  
Application.DDEExecute Context,  
"amFeature:seDataType.SetValue("+Type+")"  
Application.DDEExecute Context, "amFeature:Unit.SetValue("+Unit+")"  
Application.DDEExecute Context, "amFeature.SetRecordMode(Create)"  
Application.DDETerminate Context  
End Sub
```

Scenariusz 3: Tworzenie wywołań DDE za pomocą aplikacji Visual Basic 5.0

W tym scenariuszu przedstawiono program narzędziowy, który pozwala eksperymentować z mechanizmami wywołań DDE za pomocą prostego interfejsu graficznego. Jest to również dobra ilustracja mechanizmów DDE w języku Visual Basic 5.0.

Aby zapoznać się z tym przykładem, uruchom program **DDE TestCenter.exe**, który można znaleźć w folderze **AmDemo** AssetCenter. Program ten pozwala wykonywać polecenia DDE typu „Execute” i „Request”.

AssetCenter musi być uruchomiony, aby możliwe było odbieranie polecenia DDE wysłanego przez program BASIC.

Rozdział ten zawiera informacje na następujące tematy:

- Przed rozpoczęciem
- Wykonywanie programu
- Kod źródłowy programu

Przed rozpoczęciem

Instalacja

Aby używać tego programu, w komputerze musi być prawidłowo zainstalowany program Visual Basic 5.0. W szczególności, muszą być prawidłowo zainstalowane pewne kontrolki ActiveX. Jeśli program DDE TestCenter zwróci błąd, taki jak „Control XXXX is not registered”, wykonaj następujące czynności:

- Znajdź kontrolkę w komputerze i przejdź do jej foldera.
- Uruchom następujące polecenie

```
regsvr32 XXXX
```

- Uruchom ponownie program DDE TestCenter. Jeśli powyższa metoda nie rozwiązała problemu, dalszych informacji szukaj w dokumentacji programu Visual Basic 5.0.

Zalecenia

Dla najefektywniejszego wykorzystania tego przykładu, zalecamy:

- Uruchomienie AssetCenter, następnie zredukowanie rozmiaru okna aplikacji do około połowy ekranu.
- Uruchomienie programu DDE TestCenter.exe i przesunięcie okna aplikacji, tak aby znajdowało się obok okna AssetCenter.

Dzięki temu będzie można obserwować wyniki poleceń wysłanych przez aplikację DDE TestCenter.exe bezpośrednio w AssetCenter.

Składnia

Ten przykład jest podobny do poprzedniego, z tym, że wywołania DDE mają inną postać, zgodną ze standardem Visual Basic standard.


Wykonywanie programu

Polecenie DDE typu „Execute”

Przejdź do zakładki “Execute” (Wykonaj):

Wpisz polecenie do wykonania w polu „Command” (Polecenie). Zastosuj następującą składnię:

```
Command=<command> (<arguments>)
```

Kliknij przycisk , aby wykonać polecenie. Wszystkie błędy wyświetlane są w polu „Last DDE Error”.

Przykład 1:

Następujące polecenie typu „Execute” otwiera tabelę cech:

```
OpenTable (amFeature)
```

Przykład 2:

Następujące polecenia typu „Execute” otwierają tabelę budżetów, tworzą nowy rekord i wypełniają pole „nazwa” (nazwa SQL: Name) w oknie szczegółów. Polecenia te należy wykonać w podanej kolejności:


```
OpenTable (amBudget)
amBudget.SetRecordMode (New)
amBudget.Name.SetValue („Test“)
```

Polecenie DDE typu „Request”

Przejdź do zakładki „Request”:

Wpisz polecenie do wykonania w polu „Command”. Zastosuj następującą składnię:

```
Command=<command> (<arguments>)
```

Kliknij przycisk , aby wykonać polecenie. Wyniki zapytania są wyświetlane w polu „Request Result”. Wszelkie błędy są wyświetlane w polu „Last DDE Error”.

Przykład 1:

Następujące polecenie typu „Request” wyświetla listę nazw SQL wszystkich tabel bieżącego połączenia:

```
ListAllTables ()
```

Przykład 2:

Następujące polecenie typu „Request” wyświetla listę wszystkich nazw SQL wszystkich pól w uprzednio otwartej tabeli cech (nazwa SQL: amFeature):

```
amFeature.ListAllFields ()
```

Kod źródłowy programu

Kod źródłowy programu opatrzony komentarzami, w postaci projektu programu Visual Basic 5.0, można znaleźć w folderze **AmDemo** AssetCenter.

Rozdział 21 - Pola wyliczane

Użyj polecenia menu Narzędzia/ Administracja/ Pola wyliczane, aby uzyskać dostęp do ekranu tworzenia pól wyliczanych.

Niniejszy rozdział zawiera informacje na następujące tematy:

Wprowadzenie

- Definicja pola wyliczanego
- Użyteczność pól wyliczanych

Tworzenie pola wyliczanego

- Tworzenie pola wyliczanego
- Wprowadzenie

Zastosowanie

- Używanie pól wyliczanych

Definicja pola wyliczanego

Pole wyliczane jest to pole, którego wartość jest wyliczana zgodnie z wartością innych pól i zmiennych, za pomocą wzoru definiowanego przez użytkownika. Istnieją trzy rodzaje pól wyliczanych:

- AQL
- BASIC

- Pole wyliczane

Każdy z tych typów opiera się na innym języku formuł wyliczających wartość pól i ma wpływ na możliwości oraz ograniczenia związane z używaniem danego pola. Na przykład, w filtrach mogą być używane tylko pola wyliczane typu „AQL”.

Pola wyliczane to pola wirtualne, z parametrem tylko do odczytu (w bazie danych przechowywany jest sam wzór).

Można utworzyć dowolną ilość pól wyliczanych i przypisywać do nich uprawnienia użytkownika.

Użyteczność pól wyliczanych

Pola wyliczane pozwalają definiować dodatkowe informacje i wyliczać informacje syntetyczne dotyczące rekordów tabeli w bazie danych AssetCenter. Dzięki temu, oprócz kilku różnic, przypominają one „klasyczne” pola bazy danych:

- W odróżnieniu od pól „klasycznych”, wartość pola wyliczanego nie jest przechowywana w bazie danych AssetCenter.
- Wartość pola wyliczanego nie jest wypełniana przez użytkownika; jest ona podawana przez wzór.
- Nie można skojarzyć pola wyliczanego z pojedynczym rekordem danego pola. Tak jak wszystkie inne „klasyczne” pola bazy danych, pole wyliczane jest skojarzone ze wszystkimi rekordami tabeli i posiada wartość (która może być zerowa) dla każdego rekordu w tabeli.
- Pola wyliczane nie są wyświetlane w oknie szczegółów rekordu. Mogą one być wyświetlane w oknie listy.
- Pola wyliczane można stosować jedynie w wyliczaniu wartości domyślnych pól standardowych, jeśli są to pola typu **Łańcuch wyliczany** lub **Skrypt podstawowy**.

Tworzenie pola wyliczanego

Przed utworzeniem pola tego typu, warto się zapoznać ze specyficznymi szczegółami właściwymi dla każdego typu pola wyliczanego.

W niniejszej sekcji omówiono następujące tematy:

- Wprowadzenie
- Metodologia tworzenia pól wyliczanych

Wprowadzenie

Każdy typ pola wyliczanego ma inne właściwości, które określają jego zastosowanie.

W następnej tabeli podsumowano główne różnice występujące pomiędzy trzema typami:

Typ pola	Właściwości pól tego typu			Pole wyliczane przez	Cechy języka, na których opiera się wzór wyliczenia	
	Mogą być wyświetlane	Mogą być sortowane	Mogą być używane w filtrach		Zalety	Wady
AQL	Tak	Tak	Tak	Serwer bazy danych	Duże możliwości Zintegrowany edytor	Ograniczony język (na przykład: Brak prostych testów takich jak „IF”, „THEN”, „ELSE”). Pól tego typu nie można stosować w wartościach domyślnych.
Wyliczane ciągi znaków	Tak	Tak	Nie	Klient	Prostota	Ograniczone możliwości (proste łączenie ciągów znaków i wartości pól, lub tylko ciągów znaków).
BASIC	Tak	Nie	Nie	Klient	Wiele możliwości Elastyczny	Pola tego typu można jedynie wyświetlać.

Z powyższej tabeli wyraźnie wynika, że pola typu „AQL” mają znacznie szerszy zakres zastosowania niż dwa pozostałe typy pól wyliczanych.

Zapytanie AQL może korzystać ze wszystkich trzech właściwości (można je wyświetlać, sortować, stosować w filtrach):

Właściwość	Związane argumenty AQL
Można wyświetlać	Warunek SELECT
Można sortować	Warunki SELECT, ORDER BY, GROUP BY
Można używać w filtrach	Warunki SELECT, ORDER BY, GROUP BY, WHERE, HAVING

Więcej informacji na temat zapytań AQL znajduje się w podręczniku pt. „Opis funkcji systemu AssetCenter: **Administracja i** zaawansowane zastosowania AssetCenter”, rozdział „Pisanie zapytań w AQL”.

Wyliczanie przez serwer / stację roboczą

W przypadku pola typu „AQL”, serwer bazy danych przeprowadza wymagane obliczenia i wysyła wynik do stacji roboczej. Z tego powodu nie ma wpływu na pracę stacji roboczej, a obciążenie sieci jest niższe. Z drugiej strony, zapytania SQL wysyłane do bazy danych są bardziej skomplikowane.

Metodologia tworzenia pól wyliczanych

W rozdziale tym szczegółowo opisano metody używane do tworzenia pól wyliczanych.

Przeanalizuj potrzeby

Wybierz rodzaj pola w oparciu o dwa poniższe kryteria:

- Właściwości pola danego typu: Czy mogą być wyświetlane, sortowane, używane w filtrach lub wartości domyślne.
- „Koszt” tego rozwiązania; w sensie złożoności zastosowanego wzoru w porównaniu z możliwościami zastosowania. Trzy rodzaje pól wyliczanych można sklasyfikować następująco (pod względem wzrastającej złożoności):
 - ❖ Ciąg wyliczany
 - ❖ AQL
 - ❖ BASIC

Wszędzie tam, gdzie to możliwe, zalecamy stosowanie najmniej „kosztownego” rozwiązania.

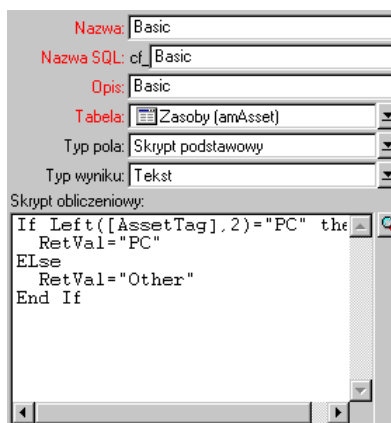
Na przykład:

- Jeśli pole służy tylko do celów informacyjnych, wystarczy pole wyliczone typu BASIC.
- Jeśli chcesz sortować rekordy według wartości pola, należy zastosować typ „AQL” lub „Ciąg wyliczany”.
- Jeśli chcesz filtrować rekordy według wartości pola, należy zastosować pole typu „AQL”.

Po określeniu potrzeb, możesz przejść do następnego kroku.

Otwórz okno tworzenia pola

Wybierz polecenie menu **Narzędzia/ Administracja/ Pola wyliczane**. AssetCenter wyświetli okno tworzenia pola wyliczanego:



Nazwa: Basic

Nazwa SQL: cf_Basic

Opis: Basic

Tabela: Zasoby (amAsset)

Typ pola: Skrypt podstawowy

Typ wyniku: Tekst

Skrypt obliczeniowy:

```
If Left([AssetTag], 2) = "PC" then
  RetVal = "PC"
Else
  RetVal = "Other"
End If
```

Szczegóły pola wyliczanego

Określ pole wyliczane

Najpierw wypełnij górną część okna, aby unikalnie zidentyfikować pole wyliczane:

- Pole **Nazwa** (nazwa SQL: Label) zawiera etykietę pola wyliczanego, używaną w nagłówkach kolumn list.
- Pole **Nazwa SQL** (nazwa SQL: SQLName) zawiera nazwę SQL pola wyliczanego. Nazwa ta, poprzedzona znakami „cf_”, jest na przykład używana przy odwoływaniu się do tego pola w skryptach typu BASIC, zapytaniach lub filtrach.

Nie można modyfikować nazwy SQL pola po jego utworzeniu. Wszelkie odwołania do tego pola przy użyciu poprzedniej nazwy SQL będą nieprawidłowe.


- Pole **Opis** (nazwa SQL: Description) zawiera krótki opis pola, używany w listach wyświetlających pole (na przykład w filtrach lub w oknie konfiguracji listy).

Zdefiniuj kontekst zastosowania pola

Pola **Tabela** (nazwa SQL: TableName) i **Typ pola** (nazwa SQL: seType) pozwalają zdefiniować kontekst zastosowania pola wyliczanego:

- Pole **Tabela** (nazwa SQL: TableName) pozwala skojarzyć pole wyliczane z tabelą. Pole będzie dostępne tylko w tej tabeli.
- Pole **Typ pola** (nazwa SQL: seType) pozwala określić typ pola wyliczanego. W zależności od typu, właściwości pola (wyświetlanie, sortowanie lub zastosowanie w filtrach) będą różne.
- Pole **Typ wyniku** pozwala określić typ wyniku pola wyliczanego. Typ ten jest używany dla potrzeb wyświetlania i formatowania. Pole wyliczane, którego typ wyniku jest data, jest zatem wyświetlane w taki sam sposób, jak wszystkie inne pola typu „Data” w bazie danych.

Wpisz wzór wyliczania pola

Teraz trzeba tylko wpisać wzór wyliczania pola. Można go wpisać bezpośrednio w polu tekstowym w dolnej części okna (zauważ, że etykieta tego pola zmienia się zgodnie z wybranym typem), lub kliknąć przycisk , lub nacisnąć przycisk „F4”, aby uzyskać dostęp do okna edytora.

Używany język różni się w zależności od typu pola.


Więcej informacji o dostępnych językach pisania wzorów można znaleźć w następującej dokumentacji:

- Podręcznik pt. „Opis funkcji systemu AssetCenter: Administracja i zaawansowane zastosowania AssetCenter”, rozdział „Wykorzystanie skryptów” dla języka Basic. Używana funkcja to **RetVal()**.
- Podręcznik pt. „Opis funkcji systemu AssetCenter: Administracja i zaawansowane zastosowania AssetCenter”, rozdział „Pisanie zapytań w AQL” dla języka AQL.
- Podręcznik pt. „Opis funkcji systemu AssetCenter: Administracja i zaawansowane zastosowania AssetCenter”, rozdział „Struktura bazy danych AssetCenter”, sekcja „Opis tabel”, punkt „Ciągi znakowe opisu tabel” dla ciągów wyliczanych.

Zdefiniuj uprawnienia użytkownika dla pola wyliczanego

Wybierz polecenie menu **Narzędzia/ Administracja/ Uprawnienia użytkownika**. AssetCenter wyświetli okno tworzenia uprawnień użytkownika.

Pola wyliczane są dostępne tylko do odczytu.

- Wprowadź krótki opis uprawnienia użytkownika w polu **Opis** (nazwa SQL: Description) i, opcjonalnie, komentarz w polu **Uwagi** (nazwa SQL: Comment).
- Rozwiń strukturę drzewa tabeli związanej z polem wyliczonym. Gałąź oznaczona ikoną  zawiera pełną listę pól wyliczanych dla danej tabeli.
- Następnie wybierz pole, dla którego chcesz edytować uprawnienia użytkownika. Pole wyboru „Read” pozwala zdefiniować uprawnienia do odczytu dla tego pola. Jeśli to pole jest zaznaczone, tylko profile z tym uprawnieniem użytkownika mogą przeglądać pole wyliczane. Jeśli pole to nie jest zaznaczone, wszyscy użytkownicy będą mieli dostęp (tylko do odczytu) do tego pola.

Używanie pól wyliczanych

Zastosowanie pola wyliczanego zależy od jego typu. Trzeba się upewnić, czy typ pola jest zgodny z jego zamierzonym zastosowaniem. W listach wyświetlających pola (tworzenie filtra, konfiguracja listy, itp.), AssetCenter przychodzi z pomocą, wyświetlając tylko te pola, które mogą być zastosowane.

Używanie pola wyliczanego w konfiguracji listy

Można wyświetlić wartość pola wyliczanego dla wszystkich rekordów w tabeli za pomocą polecenia **Konfiguruj listę** z menu kontekstowego. W liście pól i łączy tabeli, rozwiń gałąź oznaczoną „pola wyliczane”. Gałąź ta wyświetla listę dostępnych pól wyliczanych. AssetCenter wyświetla pola wyliczane zgodnie z formatem zdefiniowanym w **Narzędzia/ Opcje**, w zakładce **Wyświetlaj**:

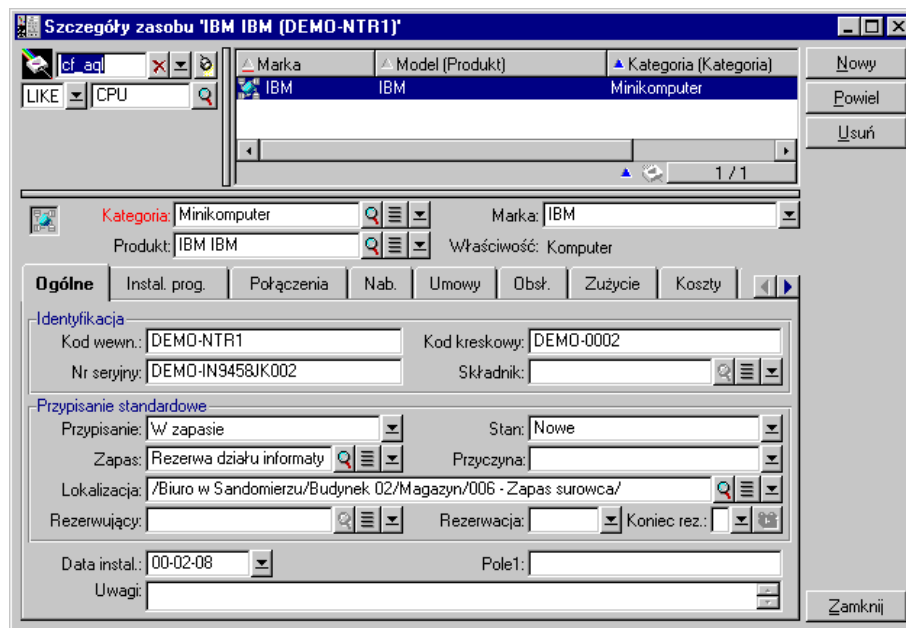
Można dodać je do listy tak samo, jak każde inne pole.

Filtrowanie rekordów tabeli

AssetCenter może filtrować rekordy tabeli zgodnie z wartością pola wyliczanego typu „AQL”. Aby to zrobić, wybierz polecenie **Prosty filtr** z menu kontekstowego i przejdź do gałęzi „Pola wyliczane”. AssetCenter wyświetli tylko pola typu „AQL”.

Odwoływanie się do pola wyliczanego

Nazwa SQL, poprzedzona znakami „cf_” służy do odwoływania się do pól wyliczanych. Poniższa kopia ekranu ilustruje zastosowanie nazwy SQL pola wyliczanego w filtrze:



Dodatkowo, do pól wyliczanych mogą się odwoływać różne moduły lub funkcje AssetCenter:

- AssetCenter Web
- AssetCenter APIs
- Raporty
- Formularze

Więcej informacji o modułach i zakresach funkcjonalności można znaleźć we właściwej dokumentacji.

Rozdział 22 - Wprowadzenie do kreatorów

Użyj polecenia menu Narzędzia/ Działania/ Edytuj, a następnie wybierz działanie typu Kreator, aby uzyskać dostęp do kreatorów

W niniejszym rozdziale omówiono następujące tematy:

Wprowadzenie

- Definicja kreatora
- Dla kogo przeznaczone są kreatory?

Klasyfikacja kreatorów

- Kreatory wymieniające informacje z bazą danych AssetCenter
- Kreatory niezależne

Definicja kreatora

Kreatory AssetCenter zostały zaprojektowane, aby pomóc w wykonywaniu prostych i często powtarzających się zadań. Za pomocą graficznego, przyjaznego dla użytkownika interfejsu, krok po kroku prowadzą one przez kolejne etapy konieczne do wykonania danego zadania. Wraz z AssetCenter dostarczono kilka gotowych kreatorów, które umożliwiają między innymi:

- Przenoszenie użytkowników i zasobów z jednej lokalizacji do innej. Należy wybrać użytkownika (pracownika), właściwie przypisane zasoby i nową lokalizację. Na podstawie tych danych kreator aktualizuje lokalizację zasobów i ich użytkownika.
- Proste zarządzanie zapasami. Użytkownik wybiera zasoby w zapasie, pracownika i lokalizację. Kreator przypisuje zasoby do wybranego pracownika i lokalizacji.
- Proste zbieranie informacji do wykonania danego działania.
- Ułatwione wprowadzanie rekordów.

Oprócz kreatorów dostarczonych z AssetCenter, możesz tworzyć własne.

Dla kogo przeznaczone są kreatory?

Kreatory są bardzo przydatne, zarówno dla początkujących, jak i zaawansowanych użytkowników:

- Początkujący użytkownicy mogą wykonywać skomplikowane zadania bez znajomości złożonych mechanizmów AssetCenter.
- Zaawansowani użytkownicy mogą tworzyć swoje własne kreatory lub modyfikować istniejące, aby dostosować je do potrzeb przedsiębiorstwa.

Klasyfikacja kreatorów

Kreatory można podzielić na dwie główne grupy:

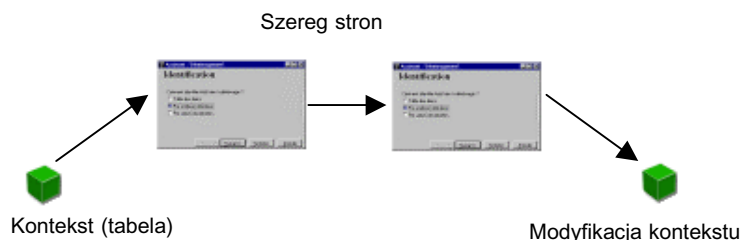
- Kreatory wymieniające informacje z bazą danych AssetCenter
- Kreatory niezależne

Kreatory wymieniające informacje z bazą danych AssetCenter

Istnieją dwie odmiany tego typu kreatorów:

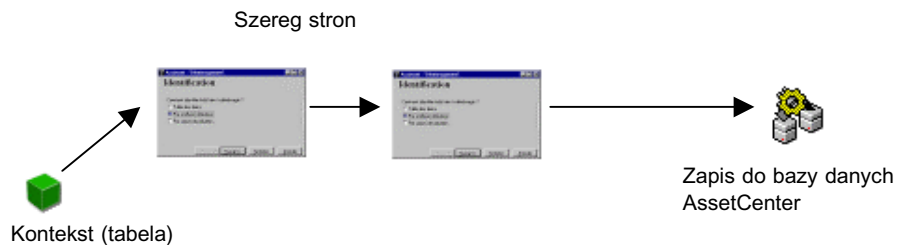
Kreatory wprowadzania danych

Kreatory te ułatwiają wprowadzanie danych i tworzenie rekordów wypełniając ekrany za użytkownika. Na przykład, kreator „Nowy użytkownik” prowadzi użytkownika przez tworzenie rekordu w tabeli wydziałów i pracowników. Kreator zbiera informacje dotyczące pracownika i tworzy właściwy rekord. Użytkownik nie musi bezpośrednio wprowadzać informacji na ekranie szczegółów. Tym zajmuje się kreator.



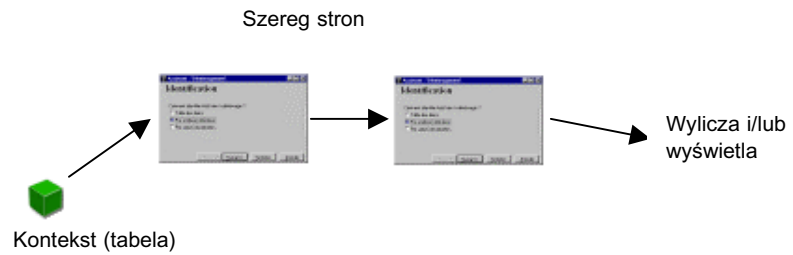
Kreatory zbierania danych

Do wykonania niektórych działań potrzebna jest wartość z bazy danych AssetCenter lub wartość zmiennej. Kreatory te zbierają i wyświetlają wymagane informacje. Na przykład, kreator „Przenieś” zbiera informacje o zasobach, które zostaną przeniesione, użytkownika tych zasobów i ich nowej lokalizacji. Następnie informacja ta jest wykorzystana do odpowiedniej modyfikacji danych w bazie danych AssetCenter.



Kreatory niezależne

Kreatory niezależne służą w zasadzie do wykonywania obliczeń i wyświetlania informacji. Można, na przykład, utworzyć kreator „Suma” podający sumę dwóch wartości wprowadzonych przez użytkownika.



Rozdział 23 - Tworzenie kreatorów

AssetCenter pozwala tworzyć własne kreatory i dostosowywać istniejące kreatory do własnych potrzeb. Kreatory są przechowywane jako pola tekstowe (**Skrypt kreatora** (nazwa SQL: WizardScript), zakładka **Kreator** szczegółów działania typu **Kreator**). Tworzenie kreatora polega na wprowadzeniu jego kodu bezpośrednio w tym polu lub zastosowaniu edytora graficznego. Czynność ta wymaga znajomości struktury kreatorów i języka skryptowego stosowanego do opisu tej struktury.

W niniejszym rozdziale omówiono następujące tematy:

Konwencje i definicje

- Zastosowane konwencje
- Definicje

Omówienie

- Model struktury
- Model strony kreatora

Struktura kreatorów

- Uwagi ogólne
- Ogólna struktura i składnia
- Właściwości węzła
- Funkcje języka Basic

Węzeł 'Root'

- Definicja węzła 'Root'
- Składnia węzła 'Root'
- Właściwości węzła 'Root'
- Węzły podrzędne węzła 'Root'

Węzeł 'Page'

- Definicja węzła 'Page'
- Składnia węzła 'Page'
- Właściwości węzła 'Page'
- Węzły podrzędne węzła 'Page'

Węzeł 'Transition'

- Definicja węzła 'Transition'
- Składnia węzła 'Transition'
- Właściwości węzła 'Transition'
- Specyfika węzła 'Transition'

Węzeł 'Finish'

- Definicja węzła 'Finish'

Węzeł 'Start'

- Definicja węzła 'Start'

Węzeł 'Control'

- Definicja węzła 'Control'
- Ogólna składnia węzła 'Control'
- Typy formantów i ich właściwości

Tworzenie kreatorów

- Przykład tworzenia kreatora
- Korzystanie z edytora graficznego
- Często zadawane pytania

Zastosowane konwencje

Do opisu struktury kreatorów zastosowano następującą notację:

[]	Nawiasy kwadratowe oznaczają odwołanie do wartości pola w bazie danych (w przypadku kreatorów kontekstowych) lub jednego z „pól specjalnych”: „CurrentSelection” i „CurrentTable”.
< >	Nawiasy ostre oznaczają wartości właściwości opisane w języku naturalnym. Nie należy ich jednak wpisywać. Nawiasy ostre oraz tekst w nie ujęty należy zastąpić właściwymi danymi.
	Znak „ ” służy jako separator możliwych wartości właściwości. Służy również do rozdzielania tytułów i wartości list wielokolumnowych.
{ }	W nawiasy klamrowe ujmuje się definicję węzła lub blok skryptu właściwości obejmujący wiele linii. Służą one również jako referencja do wartości właściwości kreatora.
'	W przykładach kodu, apostrof oznacza wiersz komentarza, który nie jest interpretowany przez AssetCenter.

Definicje

Poniżej podano definicje terminologii stosowanej w opisie struktury kreatorów.

Twip

„Twip” to jednostka wielkości i domyślnie odległości używana w kreatorach. Jest ona niezależna od rozmiaru ekranu. Ma ona następujące przeliczniki:

- 1440 „twipów” to jeden cal.

- 567 „twipów” to jeden centymetr.
- W rozdzielczości 96 dpi (standardowa rozdzielczość Windows) 15 twipów jest równe 1 pikselowi.

Formant

Formant oznacza element graficzny umożliwiający edycję elementu danych. Typowe formanty to pola wyboru, pola tekstowe, przyciski, listy rozwijane, itp.

Węzeł

Węzeł jest odpowiednikiem poziomu hierarchicznego w strukturze drzewa kreatora. Węzeł podrzędny danego węzła „N” znajduje się o jeden poziom niżej w drzewie i jest dołączony do węzła „N”.

W nazwach węzłów nie można stosować znaków z akcentami. Nazwy węzłów mogą mieć maksymalnie długość 22 znaków.

Obiekt

Obiekt, to termin ogólny, który może oznaczać:

- ❖ Całego kreatora.
- ❖ Stronę kreatora.
- ❖ Formant (pole wyboru, pole tekstowe, przycisk, listę rozwijaną, itp.) na stronie.
- ❖ Zmienną.
- ❖ Itp.

Obiekt zwierzchni i obiekt podrzędny

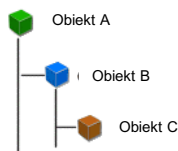
Jeśli obiekt „A” zawiera obiekt „B”:

- Obiekt „A” jest obiektem zwierzchnim obiektu „B”.
- Obiekt „B” jest „obiektem podrzędnym” obiektu „A”.

Uwaga: Związek ten dotyczy tworzenia, a nie dziedziczenia.

Pełna nazwa obiektu

Pełna nazwa obiektu składa się z nazw wszystkich jego obiektów zwierzchnich oraz nazwy samego obiektu. Każdy obiekt oddzielony jest kropką („.”). Jako przykład weźmy następującą strukturę:



Zatem pełna nazwa obiektu „C” ma następującą postać:

```
<Name of object "A">.<Name of object "B">.<Name of object "C">
```

Zmienna

Zmienna to nazwane miejsce składowania zawierające dane, które można modyfikować podczas działania kreatora. Każda zmienna ma nazwę umożliwiającą jej unikalną identyfikację wewnątrz kreatora. Wszystkie zmienne używane w kreatorze są globalne. Oznacza to, że może się do nich odwoływać dowolny węzeł kreatora za pomocą ich pełnych nazw.

W kreatorach AssetCenter używane są dwa rodzaje zmiennych:

- „Zmienne kreatora”, które są definiowane w węzłach typu „LONG” lub „STRING”. Typ węzła definiuje typ zmiennej; zmienna zdefiniowana w węźle typu „LONG” jest liczbą całkowitą typu long, zmienna zdefiniowana w węźle typu „STRING” jest ciągiem znaków. Zmienne te są ze swej definicji globalne. Oznacza to, że można się do nich odwoływać za pomocą ich pełnych nazw z dowolnego węzła kreatora. Jeśli zachodzi taka potrzeba, zmienne te są automatycznie wyliczane przez AssetCenter.
- Zmienne języka BASIC, używane w skryptach języka BASIC wewnątrz kreatora. Domyślnie zmienne te są lokalne, jednak mogą stać się globalne, jeśli zastosowana zostanie właściwość „COMMON” i „GLOBAL”. Zmienne te nie są wyliczane automatycznie przez AssetCenter.

Przejście

Element ten określa co się dzieje przy przejściu z jednej strony kreatora do innej. Dla danej strony może być zdefiniowane kilka przejść. Każde z nich ma własne warunki zdefiniowane przez użytkownika, które określają poprawność przejścia i muszą być spełnione, aby przejście zostało zainicjowane. Gdy użytkownik kliknie przycisk „Dalej”, wykonywane jest pierwsze poprawne przejście (to znaczy takie, dla którego są spełnione warunki). Jeśli żadne przejście nie jest poprawne, przycisk „Dalej” jest wyłączony.

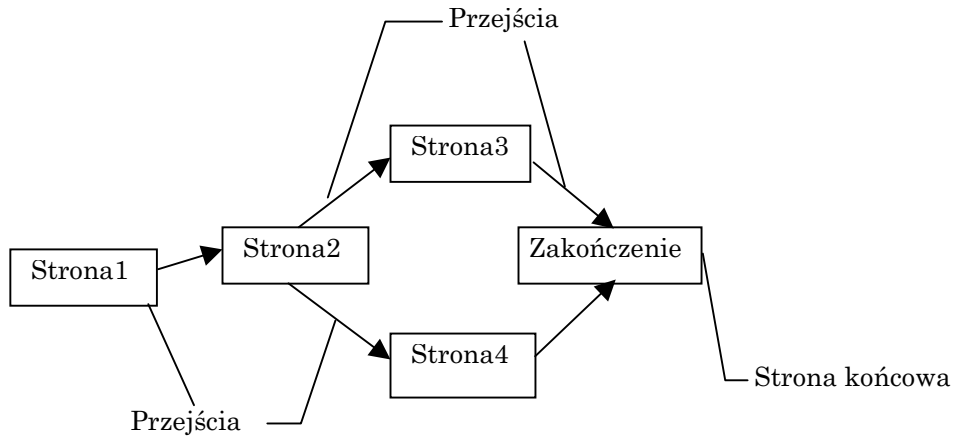
Model struktury

Działanie kreatora AssetCenter następuje po jego uruchomieniu. Z zasady, struktura kreatora opiera się na skrypcie definiującym strukturę, tzn.:

- Skrypt kreatora (a zatem kreator) składa się z węzłów.
- Każdy węzeł kreatora posiada nazwę, jeden lub więcej węzłów podrzędnych i zestaw właściwości. Typy węzłów podano poniżej:
 - ❖ „ROOT” (Węzeł główny). Węzeł ten jest unikalny i zawiera wszystkie inne.
 - ❖ „START”. Ten węzeł jest unikalny i zawiera skrypt, który jest wykonywany przy uruchomieniu kreatora.
 - ❖ „PAGE” (Strona). Ten typ węzła opisuje stronę kreatora.
 - ❖ „TRANSITION” (Przejście). Ten typ węzła opisuje przejście pomiędzy dwoma węzłami typu „PAGE”.
 - ❖ „FINISH” (Zakończenie). Ten węzeł jest unikalny i zawiera skrypt, który jest wykonywany przy zakończeniu kreatora.
 - ❖ „PARAMS” (Parametry). Ten węzeł jest unikalny i zawiera parametry przekazywane do innego kreatora. Można kolejno uruchamiać kilka kreatorów (z wymianą parametrów lub bez niej). Kreatory tego typu nazywa się szeregowymi.
 - ❖ Węzeł typu „LONG” lub „STRING”. Ten typ węzła zawiera zmienną zgodną z jego typem.
- Można używać właściwości typu „LONG” (numeryczna) lub „STRING” (ciąg znaków).

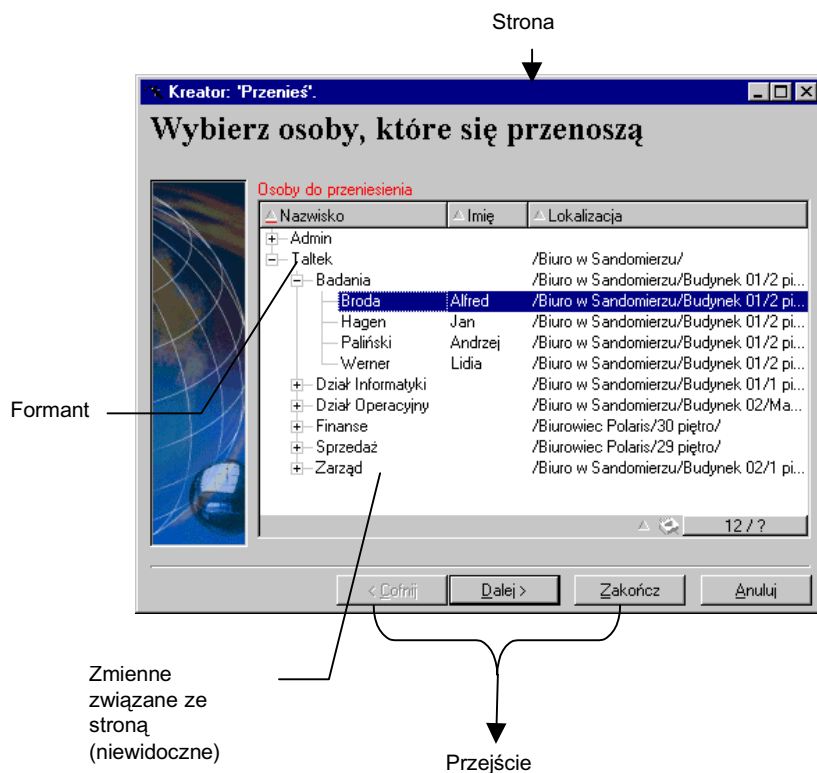
- Wartość właściwości jest określana za pomocą stałej lub za pomocą skryptu BASIC.

Kreatory składają się ze stron połączonych za pomocą przejść. Przejście z jednej strony na inną jest uwarunkowane wprowadzonymi informacjami i wyborami dokonanyymi przez użytkownika. Następujący diagram ilustruje strukturę kreatora:



Model strony kreatora

Strona kreatora jest zorganizowana w następujący sposób:



Uwagi ogólne

Kod kreatora (wprowadzany w polu **Skrypt kreatora** (nazwa SQL: WizardScript) w zakładce **Kreator** szczegółów działania korzystającego z kreatora) ma postać uporządkowanego tekstu, składającego się z bloków ujętych w klamry ({}). Tekst ten definiuje strukturę kreatora.

Każdy węzeł („Root”, „Page”, itp.) drzewa kreatora ma nieograniczoną liczbę węzłów podrzędnych i zestaw właściwości.

Ogólna struktura i składnia

Węzły mają następującą strukturę i składnię:

```
; This is a comment outside of the script
{ <Node type> <Node name>
  <Name of the property> = <Value of the property>
' This is a comment inside the script
  ...
  { <Name of the property> =
    ...
  }
  { <Node type> <Node name>
    <Name of the property> = <Value of the property>
    ...
    { <Name of the property> =
      ...
    }
  }
}
```

W odniesieniu do węzłów znajdują zastosowanie następujące zasady:

- Nazwy węzłów są opcjonalne. Jeśli nie określono nazwy węzła, AssetCenter automatycznie przydziela mu nazwę i numer.
- Nazwy węzłów nie mogą zawierać spacji.

- Jeśli nazwa węzła ma postać „=”, nie jest to już węzeł, ale wielowierszowa właściwość. Więcej informacji o wielowierszowych właściwościach znajduje się w sekcji „Składnia właściwości” w tym rozdziale.
- Wiersze zaczynające się od średnika („;”) na zewnątrz skryptu i wiersze zaczynające się od apostrofu (') wewnątrz skryptu są interpretowane jako komentarze i pomijane.

Ostrzeżenie: Spacja pomiędzy klamrą („{”) i typem węzła musi być zachowana. Jeśli jej nie będzie, AssetCenter nie uruchomi kreatora.

Właściwości węzła

Wartości właściwości można definiować za pomocą stałych lub skryptów. Stałe mogą być numeryczne, logiczne, lub tekstowe.

Właściwości związane z obiektami mogą być opcjonalne lub obowiązkowe. Mogą one być „logiczne” (uzupełniające definicję obiektu) lub „fizyczne” (mają one wpływ na aspekt wizualny obiektu).

Model deklaracyjny

Właściwość jest definiowana zgodnie z trybem deklaracyjnym definiującym referencje cykliczne ($A=\{B\}$, $B=\{A\}$):

```
<Name of the property>=<Script>
```

Z tą definicją jest związana lista zależności. Zatem jeśli mamy:

```
A=B+C
```

Właściwość „A” zależy od właściwości „B” i „C”. Lista zależności „A” składa się zatem z: „B”, „C”.

Zmiana właściwości następuje:

- Przy zmianie jednej z właściwości znajdującej się na liście zależności.

- Po działaniu użytkownika zmieniającym właściwość lub właściwość zależną.

Definicja stałej jako wartości właściwości

Następujące składnie definiują wartość stałą właściwości:

- Właściwość typu tekstowego:
 - <Nazwa właściwości> = „<Tekst>”
- Właściwość typu logicznego:
 - < Nazwa właściwości > = TRUE
 - < Nazwa właściwości > = FALSE
 - < Nazwa właściwości > (równoważnik < Nazwa właściwości > = TRUE)
- Właściwość typu numerycznego:
 - < Nazwa właściwości > = 42
 - <Nazwa właściwości> = {<Pełna nazwa zmiennej języka BASIC lub właściwości>}

Wartość logiczna „TRUE” jest równoważnikiem wartości numerycznej innej niż „0”. „FALSE” jest równoważnikiem wartości numerycznej „0”.

Odwoływanie się do właściwości

Aby odwołać się do właściwości obiektu (to znaczy do zawartości właściwości obiektu, a w szczególności jej wartości), należy zastosować następującą składnię:

```
{<Full name of the property>}
```

Zatem, jeśli chcemy odwołać się do właściwości „Prop” strony „Page1”, musimy wpisać:

```
{Page1.Prop}
```

W tej składni, w pełnej nazwie właściwości wielkość liter jest nieistotna.

Definiowanie skryptu jako wartości właściwości

Pojęcie skryptu

Skrypt jest to program języka BASIC składający się z jednego lub wielu wierszy, zwracający wartość w zmiennej globalnej „RetVal”. W przypadku skryptu jednowierszowego, zmienna ta jest niejawna. W przypadku skryptu wielowierszowego należy ją podać.

W przypadku wszystkich skryptów BASIC, należy zwracać uwagę na typ zwracanej wartości. Zależy ona od typu właściwości wyliczanej przez skrypt.

Składnia skryptu jednowierszowego

```
<Name of the property>=<Script>
```

Na przykład:

```
Variable="The name is: " & {Name}
```

Poprzedni skrypt jednowierszowy jest równoważny z poniższym skryptem wielowierszowym:

```
{Variable=  
RetVal="The name is: " & {Name}  
}
```

Składnia skryptu wielowierszowego

```
{ <Name of the property>=  
  <Script>  
}
```

Na przykład:

```
{ LABEL =  
  IF {Page1.Title}="Choose an employee" THEN  
    RetVal="Employee"  
  ELSE  
    RetVal="Department"  
  END IF  
}
```

Metody stosowane we właściwościach

Metoda pozwala pobrać wartość związaną z właściwością lub węzłem lub nawet wykonać funkcję odnoszącą się do właściwości. W tym sensie można ją uważać za zaawansowaną funkcję. Składnia jest następująca:

```
{node.node.node[.property][.method()]}
```

W tym przykładzie, znaki „[, i]” obejmują elementy opcjonalne.

Na przykład, aby pobrać liczbę linii z formantu „LISTBOX” na stronie „PAGE1”, używamy metody „COUNT” związanej z tym typem formantu. Polecenie ma następującą postać:

```
PAGE1.LISTBOX.VALUES.COUNT()
```

W poniższej tabeli podano właściwości posiadające metody:

Lista metod stosowanych z właściwościami		
Właściwość	Metoda	Opis
DBLISTBOX.TABLE	LABEL()	Zwraca etykietę tabeli umożliwiając wygenerowanie formantu DBLISTBOX.
DBEDIT.TABLE	LABEL()	Zwraca etykietę tabeli umożliwiając wygenerowanie formantu DBEDIT.
DBEDIT.FIELD	LABEL()	Zwraca etykietę pola tabeli umożliwiając wygenerowanie formantu DBEDIT.
LISTBOX	COUNT()	Zwraca liczbę wierszy w formancie LISTBOX.
LISTBOX	CELL(I,j)	Zwraca zawartość komórki (i,j) właściwości typu tabela.
LISTBOX	VALUE(i)	Zwraca wartość wiersza „i” właściwości typu tabela.
LISTBOX	LINE(i)(Zwraca zawartość wiersza „i” właściwości typu tabela.
LINKEDIT.TABLE	LABEL()	Zwraca etykietę tabeli umożliwiając wygenerowanie formantu LINKEDIT.

Właściwość typu tabela

Właściwości typu tabela to właściwości, których wartość jest definiowana zgodnie z następującym formatem:

```
<Column|Column|Column|...>=<Value of the line>,
<Column|Column|Column|...>=<Value of the line>, ...
```

<Wartość wiersza> jest odpowiednikiem elementu („Identyfikator”) danego rekordu.

Wartości tych właściwości można przeglądać w postaci tabeli:

		Kolumna 1	Kolumna 2	Kolumna 3
Numer wiersza: 1	Wartość wiersza (np.: 18)	Komórka (1,1)	Komórka (2,1)	Komórka (3,1)
Numer wiersza: 2	Wartość wiersza (np.: 29)	Komórka (1,2)	Komórka (2,2)	Komórka (3,2)
Numer wiersza: 3	Wartość wiersza (np.: 78)	Komórka (1,3)	Komórka (2,3)	Komórka (3,3)
...

Przykład

Rozważmy właściwość „VALUES” zawierającą wartość będącą wynikiem zapytania do tabeli wydziałów i pracowników. Zapytanie zwraca wartości pól **Nazwisko** (nazwa SQL: Name) i **Imię** (nazwa SQL: FirstName) dla każdego rekordu w tej tabeli. Załóżmy, że ta właściwość ma następującą wartość:

```
VALUES="Colombo|Gerard=32,Lubeck|Alexander=64,Daquin|William=24"
```

Wartość tę można przeglądać w postaci tabeli:

		Nazwisko	Imię
1	32	Borewicz	Gerard
2	64	Lubeck	Alexander
3	24	Daquin	William

Używanie zmiennych globalnych CXTXTABLE i CXTXTSELECTION

Zawartość tych zmiennych można pobrać za pomocą następującej składni:

```
[CXTXTABLE]  
[CXTXTSELECTION]
```

Następująca tabela przedstawia cechy tych dwóch zmiennych:

Nazwa zmiennej	Opis zmiennej	Uwagi
CTXTTABLE	Zawiera tabelę aktywną w momencie uruchomienia kreatora. Zmienna typu „String”.	Zmienna ta jest automatycznie wypełniana przez AssetCenter. Użytkownik nie może spowodować zmiany wartości.
CTXTSELECTION	Zawiera listę wewnętrznych identyfikatorów rekordów wybranych w momencie uruchomienia kreatora. Zmienna typu „String”.	Zmienna ta jest automatycznie wypełniana przez AssetCenter. Użytkownik nie może spowodować zmiany wartości.

Sekwencjonowanie kreatorów

Kreatora można użyć do zainicjowania wykonania innego kreatora i przekazania parametrów (zmiennych) do tego kreatora. Metoda ta jest nazywana sekwencjonowaniem kreatorów.

Wykonywanie

Aby kreator A zainicjował wykonanie kreatora B, jego węzeł Finish (Zakończenie) musi posiadać właściwość CHAIN. Właściwość ta musi zawierać wartość nazwy SQL działania typu **Kreator** do wykonania, w tym przypadku „B”.

Parametry

Parametry są przekazywane do kreatora B za pomocą węzła PARAMS kreatora A. Parametry te są dodawane do parametrów znajdujących się w węźle PARAMS kreatora B. Jeśli ten sam parametr jest zdefiniowany w węźle PARAMS kreatora A i w węźle PARAMS kreatora B, parametry kreatora mają pierwszeństwo nad parametrami kreatora B.

Funkcje języka Basic

Oprócz ogólnych funkcji AssetCenter (za wyjątkiem funkcji „AmCounter”), kreatory akceptują następujące dodatkowe funkcje:

- AmComputeString()
- AmProgress()
- AmLog()
- AmMsgBox()
- AmExecTransition()

Uwaga: Przy wywoływaniu funkcji języka Basic ze skryptu kreatora, zawsze należy przypisywać do zmiennej wartość zwracaną przez funkcję. W przeciwnym razie Basic zwróci błąd. Zatem następujący przykład nie zostanie skompilowany:

```
AmGetFieldLongValue(hRecord, "lUserId", {lEmplDeptId})
```

Prawidłowy skrypt ma następującą postać:

```
Dim lValue as Long  
lValue=AmGetFieldLongValue(hRecord, "lUserId", {lEmplDeptId})
```

Definicja węzła ‘Root’

Węzeł „Root” (główny) opisuje całość kreatora. Składa się on z bloku właściwości ogólnych, które mogą się odnosić do całego kreatora, oraz szeregu węzłów podrzędnych, które przedstawiają obiekty składowe kreatora.

Składnia węzła 'Root'

Składnia węzła „Root” jest następująca:

```
' Block of general properties of the root node
NAME=...
IMAGE=...
...
' Definition of sub-nodes of the root node
{ FINISH
  ...
}
{ PAGE
  ...
}
{ TRANSITION
  ...
}
```

Właściwości węzła 'Root'

W następnej tabeli podano wszystkie właściwości ogólne, które mogą być zdefiniowane w węźle „Root”:

Właściwości logiczne węzła „Root” (głównego)			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
NAME=„<Nazwa kreatora>”	Określa nazwę kreatora. Właściwość typu „String” (ciąg znaków).	NAME=„Kreator Przenies”	Należy zdefiniować wartość tej właściwości. Nazwa kreatora ograniczona jest do 22 znaków. Właściwość ta jest używana do sekwencjonowania kreatora: Dane związane z tym kreatorem są przechowywane pod tą nazwą. Dlatego lepiej, aby różne kreatory miały różne nazwy.
TITLE=„<Tytuł okna>”	Określa tytuł okna kreatora. Właściwość typu „String”.	TITLE=„KreatorPrzenies”	Usilnie zalecamy zdefiniowanie wartości dla tej właściwości.
GLOBAL=<Skrypt>	Zawiera skrypt języka BASIC (lub kod). Właściwość ta jest przydatna do definiowania zmiennych globalnych, (odnoszących się do całego kreatora) i globalnych funkcji, które mogą być używane we wszystkich węzłach kreatora. Właściwość typu „Script”.	{GLOBAL=Dim Filter As String}	

Właściwości logiczne węzła „Root” (głównego)			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
SERIALIZE=<TRUE FALSE>	Pozwala (=TRUE) lub nie pozwala (=FALSE) sekwencjonować kreatora. Jeśli kreator jest sekwencjonowany, zachowuje on poprzednio wprowadzone wartości (zapisując je w pliku „ini”) do wykorzystania przy następnym uruchomieniu. Właściwość typu logicznego.	SERIALIZE=TRUE	Domyślnie właściwość ta ma wartość „FALSE”.

Właściwości fizyczne węzła „Root” (głównego)			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
IMAGE=„<Ścieżka bitmapy>” IMAGE16=„<Ścieżka bitmapy>”	Określa plik graficzny typu bitmapa (.bmp) wyświetlany w kreatorze. Właściwość typu „String”.	IMAGE=„Wizard.bmp”	Jeśli dla tej właściwości nie jest zdefiniowana żadna wartość, nie jest wyświetlany żaden obraz. Ścieżka pliku graficznego jest określana względem foldera „Config” AssetCenter. AssetCenter najpierw szuka obrazu w bazie danych. Jeśli zdefiniujesz wartość właściwości „IMAGE16”, jest ona używana zamiast właściwości „IMAGE” jeśli głębia koloru ekranu wynosi 16.
WIDTH=<Szerokość>	Definiuje domyślną szerokość („<Szerokość>”) okna kreatora. Jest ona wyrażona w jednostce twip. Właściwość typu „Long”.	WIDTH=6000	
HEIGHT=<Wysokość>	Definiuje domyślną wysokość („<Wysokość>”) okna kreatora. Jest ona wyrażona w jednostce twip. Właściwość typu „Long”.	HEIGHT=5000	

Węzły podrzędne węzła 'Root'

Typy węzłów podrzędnych, które można zdefiniować dla węzła głównego, podane są w następującej tabeli. Każdy typ węzła stanowi „Obiekt”.

Węzły podrzędne węzła „Root” (głównego)	
Typ węzła	Opis
PAGE <Nazwa strony>	Opisuje stronę kreatora.
TRANSITION <Nazwa przejścia>	Opisuje przejście pomiędzy dwiema stronami
FINISH <Nazwa węzła zakończenia>	Opisuje końcowe przejście z ostatniej strony kreatora (do zakończenia). Ten węzeł typu „Transition” (przejście) nie ma właściwości „FROM” (z) i „TO” (do).
START <Nazwa węzła>	Zawiera na przykład skrypt wykonywany przy uruchomieniu kreatora (za pomocą właściwości „DO” –wykonaj) oraz nazwę strony początkowej kreatora (właściwość „TO” –do).

Definicja węzła 'Page'

Węzeł „Page” (strona) opisuje stronę w kreatorze. Składa się on z bloku właściwości odnoszących się do bieżącego węzła i wszystkich węzłów podrzędnych; oraz zbioru węzłów podrzędnych definiujących obiekty zdefiniowane na stronie.

Składnia węzła 'Page'

Składnia węzła „Page” jest następująca:

```
` Declaration of the page
{ Page <Name of the page>
` Block of properties of the page node
IMAGE=...
TITLE=...
` Definition of sub-nodes of the „Page” node
{ TRANSITION
  ...
}
{ <Control type> <Control name>
  ...
}
...
}
```

Właściwości węzła 'Page'

W poniższej tabeli podano wszystkie właściwości, które można zdefiniować w węźle „Page”:

Właściwości logiczne węzła „Page” (strona)			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
TITLE=„<Tytuł strony>”	Określa tytuł strony. Jest on wyświetlany czcionką wytłuszczoną u góry strony. Właściwość typu „String” (ciąg znaków).	TITLE=„Przenies”	Jeśli dla tej właściwości nie jest zdefiniowana żadna wartość, dziedziczy ona wartość właściwości „TITLE” węzła „Root”. W odróżnieniu od etykiet, ciąg ten nie obsługuje znaczników HTML.
ONENTER=<Skrypt>	Określa skrypt języka BASIC wykonywany przy otwarciu strony. Właściwość typu „Script”.	{ONENTER=Am MsgBox(„Hello”)}	

Fizyczne właściwości węzła „Page” (strona)			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
IMAGE=„<Ścieżka do pliku bitmapy>” IMAGE16=„<Ścieżka do pliku bitmapy>”	Określa plik graficzny typu bitmapa (.bmp) wyświetlany na stronie kreatora. Właściwość typu „Script”.	IMAGE=„C:\Images\Page1.bmp”	Jeśli dla tej właściwości nie zdefiniowano wartości, dziedziczy ona wartość właściwości „IMAGE” węzła „Root”. Jeśli dla tej właściwości zostanie zdefiniowana pusta wartość, nie będzie wyświetlany żaden obraz. Jeśli zostanie zdefiniowana wartość właściwości „IMAGE16”, będzie ona używana zamiast właściwości „IMAGE” gdy głębokość koloru ekranu wynosi 16.

Węzły podrzędne węzła 'Page'

Dla węzła „Page” można zdefiniować dwa typy węzłów podrzędnych:

Węzły podrzędne węzła „Page” (strona)	
Typ węzła / „Obiekt”	Opis
<Typ formantu> <Nazwa formantu>	Określa formant wyświetlany na bieżącej stronie.
TRANSITION <Nazwa przejścia>	Określa przejście pomiędzy bieżącą stroną, a inną stroną kreatora.

Definicja węzła 'Transition'

Węzeł „Transition” (przejście) opisuje przejście pomiędzy dwiema stronami kreatora. Składa się on wyłącznie z bloku właściwości.

Przejścia mogą być definiowane z wewnątrz węzła „Page” (w tym przypadku, nie wymagają one właściwości „FROM”) lub z węzła „Root”. Końcowe przejście prowadzące do zakończenia kreatora, jest opisane w węźle „FINISH” (na poziomie węzła „Root”) i nie ma ono właściwości „FROM”, ani „TO”.

Składnia węzła 'Transition'

Składnia węzła „Transition” jest następująca:

```
' Declaration of the transition
{ TRANSITION0 <Name of the transition>
' Block of properties of the transition node
FROM=...
TO=...
CONDITION=...
}
```

Właściwości węzła 'Transition'

W poniższej tabeli podano wszystkie właściwości, które można zdefiniować w węźle „Transition”:

Logiczne właściwości węzła „Transition”			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
FROM=„<Nazwa strony wyjściowej>”	Określa wyjściową stronę przejścia. Właściwość typu „String” (ciąg znaków).	FROM=„Page2”	Ta właściwość jest obowiązkowa, jeśli przejście jest zdefiniowane wewnątrz węzła „Root”, a nie jest stosowana, jeśli przejście jest definiowane w węźle „Page”, „Finish” lub „Start”.
TO=„<Nazwa strony docelowej>”	Określa docelową stronę przejścia. Właściwość typu „String”.	TO=„Page3”	Ta właściwość jest obowiązkowa, jeśli przejście jest zdefiniowane wewnątrz węzła „Root” lub „Page”, a nie jest stosowana, jeśli przejście jest definiowane w węźle „Finish”.
CONDITION=<Skrypt>	Określa warunek, po którego spełnieniu inicjowane jest przejście. Właściwość typu „Script” zwracająca wartość logiczną.	CONDITION={Comment}=„użytkownik”	Właściwość ta jest niedostępna w węzłach typu „Start”.
DO=<Skrypt>	Definiuje skrypt do wykonania w momencie przejścia. Właściwość typu „Script”.	{DO= Filter=„”}	

Specyfika węzła 'Transition'

Węzeł „Transition” (przejście) nie ma węzła podrzędnego.

Dlaczego przejścia są definiowane w węźle „Root”?

Oddzielenie przejść od węzłów typu „Page” pozwala tworzyć strony, które mogą być zastosowane w dowolnych skryptach, a także usprawnia pisanie skryptów.

Definicja węzła 'Finish'

Węzeł „Finish” (końcowy) opisuje końcowe przejście, które prowadzi do końcowej strony kreatora. Jest to szczególny typ węzła „Transition” (przejście), który nie ma właściwości „FROM” i „TO”. Oprócz tego wyjątku, składnia właściwości w węźle „Finish”, jest taka sama, jak w węźle „Transition”.

Właściwość CHAIN (dołącz), specyficzna dla węzła końcowego, pozwala zainicjować wykonanie następnego kreatora.

Właściwość logiczna węzła „Finish” końcowego			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
CHAIN=<nazwa SQL kreatora do uruchomienia>	Określa kreatora do uruchomienia przy zakończeniu bieżącego kreatora. Właściwość typu „String”	CHAIN=„Przenieś”	

Węzeł PARAMS pozwala przekazywać parametry do następnego kreatora.

Definicja węzła 'Start'

Węzeł „Start” opisuje sposób uruchomienia kreatora. Jest to specyficzny węzeł typu „Transition” (przejście), który nie ma właściwości „FROM”, ani „CONDITION”. Oprócz tego wyjątku, składnia i właściwości węzła „Start” są takie same jak w przypadku węzła „Transition”.

Definicja węzłów typu Long i String

Węzły typu Long i String służą do definiowania zmiennych. Odwołania do nich mogą występować we wszystkich węzłach kreatora. Nazwa węzła określa nazwę zmiennej.

Węzły tego typu mają pojedynczą właściwość, której typ zależy od węzła; jest ona typu LONG, jeśli węzeł jest typu Long; lub typu STRING, jeśli węzeł jest typu String. Właściwość ta, VALUE, pozwala definiować wartość zmiennej.

Właściwość logiczna węzła typu Long lub String			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
VALUE=<Wartość>	Określa wartość zmiennej, której nazwa jest identyczna z nazwą węzła. Właściwość typu „Long” dla węzła typu Long lub „String” dla węzła typu String.	VALUE=12	

Węzły typu Long i String można definiować w dowolnym węźle kreatora. Nie mają one węzłów podrzędnych.

Definicja węzła 'Control'

Formanty na stronie służą do interakcji z użytkownikiem. Na danej stronie można zdefiniować dowolną ilość formantów. AssetCenter w pełni kontroluje organizację formantów na stronie. Nie trzeba określać pozycji każdego definiowanego formantu.

Węzły typu „Control” (formant) składają się wyłącznie z bloku właściwości dotyczących danego formantu.

Ogólna składnia węzła 'Control'

Ogólna składnia węzła typu „Control” (formant) jest następująca:

```
' Declaration of the control
{ <Control type> <Control name>
' Properties of the control
...
}
```

Typy formantów i ich właściwości

Wszystkie formanty mają wspólne właściwości. Istnieją jednakże właściwości specyficzne dla poszczególnych formantów. Poniższa sekcja zawiera następujące informacje:

- Wspólne właściwości formantów
- Formant CheckBox
- Formant ComboBox
- Formant ListBox
- Formant Label
- Formant ProgressBar
- Formant CommandButton

- Formant OptionButtons
- Formant DBListBox
- Formant DBQueryBox
- Formant DBEdit
- Formant DBTable
- Formant DBPath
- Formant LinkEdit
- Formant TextBox

Wspólne właściwości formantów

W poniższej tabeli podano opcjonalne właściwości odnoszące się do wszystkich formantów:

Właściwości logiczne wspólne dla wszystkich formantów			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwaga
MANDATORY=<TRUE FALSE> (OBOWIĄZKOWE=<PRAWDA FAŁSZ>)	Zmusza użytkownika do wypełnienia formantu dla zachowania poprawności przejścia. Właściwość typu logicznego.	MANDATORY=TRUE	Właściwość ta nie jest dostępna dla formantów typu „CHECKBOX” (pole wyboru) i „LABEL” (etykieta)
VALUE=<Wartość>	Określa domyślną wartość tworzonego formantu. <Wartość> zależy od danego formantu. Właściwość typu logicznego lub „String”.	Na przykład, jeśli jest to formant „CHECKBOX”, wartości mogą być następujące: „TRUE” (prawda) lub „FALSE” (fałsz).	
Właściwości fizyczne wspólne dla wszystkich formantów			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwaga
VISIBLE=<TRUE FALSE>	Określa, czy formant jest widoczny (=TRUE) lub nie (=FALSE). Właściwość typu logicznego	Label1.Visible=TRUE	
ENABLED=<TRUE FALSE>	Określa, czy formant jest aktywny (=TRUE) lub nie (=FALSE). Właściwość typu logicznego.	Choice1.Enabled=FALSE	

Właściwości logiczne wspólne dla wszystkich formantów			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwaga
READONLY=<TRUE FALSE>	Określa, czy formant jest tylko do odczytu (=TRUE) i nie może być edytowany przez użytkownika; lub edytowalny (=FALSE)	READONLY=TRUE	
LABEL=„<Tekst etykiety>”	Definiuje opcjonalny tekst, wyświetlany nad formantem. Właściwość typu „String”.	Choice1.Label=„Wybierz osobę”	Ten formant obsługuje znaczniki HTML

Formant CheckBox

Formant „CHECKBOX” definiuje pole wyboru.

Właściwości

Oprócz opcjonalnych właściwości, wspólnych dla wszystkich formantów, formant „CHECKBOX” przyjmuje następującą właściwość:

Właściwość formantu „CHECKBOX” (pole wyboru)		
Nazwa właściwości=Wartość	Opis właściwości	Przykład
TEXT=„<Tekst>”	Definiuje tekst pola wyboru. Właściwość typu „String”.	TEXT=„Określ według nazwy”

Formant ComboBox

Formant „COMBOBOX” (lista kombinowana) definiuje pojedynczy wybór z wyczerpania (listy pozycji) predefiniowanych wartości.

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich formantów, formant „COMBOBOX” przyjmuje następującą właściwość:

Właściwości fizyczne formantu „COMBOBOX” (lista kombinowana)			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwaga
VALUES=„<Nazwa=Wartość, Nazwa=Wartość, Nazwa=Wartość, ...>”	Definiuje pary wartości („Nazwa”=„Wartość”) dla formantu listy kombinowanej. „Nazwa” określa tekst, który jest wyświetlany w formancie, „Wartość” – przypisywaną wartość, jeśli dana nazwa zostanie wybrana przez użytkownika. Właściwość typu „String”.	VALUES=„Tabela zasobów=zasób, Użytkownik= użytkownik”	Jeśli parametr „Value” zostanie pominięty, AssetCenter automatycznie przypisze wartość. Na przykład, wyrażenie: VALUES=„A,B,C” Jest równoważne z VALUES=„A=1,B=2,C=3”

Formant ListBox

Formant „LISTBOX” (pole listy) definiuje listę obiektów do wyboru. Formanty „LISTBOX” mogą być formantami wielokolumnowymi.

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich formantów, formant „LISTBOX” przyjmuje następujące właściwości:

Właściwości fizyczne formantu „LISTBOX” (pole listy)			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwaga
LISTHEIGHT=<Procent>	Określa rozmiar formantu „LISTBOX” w odniesieniu do innych formantów „LISTBOX” w całym kreatorze. Właściwość typu „Long”.	LISTHEIGHT=50	Jeśli istnieją dwa formanty „LISTBOX” o wartościach tej właściwości odpowiednio „10” i „20”, drugi formant będzie dwukrotnie większy od pierwszego.
MULTISEL=<TRUE FALSE>	Określa, czy formant obsługuje zaznaczanie wielu pozycji (=TRUE), czy nie (=FALSE). Właściwość typu logicznego.	MULTISEL=TRUE	
COLTITLE=„<Kolumna Kolumna Kolumna...>”	Określa nazwę i właściwości kolumn listy. Parametr „Kolumna” definiuje tekst kolumny. Właściwość typu „String”.	COLTITLE=„Nazwisko Imię”	
COLWIDTH=„<Szerokość Szerokość Szerokość...>”	Określa szerokość kolumny proporcjonalnie do szerokości formantu. Właściwość typu „String”.	COLWIDTH=„50 50”	

Właściwości fizyczne formantu „LISTBOX” (pole listy)			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwaga
VALUES=„<Tekst Tekst ...= Wartość, Tekst Tekst ...= Wartość ,...>”	Definiuje pary wartości („Tekst Tekst ...” = „Wartość”) dla formantu „LISTBOX”. Parametr „Tekst Tekst ..” określa tekst wyświetlany w każdej kolumnie danego wiersza formantu „LISTBOX”, Parametr „Wartość” określa wartość, jaką przybiera formant jeśli użytkownik wybierze dany wiersz. Właściwość typu „String”.	VALUES=„Tabela zasobów=zasób, Użytkownik=użytkownik,”	Jeśli parametr „Wartość” zostanie pominięty, AssetCenter automatycznie przypisze wartość. Na przykład: VALUES=„A,B,C” jest odpowiednikiem VALUES=„A=1,B=2,C=3” Właściwość tę można wypełnić bezpośrednio lub za pomocą funkcji AmdbGetList, wpisując na przykład: VALUES=AmDbGetList(„SELECT Name, FirstName FROM amEmplDept WHERE Name Like 'A%' , „ ” , „” , „=”) Nie należy mylić właściwości „VALUES” i „VALUE”.

Obowiązkowa logiczna właściwość formantu „LISTBOX” (lista)		
Nazwa właściwości=Wartość	Opis właściwości	Przykład
TABLE=<Nazwa tabeli>	Nazwa tabeli używanej do pobierania nazw kolumn. Właściwość typu „String”.	TABLE= „amAsset”
COLNAME=<Tytuł Tytuł ...>	Definiuje tytuły kolumn za pomocą nazw SQL pól w tabeli określonej we właściwości „TABLE”. Właściwość ta pozwala również określać zastosowane pola edycji. Formant jest taki sam, jak stosowany w AssetCenter do wypełniania pola. AssetCenter przy określaniu tytułów kolumn najpierw przyjmuje wartości właściwości „COLTITLE” (jeśli została zdefiniowana). Właściwość typu „String”.	COLNAME= „Nazwisko Imię”
EDITABLE=<„wartość_logiczna wartość_logiczna ... wartość_logiczna”>	Określa, czy każda kolumna listy jest edytowalna. Właściwość typu „String”.	EDITABLE=„1 0 1 1”

Formant Label

Formant „LABEL” (etykieta) określa etykietę. Formant ten ma następującą właściwość:

Właściwości fizyczne formantu „LABEL”			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwaga
CAPTION=<Tekst>	Zawiera tekst wyświetlany w etykiecie.	CAPTION= „Wybierz lokalizację”	

Formant OptionButtons

Formant „OPTIONBUTTONS” (przyciski opcji) definiuje grupę przycisków opcji (przycisków „radiowych”).

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich formantów, formant „OPTIONBUTTONS” przyjmuje następujące właściwości:

Właściwości fizycznie formantu „OPTIONBUTTONS” (przyciski opcji)		
Nazwa właściwości=Wartość	Opis właściwości	Przykład
VALUES=„<Tytuł=Wartość, Tytuł=Wartość, Tytuł=Wartość,....>”	Określa pary wartości („Tytuł”=„Wartość”) dla formantu „OPTIONBUTTONS”. Parametr „Nazwa” określa tekst przycisku opcji, parametr „Wartość” określa wartość przypisaną do formantu jeśli użytkownik wybierze dany przycisk opcji. Właściwość typu „String”.	VALUES=„Tabela zasobów=zasób, Użytkownik=użytkownik”
BORDER=<TRUE FALSE>	Określa, czy wokół przycisków opcji rysowana jest ramka (=TRUE), czy nie (=FALSE) Właściwość typu logicznego.	BORDER= TRUE

Formant ProgressBar

Formant „PROGRESSBAR” definiuje wskaźnik zaawansowania.

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich formantów, formant „PROGRESSBAR” przyjmuje następujące właściwości:

Właściwości fizyczne formantu „PROGRESSBAR”		
Nazwa właściwości=Wartość	Opis właściwości	Przykład
MAXVALUE=<Wartość maksymalna>	Określa wartość maksymalną odpowiadającą 100% wskaźnika zaawansowania. Właściwość „VALUE” wskazuje bieżącą wartość formantu. Właściwość typu „Long”.	MAXVALUE=200

Formant CommandButton

Formant „COMMANDBUTTON” definiuje przycisk polecenia inicjujący działanie.

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich formantów, formant „COMMANDBUTTON” przyjmuje następujące właściwości:

Właściwości fizyczne formantu „COMMANDBUTTON”		
Nazwa właściwości=Wartość	Opis właściwości	Przykład
WIDTH=<Szerokość>	Określa szerokość przycisku w jednostkach twip. Właściwość typu „Long”.	WIDTH=250
HEIGHT=<Wysokość>	Określa wysokość przycisku w jednostkach twip. Właściwość typu „Long”.	HEIGHT=125
CAPTION=<Tekst>	Określa tekst wyświetlany na przycisku. Właściwość typu „String”.	CAPTION=„Start”
CLICK=<skrypt BASIC>	Określa skrypt języka BASIC wykonywany po kliknięciu przycisku przez użytkownika.	

Formant DBListBox

Formant „DBLISTBOX” określa listę rekordów do wyboru z bazy danych. Może mieć on postać formantu wielokolumnowego. Lista wyświetlana w formancie jest wynikiem częściowego zapytania AQL (używany jest tylko warunek WHERE) bazy danych AssetCenter.

Właściwość „VALUE” zwraca listę identyfikatorów („Identyfikator”) wybranych linii. Nie można uzyskać dostępu do wartości komórek na liście. W tym celu trzeba wykonać następną zapytanie lub zastosować formant typu „LISTBOX” (pole listy).

Właściwości

Oprócz opcjonalnych formantów, wspólnych dla wszystkich właściwości, formant „DBLISTBOX” przyjmuje następujące właściwości:

Właściwości fizyczne formantu „DBLISTBOX”			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
TABLE=„<Nazwa SQL tabeli>”	Określa tabelę, do której kierowane jest zapytanie. Właściwość typu „String”.	TABLE=„amAsset”	Ta właściwość jest obowiązkowa.
COLNAME=„<nazwa SQL pola lub łącza nazwa SQL pola lub łącza ...>”	Określa elementy danych pobierane z bazy danych (określane za pomocą ich nazw SQL). Właściwość typu „String”.	COLNAME=„Nazwisko Imię”	
COLWIDTH=„<Szerokość Szerokość Szerokość ...>”	Określa format wyświetlania danych pobranych z bazy, jako procent ogólnego rozmiaru formantu „DBLISTBOX”. Właściwość typu „String”.	COLWIDTH=„40 60”	
LISTHEIGHT=<Procent>	Określa rozmiar formantu „DBLISTBOX” względem innych formantów „DBLISTBOX” w całości kreatora. Właściwość typu „Long”.	LISTHEIGHT=50	Jeśli istnieją dwa formanty „DBLISTBOX” o wartościach tej właściwości odpowiednio „10” i „20”, drugi formant będzie dwukrotnie wyższy względem pierwszego.
TREE=<TRUE FALSE>	Wyświetla dane w postaci drzewa (=TRUE) lub nie (=FALSE). Właściwość typu logicznego.	TREE=TRUE	Właściwość ta ma domyślnie wartość „FALSE”

Właściwości fizyczne formantu „DBLISTBOX”			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
MULTISEL=<TRUE FALSE>	Określa, czy formant obsługuje zaznaczanie wielu pozycji (=TRUE), czy nie (=FALSE). Właściwość typu logicznego.	MULTISEL=TRUE	
DBCLICK=<TRUE FALSE>	Jeśli właściwość ta jest ustawiona jako TRUE, AssetCenter będzie symulować kliknięcie przycisku „Dalej” bieżącej strony.	DBCLICK=FALSE	
FILTER=„<Warunek>”	Określa warunek „WHERE” języka AQL filtrujący rekordy przetwarzane przez zapytanie. Właściwość typu „String”.	FILTER=„User.IE mplDeptId='Borewicz, Gerard'”	

Formant DBQueryBox

Formant „DBQUERYBOX” definiuje listę rekordów do wyboru. Formant ten może mieć postać wielokolumnową. Lista wyświetlana w formancie jest wynikiem zapytania AQL kierowanego do bazy danych AssetCenter.

Właściwości


Oprócz opcjonalnych właściwości wspólnych dla wszystkich formantów, formant „DBQUERYBOX” przyjmuje następujące właściwości:

Właściwości fizyczne formantu „DBQUERYBOX”			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
QUERY=„<Pełne zapytanie AQL>”	Definiuje zapytanie AQL zwracające informacje wyświetlane w formancie „DBLIST”. Właściwość typu „String”.	QUERY=„SELECT Name, FirstName FROM amEmplDept WHERE Location=Biurowiec Polaris'”	
COLTITLE=„<Kolumna Kolumna ...>”	Definiuje tytuły kolumn listy. Właściwość typu „String”.	COLTITLE=„Nazwisko Imię”	
COLWIDTH=„<Szerokość Szerokość ...>”	Definiuje rozmiar kolumn listy jako procent ogólnego rozmiaru formantu. Właściwość typu „String”.	COLWIDTH=„50 50”	
LISTHEIGHT=<Procent>	Definiuje rozmiar formantu „DBQUERYBOX” proporcjonalnie do innych formantów „DBQUERYBOX” w całym kreatorze. Właściwość typu „Long”.	LISTHEIGHT=50	Jeśli istnieją dwa formanty „DBQUERYBOX” o wartościach tej właściwości odpowiednio „10” i „20”, drugi formant będzie dwukrotnie wyższy od pierwszego.
TREE=<TRUE FALSE>	Wyświetla dane w trybie drzewa (=TRUE) lub nie (=FALSE). Właściwość typu logicznego.	TREE=TRUE	Domyślnie, właściwość ta ma wartość „FALSE”
MULTISEL=<TRUE FALSE>	Określa, czy formant obsługuje zaznaczanie wielu pozycji, (=TRUE), czy nie (=FALSE). Właściwość typu logicznego.	MULTISEL=TRUE	

Właściwości fizyczne formantu „DBQUERYBOX”			
Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
DBCLICK=<TRUE FALSE>	Jeśli właściwość ta ma wartość TRUE, AssetCenter będzie symulować kliknięcie przycisku „Dalej” bieżącej strony. Właściwość typu logicznego.	DBCLICK=FALSE	

Formant DBEdit

Formant „DBEDIT” tworzy formant identyczny z formantem do wypełniania pola w bazie danych AssetCenter. Postać formantu różni się w zależności od typu pola (data, walutowe, itp.).

Przycisk z lupą  w tym formancie pozwala wybierać wartości znajdujące się w bazie danych, można jednak wprowadzać inne wartości.

W przypadku tego formantu, właściwość „VALUE” jest typu „Variant” (typ zależny od formantu).

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich formantów, formant „DBEDIT” przyjmuje następujące właściwości:

Obowiązkowe właściwości fizyczne formantu „DBEDIT” w trybie „Normal”		
Nazwa właściwości=Wartość	Opis właściwości	Przykład
TABLE=„<Nazwa SQL tabeli>”	Nazwa tabeli zawierającej pole, którego formant chcemy skopiować. Właściwość typu „String”.	TABLE=amAsset
FIELD=„<Nazwa SQL pola>”	Nazwa pola, którego formant chcemy skopiować. Właściwość typu „String”.	FIELD=seAcquMethod

Formant DBTable

Formant „DBTABLE” tworzy formant służący do wprowadzania tabeli w bazie danych AssetCenter.

Właściwości

Formant ten nie ma dodatkowych właściwości.

Formant DBPath

Formant „DBPATH” tworzy formant do wprowadzania pola w bazie danych AssetCenter.

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich formantów, formant „DBPATH” musi posiadać następującą obowiązkową właściwość:

Obowiązkowa właściwość logiczna formantu „DBPATH”		
Nazwa właściwości=Wartość	Opis właściwości	Przykład
TABLE=„<Nazwa SQL tabeli>”	Nazwa tabeli, w której chcemy wybrać pole. Właściwość typu „String”.	TABLE=amAsset

Formant LinkEdit

Formant „LINKEDIT” tworzy formant do wprowadzania łącza w bazie danych AssetCenter.

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich formantów, formant „LINKEDIT” posiada następujące właściwości:

Obowiązkowa właściwość logiczna formantu „LINKEDIT”		
Nazwa właściwości=Wartość	Opis właściwości	Przykład / Uwagi
TABLE=„<nazwa SQL tabeli>”	Nazwa tabeli, w której chcemy wybrać łącze. Właściwość typu „String”.	TABLE=amAsset
FILTER=<warunek WHERE zapytania AQL>	Definiuje filtr na podstawie wyniku zapytania AQL. Właściwość typu „String”.	Ta właściwość jest opcjonalna.

Formant TextBox

Formant „TEXTBOX” (pole tekstowe) tworzy formant do wprowadzania tekstu.

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich formantów, formant „TEXTBOX” może mieć następujące właściwości:

Właściwości fizyczne formantu „TEXTBOX”		
Nazwa właściwości=Wartość	Opis właściwości	Przykład
MULTILINE=<Liczba>	Właściwość ta ma wartość „0”, jeśli formant „TEXTBOX” jest jednowierszowy, lub wartość numeryczną określającą procent wyświetlanej wysokości formantu, jeśli formant jest wielowierszowy.	MULTILINE=50

Przykład tworzenia kreatora

Aby zilustrować część teoretyczną programowania kreatora, utworzymy kreator „Przeniesienie”. W połączeniu z działaniem typu „Baza danych”, upraszcza on proces przenoszenia użytkownika i skojarzonych zasobów z jednej lokalizacji do innej. Szczegółowo opisaliśmy każdy etap tworzenia kreatora. Zapraszamy do samodzielnego utworzenia kreatora i sięgania do tej sekcji w razie wystąpienia problemów.

- Przykład tworzenia kreatora
- Krok 1: Analiza potrzeb
- Krok 2: Określenie organizacji kreatora

Krok 1: Analiza potrzeb

Zadaniem tego kreatora jest przeniesienie zasobów z jednej lokalizacji do innej. W tym celu musimy:

- ↳ Określić przenoszone zasoby.
- ↳ Wybrać nową lokalizację dla tych zasobów.

Jak określić przenoszone zasoby

Istnieją trzy sposoby określania przenoszonych zasobów:

- Określenie ich według użytkownika. Po wybraniu użytkownika, należy wybrać przenoszone zasoby.
- Bezpośrednie określenie przenoszonych zasobów przez wybranie ich rekordów w tabeli zasobów.
- Identyfikacja przenoszonych zasobów według lokalizacji. Najpierw trzeba wybrać lokalizację, a następnie zasoby przenoszone z tej lokalizacji.

Należy zatem utworzyć stronę z opcjami do wyboru dla użytkownika, umożliwiającą użytkownikowi określenie metody wyboru przenoszonych zasobów.

Wybór nowej lokalizacji

Aby wybrać nową lokalizację dla zasobów, należy po prostu wybrać rekord z tabeli lokalizacji.

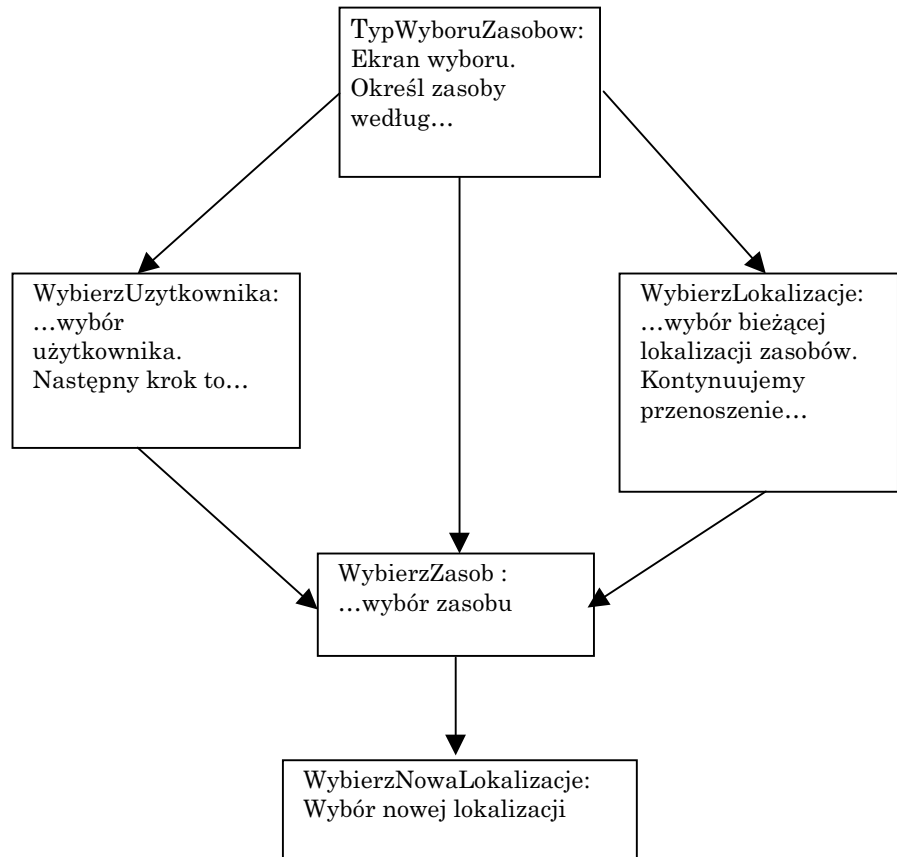
Krok 2: Określenie organizacji kreatora

Zgodnie z potrzebami określonymi w kroku 1, należy zdefiniować organizację kreatora. Tzn:

- Liczbę stron.
- Powiązania między poszczególnymi stronami.
- Zawartość każdej z tych stron.

Wiemy z kroku 1, że należy utworzyć stronę z opcjami do wyboru. Ta strona będzie pierwszą stroną kreatora. Nazwiemy ją „TypWyboruZasobow”.

Następnie, na podstawie następnego diagramu, określamy sposób definiowania kreatora:



Na podstawie tego schematu, możemy zdefiniować przejścia dla kolejnych stron:

Strona	Może prowadzić do stron
TypWyboruZasobow	WybierzZasob, WybierzUzytkownika, WybierzLokalizacje
WybierzZasob	WybierzNowaLokalizacje
WybierzUzytkownika	WybierzZasob
WybierzLokalizacje	WybierzZasob
WybierzNowaLokalizacje	Brak

Następnie definiujemy zawartość stron. Oznacza to formanty umożliwiające użytkownikowi dokonanie wyboru:

Strona	Zadanie tej strony	Stosowany formant
Typ Wyboru Zasobow	Pozwala użytkownikowi wybrać jedną z trzech możliwości	Formant „CHOICE”
Wybierz Zasob	Pozwala użytkownikowi wybrać zasoby z listy rekordów w tabeli zasobów.	Formant „ADBLIST”
Wybierz Uzytkownika	Pozwala użytkownikowi wybrać z tabeli wydziałów i pracowników użytkownika, którego zasoby będą przenoszone.	Formant „ADBLIST”
Wybierz Lokalizacje	Pozwala użytkownikowi wybrać bieżącą lokalizację z tabeli lokalizacji.	Formant „ADBLIST”
Wybierz Nowa Lokalizacje	Pozwala użytkownikowi wybrać nową lokalizację z tabeli lokalizacji.	Formant „ADBLIST”

Krok 3: Zapis struktury kreatora za pomocą języka skryptowego

Etap ten polega na napisaniu skryptu kreatora. Aby to zrobić, należy opisać strukturę wszystkich węzłów kreatora. Kod źródłowy kreatora Przeniesienie z komentarzami podano poniżej. Jest to jeden z wielu sposobów napisania kreatora. Wiele innych kreatorów może wykonać to samo zadanie.

```

;=====
;(c) Peregrine Systems 1999
;=====
NAME = "Move"
TITLE = "Move user"
VERSION = "699"
;=====
=====
;Ask which user to move. By default, use selection in amEmplDept if
context is on this table
;=====
=====
{ PAGE pgUser
  TITLE = "Choose the persons who are moving"
  { DBLISTBOX Users
    COLNAME = "Name|FirstName"
    COLWIDTH = "50|50"
  }
}

```

```

DBLCLICK = 1
LABEL = "Persons to move"
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 = "Choose the new location"
  { STRING UserName
    VALUE = AmDbGetString("SELECT FirstName + ' ' + Name FROM
amEmplDept WHERE lEmplDeptId IN (" & {pgUser.Users} & ")" )
  }
  { LABEL LABEL1
    CAPTION = "User(s): " & {UserName}
  }
  { DBLISTBOX NewLocId
    COLNAME = "Name"
    COLWIDTH = "100"
    DBLCLICK = 1
    TABLE = "amLocation"
    VALUE = "-1"
  }
  { TRANSITION trNewLocToAssets
    TO = "pgRecap"
  }
}
}

;=====
;=====
;Recapitulation
;=====
;=====
{ PAGE pgRecap
  TITLE = "Recapitulation"
  { LISTBOX Users
    COLTITLE = "Name"
    COLWIDTH = "100"
    LABEL = "Persons to move"
  }
}

```

```

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("Moving the employee " +
{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("The transaction has been cancelled")

```

```
RetVal = 1
Exit function
}
SUMMARY = 1
}
```

Korzystanie z edytora graficznego

AssetCenter umożliwia tworzenie kreatorów za pomocą zintegrowanego edytora graficznego. Zadaniem tego edytora jest przyspieszenie i uproszczenie procesu tworzenia kreatora. Nie zastępuje on języka skryptowego, którego zrozumienie ma podstawowe znaczenie przy projektowaniu kreatorów.

- Omówienie interfejsu
- Tworzenie nowego węzła
- Edycja właściwości węzła
- Kompilacja, uruchamianie i usuwanie błędów z kreatora

Aby skorzystać z edytora graficznego, tworzone lub modyfikowane działanie musi być typu „Kreator” (nazwa SQL: seActionType).

Omówienie interfejsu

Aby uzyskać dostęp do edytora graficznego, wybierz polecenie menu **Narzędzia/ Działania/ Edytuj**. Edytor graficzny zostanie wyświetlony w zakładce **Kreator** szczegółów działania. Składa się on z trzech części:





- Pasek narzędzi z najczęściej używanymi funkcjami.
- Sekcja **Hierarchia** z widokiem drzewa struktury kreatora.
- Sekcja zawierająca właściwości węzła wybranego w hierarchii.

Pasek narzędzi

Pasek narzędzi umożliwia bezpośrednie uruchamianie poleceń edycji. Po krótkim zatrzymaniu wskaźnika myszy nad przyciskiem wyświetlana jest etykieta opisująca ikonę.

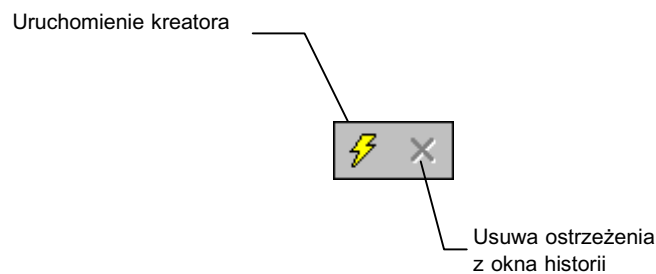
Polecenia edycji

Dostępne są cztery polecenia edycji:

-  przelacza edytor pomiedzy trybem tekstowym i trybem graficznym.
-  przenosi węzeł o jeden poziom wyżej wewnątrz węzła zwierzchniego.
-  przenosi węzeł o jeden poziom niżej wewnątrz węzła zwierzchniego.
-  usuwa wybrany węzeł.

Polecenia wykonywania i usuwania błędów

Polecenia te pozwalają kompilować, usuwać błędy i wykonywać skrypt:



Narzędzie do wyszukiwania

Na pasku narzędzi umieszczony jest narzędzie wyszukiwania, za pomocą którego można wyszukać ciąg znaków w strukturze drzewa kreatora (skrót klawiszowy „Ctrl+F” uruchamia ten formant bezpośrednio):

Po kliknięciu tego przycisku należy wprowadzić wyszukiwany tekst. Jeśli wyszukiwanie zakończy się powodzeniem, AssetCenter zaznacza znalezione wystąpienie.

(Skróty klawiszowe „F3” i „Shift+F3” służą do wyszukiwania następnego i poprzedniego wystąpienia szukanego tekstu.

W trybie tekstowym wyszukiwanie jest pełnotekstowe. W trybie graficznym wyszukiwanie dotyczy tylko nazwy właściwości.

Widok drzewa kreatora

W lewej części edytora graficznego znajduje się widok drzewa kreatora: Po wybraniu węzła w strukturze kreatora, AssetCenter wyświetla właściwości związane z tym węzłem w prawej części ekranu.

Lista właściwości wybranego węzła.

W prawej części ekranu można wprowadzać wartości właściwości węzła: Każda właściwość ma ustaloną wartość lub skrypt. Stosowane są następujące oznaczenia barwne:

- Jeśli właściwość korzysta ze swej domyślnej wartości, jej nazwa i wartość jest wyświetlana w kolorze szarym. Właściwości można nadać inną wartość. Zostanie ona wyświetlona w kolorze czarnym.
- Jeśli właściwość korzysta z wartości zdefiniowanej przez użytkownika lub ze skryptu, jej nazwa i wartość wyświetlane są w kolorze czarnym.
- Jeśli właściwość jest obowiązkowa, jej nazwa i wartość wyświetlane są w kolorze czerwonym.

Tworzenie nowego węzła

W tej sekcji opisano czynności, które można wykonać w odniesieniu do węzła. Pasek narzędzi pozwala przesunąć węzeł w górę lub w dół albo usunąć go. Omówimy tworzenie nowego węzła.

Węzeł można przesunąć w górę lub w dół, albo usunąć za pomocą menu kontekstowego. Aby otworzyć menu kontekstowe, należy kliknąć węzeł prawym przyciskiem myszy.

Aby utworzyć węzeł, należy najpierw wybrać węzeł zwierzchni. Na przykład, aby utworzyć nowy węzeł „Page”, należy najpierw kliknąć węzeł „Root”. Po wybraniu węzła zwierzchniego, należy kliknąć prawym przyciskiem myszy, aby otworzyć menu kontekstowe. W poleceniu menu „Nowy” zebrane są węzły, które można utworzyć:

AssetCenter umieści węzeł w strukturze drzewa kreatora.

Edycja właściwości węzła

Po utworzeniu węzła, można przypisywać wartości jego właściwościom. Umożliwia to prawa część edytora.

Istnieją dwa sposoby określenia wartości właściwości:

- Wprowadzenie wartości stałej
- Zdefiniowanie skryptu

Skrypt zawsze ma pierwszeństwo nad wartością stałą. Jeśli do właściwości przypisany jest skrypt i wartość, AssetCenter zignoruje wartość stałą i zinterpretuje skrypt.

Przypisanie wartości stałej do właściwości


Kliknij kolumnę „Wartość” dotyczącą danej właściwości. Zgodnie z typem danych przyjmowanym przez właściwość (tekst, wartość logiczna, wartość numeryczna o podwójnej precyzji, itp.), AssetCenter umożliwia wybór z listy dopuszczalnych wartości lub wypełnienie pola tekstowego.

Przypisanie skryptu do właściwości

Kliknij właściwość, do której ma być przypisany skrypt. Sam skrypt należy wprowadzić w polu **Skrypt** pod listą właściwości.

Po wybraniu pozycji **Przywróć wartość domyślną** z menu kontekstowego (kliknięcie właściwości prawym przyciskiem myszy), AssetCenter anuluje wartość stałą lub skrypt i przywraca właściwości jej wartość domyślną. Operacja ta jest możliwa jedynie w przypadku właściwości, których wartość lub skrypt zostały zdefiniowane przez użytkownika (właściwości te wyświetlane są w kolorze czarnym).

Kompilacja, uruchamianie i usuwanie błędów z kreatora

Kreator można uruchomić, klikając przycisk  na pasku narzędzi edytora. Wszelkie błędy występujące podczas wykonywania wyświetlane są w oknie historii błędów (dostępnym przez zintegrowany debugger kreatora). Naciskając przyciski Shift+F9 można przerwać wykonywanie (jeśli kreator jest modalny) i uruchomić debugger.

W ten sposób można łatwo naprawić i poprawić błędy występujące w kreatorze.

Przycisk uruchomienie jest niedostępny, jeśli kreator jest kontekstowy.

Często zadawane pytania

W tym rozdziale podajemy odpowiedzi na pytania, które mogą się pojawić podczas tworzenia kreatora.

Pytanie

Poniższy kod nie działa:

```
{lbxMyListBox.Values.Count}
```

Odpowiedź

W składni metody należy wprowadzić nawias otwierający i zamykający. Oto prawidłowy kod:

```
{lbxMyListBox.Values.Count ()}
```

Pytanie

Poniższy kod nie działa:

```
{lbxMyListBox.Line (lRow)}
```

Odpowiedź

Metoda „LINE” związana jest z właściwością „VALUES” formantu „LISTBOX”. Oto prawidłowy kod:

```
{lbxMyListBox.Values.Line(lRow)}
```

Pytanie

Poniższy kod nie działa:

```
{lbxMyListBox.Values.Line({lbxTmp})}
```

Odpowiedź

W metodzie nie można zastosować odwołań do właściwości. Należy to zapisać tak:

```
Dim lRow As Long  
lRow = {lbxTmp}  
{lbxMyListBox.Values.Line(lRow)}
```

Pytanie

Następujący kod, przypisujący wartość stałą do właściwości, nie działa:

```
{Property} = 123
```

Odpowiedź

Aby przypisać wartość do właściwości, należy użyć funkcji „AmSetProperty()”, jak poniżej:

```
Dim irc as Integer  
irc= AmSetProperty("Property", 123)
```

Nie należy zapominać o pobraniu kodu powrotu funkcji (w tym przykładzie „irc”), nawet jeśli nie będzie on wykorzystywany.

Pytanie

Podczas uruchamiania kreatora tworzącego zasób w bazie danych, pojawia się następujący błąd:

```
12001 - You do not have write-access rights
```

Komunikat ten jest wyświetlany nawet wtedy, gdy użytkownik jest połączony jako administrator.

Odpowiedź

Komunikat ten jest wyświetlany przy próbie uzyskania dostępu z zapisem poza węzłem „FINISH.DO” kreatora. Kreator wykonuje następujące czynności:

- Zbiera informacje za pomocą szeregu stron (dostęp z zapisem jest zabroniony nawet dla administratora AssetCenter)
- Wykonuje skrypt zapisany w węźle „FINISH.DO” (dostęp do zapisu dozwolony zgodnie z uprawnieniami użytkownika)

Pytanie

Komunikaty o błędach wyświetlane podczas działania kreatora są niekiedy niekompletne.

Odpowiedź

Naciśnij SHIFT+F9, aby wyświetlić debugger. Komunikaty o błędach w oknie historii często są bardziej czytywiste.

Pytanie

Przy zastosowaniu formantu „DBLISTBOX” w stronie kreatora spada wydajność. Czy to normalne?

Odpowiedź

Problem ten występuje przy zastosowaniu formantu „DBLISTBOX” w połączeniu z filtrem. W takim przypadku, przy każdej zmianie wyboru wysyłane jest zapytanie do bazy danych w celu sprawdzenia, czy wybór jest zgodny z filtrem. Dodatkowe zapytanie nie jest wykonywane, gdy wybór jest dokonywany przez użytkownika.

Pytanie

Jak umożliwić lub uniemożliwić edycję niektórych kolumn w formancie „LISTBOX”.

Odpowiedź

Należy użyć właściwości „EDITABLE” tego formantu. Wartość przypisana do tej właściwości jest ciągiem znaków składającym się z „0” i znaków oddzielonych separatorem „|”, który oddziela kolumny. Znak „0” definiuje kolumnę jako „nieedytowalną”, „1” jako „edytowalną”. Jeśli wartość zostanie pominięta, związana z nią kolumna nie będzie edytowalna, w poniższym przykładzie możliwa jest edycja tylko kolumny 2 i 4:

```
EDITABLE = "|1||1"
```

Pytanie

Jak sprawić, aby kreator otwierał okno szczegółów?

Odpowiedź

Trzeba zastosować wywołania DDE (za pomocą funkcji) wewnątrz kreatora. Kreator nie może być modalny. Oto przykład otwierania tabeli zasobów z wewnątrz kreatora:

```
Dim irc as Long  
irc = AmActionDDE("aam", "AssetCenter", "OpenTable(amAsset) ")
```

Pytanie

Jaka jest różnica między właściwościami „COLNAME” i „COLTITLE” formantu „LISTBOX”?

Odpowiedź

Tytuły kolumn w formancie „LISTBOX” mogą być zdefiniowane automatycznie lub ręcznie:

- Właściwość „COLNAME”, związana z właściwością „TABLE” umożliwia automatyczne określenie tytułów kolumn w formancie „LISTBOX” za pomocą etykiet pól z bazy danych.
- Właściwość „COLTITLE”, jeśli jest wypełniona, wymusza tytuły kolumn. Jeśli ta właściwość nie jest zdefiniowana, tytuły kolumn będą zgodne z definicją właściwości „COLNAME”.

Zatem poniższy przykład:

```
...  
TABLE = "amEmplDept"  
COLNAME = "Nazwisko||Imię"  
COLTITLE = "|A|B"  
...
```

wyświetla następujące etykiety kolumn formantu „LISTBOX”: Nazwisko, A, B.

Właściwość „COLNAME” określa również typ formantu używanego, gdy możliwa jest edycja wartości kolumn.

Rozdział 24 - Dostrajanie AssetCenter do stosowania w środowisku WAN

Sieci WAN (sieci rozległe) mają często następujące cechy:

- Niska przepustowość.
- Długi czas oczekiwania.

Można skonfigurować AssetCenter tak, aby zminimalizować oddziaływanie tych warunków. Jednakże, konfiguracje te odbijają się negatywnie na niektórych funkcjach AssetCenter.

W tym rozdziale przedstawiono wskazówki pomocne w dostosowaniu aplikacji do ograniczeń sieci WAN. Tym niemniej jest bardzo ważne, aby przeprowadzić próby celem stwierdzenia, czy kompromis pomiędzy szybszymi czasami reakcji, a utratą funkcjonalności będzie opłacalny.

Opcje w menu *Narzędzia/ Opcje*

Można ograniczyć długość dostępu do bazy danych za pomocą następujących opcji:

- Opcja **Samouzupełnianie po** w zakładce **Nawigacja**: Można wyłączyć samouzupełnianie lub przypisać wyższą wartość (na przykład, samouzupełnianie po 10 000 ms).
- Opcja **Drzewa w listach rozwijanych** w zakładce **Nawigacja**: Można wyłączyć tę opcję, ponieważ widoki typu drzewo bardziej obciążają wydajność niż widoki typu lista.
Jednakże traci się przyjazną dla użytkownika cechę widoku typu drzewo w listach rozwijanych.

Można ograniczyć wymianę informacji pomiędzy stacją roboczą i serwerem bazy danych dostosowując następujące opcje:

- Opcje **Nie wczytuj przez czas dłuższy, niż** i **Nie wczytuj więcej, niż** w zakładce **Listy** (listy główne lub inne): Zalecamy ograniczenie liczby wczytywanych linii (na przykład, można określić maksymalną wartość 50 linii w głównych listach i 15 w listach zakładek). Decyzja o liczbie wczytywanych linii zależy od użytkownika, zgodnie z filtrami stosowanymi do wyświetlanych list i prawdopodobieństwa znalezienia żądanych informacji w danej liczbie linii.
- **Sprawdzaj nowe wiadomości**, zakładka **Poczta**: Można skonfigurować tę opcję tak, aby sprawdzanie nowych wiadomości było wykonywane tylko przy połączeniu z bazą danych lub wyznaczyć odstępy, w jakich ma być wykonywane sprawdzanie (na przykład co 10 minut).
- Zakładka **Pamięci podręczne**: Można zwiększyć odstępy określające odświeżanie pamięci podręcznych (kolumna **Co**) lub nawet wyłączyć odświeżanie pamięci podręcznych podczas sesji. W tym przypadku pamięci podręczne będą odczytywane tylko podczas nawiązywania połączenia z bazą danych.
Jeśli pamięci podręczne nie będą odświeżane regularnie, może się zdarzyć, że wyświetlane dane nie będą aktualne. Jednakże, większość danych przechowywanych w pamięci podręcznej jest generowana podczas instalacji AssetCenter i nie podlega regularnym zmianom (listy pozycji, słownik cech, waluty, kalendarze dni roboczych, itp.).

Sortowanie list

Główne listy i listy zakładek można skonfigurować za pomocą polecenia menu **Konfiguruj listę** w kilku przypadkach:

- listy wyświetlane w menu dające dostęp do tabel (na przykład, polecenie menu **Repozytorium/ Zasoby**),
- listy wyświetlane przez widoki (polecenie menu **Narzędzia/Widoki**),
- listy wyboru (polecenie menu kontekstowego **Wybierz łącze**).

Listy te można sortować na kilka sposobów:

- wybierając własne warunki sortowania (kolumna **Sortuj**),

- używając predefiniowanych indeksów (pole **Sortuj według indeksu**).

Wydajność tych dwóch opcji bywa różna. Trudno przewidzieć, która opcja oferuje najlepsze rozwiązanie.

Należy wypróbować oba rozwiązania dla każdej listy w AssetCenter przed podjęciem decyzji, która metoda jest lepsza dla konkretnej bazy danych.

Optymalizacja ekranów

Aby skrócić czas reakcji aplikacji, można ograniczyć liczbę elementów danych wyświetlanych na ekranie, wyświetlając tylko te kolumny, listy i zakładki, które są niezbędnie potrzebne.

Pamięć podręczna połączenia

Można również uaktywnić pamięć podręczną połączenia, za pomocą zakładki **Pamięć podręczna** na ekranie szczegółów połączenia:

Uaktywnienie pamięci podręcznej połączenia:

- Skraca czas połączenia z bazą danych.
- Zaoszczędza również czas, jeśli stosowane są obrazy i ikony.

Z zasady jednak domyślne rozmiary pamięci podręcznych są dobrze zoptymalizowane.

Ograniczenia dostępu

Wyświetlanie okien szczegółów i list jest spowolnione, gdy występują ograniczenia dostępu dla danej nazwy logowania. Jest to spowodowane wykonywaniem przez AssetCenter sprawdzenia przed wyświetleniem danych.

W razie wątpliwości, należy wyświetlić listę lub okno szczegółów używając nazwy logowania nie obłożonej ograniczeniami i porównać prędkość wyświetlania.

Należy odpowiednio usunąć ograniczenia dostępu.

Używanie konfiguracji jednej stacji roboczej w innych

Po zoptymalizowaniu wydajności jednej stacji roboczej, trzeba powielić zmiany konfiguracji na inne klienckie stacje robocze.

Dla zaoszczędzenia czasu, można skopiować pliki „*.ini” stosownie do wykonanych modyfikacji.

Szczegółowa lista, zawartość i lokalizacja plików „*.ini” znajduje się w podręczniku pt. „Opis funkcji systemu AssetCenter: Administracja i zaawansowane zastosowania AssetCenter”, rozdział „Pliki .ini”.

Rozdział 25 - Pliki .ini

Z programami AssetCenter związane są pliki konfiguracyjne o rozszerzeniu **.ini**.

W niniejszym rozdziale omówiono następujące zagadnienia:

- Lista dostępnych plików .ini
- Edycja plików .ini

Lista dostępnych plików .ini

W następnym tabeli podano listę dostępnych plików **.ini**:

Program	Plik „.ini”	Opis
AssetCenter aam32.exe aam16.exe	aamdisk30.ini	Opcje wyświetlania dla użytkownika. Aby przywrócić domyślne ustawienia dla wszystkich okien, usuń ten plik.
	aam.ini	Opcje użytkownika AssetCenter.
AssetCenter Database Administrator amdba32.exe	amdba32.ini	Opcje użytkownika AssetCenter Database Administrator. Opcje wyświetlania dla użytkownika.
AssetCenter Export amexp32.exe amexpl32.exe	amexp32.ini amexpl32.ini	Opcje użytkownika aplikacji AssetCenter Export. Opcje wyświetlania dla użytkownika.
Serwer AssetCenter Aamsrv32.exe	amsrv.ini	Opcje użytkownika aplikacji Serwer AssetCenter. Opcje wyświetlania dla użytkownika.
	amsrvcf.ini	Parametry aplikacji Serwer AssetCenter jako serwera sieci WWW.
Serwer WWW AssetCenter	amw3.ini	Parametry aplikacji Serwer WWW AssetCenter.
Aamapi30.dll	aamapi30.ini	Opcje użytkownika.
Wszystkie powyższe programy	amdb.ini	Lista połączeń bazy danych.

Pliki **.ini** znajdują się w folderze instalacyjnym systemu Windows, za wyjątkiem następujących plików:

- plik „amsrvcf.ini” znajduje się w folderze podrzędnym „amsrv/bin32” foldera instalacyjnego AssetCenter.
- plik „amw3.ini” znajduje się w folderze podrzędnym „webserv/bin32” foldera instalacyjnego AssetCenter.

Ustawienia zapisane w danym pliku **.ini** dotyczą każdej stacji roboczej AssetCenter z osobna.

Edycja plików .ini

Pozycje plików **.ini** można modyfikować:

- Za pomocą programu: są one zapisywane albo po zatwierdzeniu modyfikacji, lub po wyjściu z programu. W tym drugim przypadku, jeśli działanie aplikacji nie zostanie zakończona za pomocą polecenia menu „Plik/ Zakończ”, modyfikacje nie zostaną zapisane.
- I/ lub ręczne.

Wszędzie tam, gdzie to możliwe, lepiej jest modyfikować pozycje plików **.ini** za pomocą AssetCenter.

Jednakże niektóre pozycje można tworzyć i modyfikować jedynie ręcznie.

Ostrzeżenie: Modyfikowanie plików **.ini** może być trudnym zadaniem, które powinien wykonywać ktoś o odpowiednim doświadczeniu.

W następnym tabeli opisano pliki **.ini**, które można modyfikować wyłącznie ręcznie.

Ostrzeżenie: W następnym tabeli opisano tylko niektóre pozycje plików **.ini**. Nie jest ona wyczerpująca. Sekcji i pozycji, których nie opisano, w żadnym wypadku nie można modyfikować ręcznie.

Niektóre pozycje typu boolean opisane są wartościami „1” lub „0”. Wartość „1” można zastąpić słowem „True” a „0” słowem „False”.

Pozycje pliku Aam.ini

Sekcja [OPTION]

Pozycja	Znaczenie
BsaveOptionOnExit	Wpisz „0” jeśli chcesz, aby modyfikacje sekcji [Option] wprowadzane za pomocą programu nie były zapisywane przy zamykaniu AssetCenter. Domyślnie, modyfikacje są zapisywane.
CAdbLinkEdit.opt_bCreation	Umożliwia lub uniemożliwia tworzenie rekordów w trybie szybkim: 0: tworzenie „szybkie” zabronione. 1: tworzenie „szybkie” dozwolone. Wartość domyślna: 1.
CmdComboLines	Ogranicza liczbę linii wyświetlanych w listach i działań dostępnych za pomocą paska narzędzi.
CNtbkTabCfg.bShowFlyby	Wyświetla etykiety w zakładkach na ekranach szczegółów: 0: nie. 1: tak.
KeyIniFileName	Określa ścieżkę pliku „aamdisk30.ini”. Przykład: KeyIniFileName=aamdisk30.ini Opcja ta jest przydatna, jeśli wszyscy użytkownicy AssetCenter mają korzystać z tego samego pliku „aamdisk30.ini”, znajdującego się na przykład na dysku sieciowym. W tym przypadku, można nadać temu plikowi atrybut tylko do odczytu aby uniemożliwić użytkownikom zmianę konfiguracji.
NewMailLastCheck	Czas ostatniego odczytania wiadomości AssetCenter. Jednostka: liczba sekund od godziny 00:00 1 stycznia 1970.
opt_bAskForConcurrentModifications	Pozycja ta określa, czy AssetCenter powinien pytać o potwierdzenie po kliknięciu polecenia „Modyfikuj” jeśli inny użytkownik modyfikuje ten sam rekord: 1: Wyświetla okno dialogowe z potwierdzeniem. 0: Nie wyświetla okna dialogowego z potwierdzeniem i automatycznie zapisuje modyfikacje.

Pozycja	Znaczenie
opt_bCommitDeletesOneByOne	Opcja ta jest użyteczna przy usuwaniu grupy rekordów. Jeśli jest włączona, AssetCenter usuwa rekordy pojedynczo (jedna transakcja – jeden usunięty rekord). W przeciwnym razie AssetCenter usuwa wszystkie rekordy przy jednej transakcji. Wartość domyślna: 0.
opt_ImportCacheSize	Przy importowaniu danych za pomocą kluczy uzgodnień, pozycja ta wskazuje rozmiar pamięci podręcznej, który pozwoli poprawić wydajność importu. Jednostka: Liczba znalezionych rekordów. Wartość domyślna: 100.
StartSunday	Określa, czy tydzień rozpoczyna się w poniedziałek (StartSunday=0), czy w niedzielę (StartSunday=1). Opcja ta jest używana na poziomie kalendarza.

Sekcja [SQL]

Pozycja	Znaczenie
OracleDLL	Pozwala określić nazwę wczytywanej biblioteki Oracle DLL umożliwiającej wymianę danych z aplikacją Oracle.

Pozycje pliku „amsrv.ini”

Sekcja [OPTION]

Pozycja	Znaczenie
MaxRentPerTrans	Pozycja ta pozwala generować splatę. Określa ona maksymalną liczbę obliczeń splat na transakcję. Wartość domyślna: 200.
MaxMsgInList	Określa liczbę linii wyświetlanych na liście w oknie głównym aplikacji Serwer Asset Center. Wartość domyślna: 5000.
<Moduł>LastCheck gdzie <Moduł> może przyjmować następujące wartości: Alarms, CostCenter, History, LostVal, Rent, Stats, Stock, TimeZone, UpdateToken, WkGroup, WkGroup <xxx>, WorkflowFinder	Linie z przyrostkiem „LastCheck” są oznaczeniem ostatniej daty uruchomienia modułu. Umożliwiają one wyliczenie następnej daty uruchomienia modułu przy ponownym uruchomieniu aplikacji Serwer Asset Center. Linie „WkGroup <xxx> LastCheck” można usunąć, jeśli grupa wykonawcza <xxx> już nie istnieje (lub linię „WkGroupLastCheck”, jeśli nie ma żadnych schematów workflow bez grupy wykonawczej), ponieważ program nie robi tego automatycznie.

Pozycje plików „amsrvcf.ini” i „amw3.ini”

Pozycje w pliku „amsrvcf.ini” są podobne do pozycji w pliku „amw3.ini”.
Dalszych informacji należy szukać w podręczniku pt. „Web User's Guide”, rozdział „Adminstrating AssetCenter Web”, sekcja „Configuring AssetCenter Web via „amw3.ini”.

Pozycje pliku „amexp32.ini”

Sekcja [OPTION]

Pozycja	Znaczenie
MaxOldDoc	Maksymalna liczba ostatnio używanych plików wyświetlanych w menu „Plik”.

Pozycje pliku „amdb.ini”

W każdej sekcji odpowiadającej części programu AssetCenter można modyfikować następujące pozycje:

Pozycja	Znaczenie
AmApiDll	Określa ścieżkę biblioteki DLL „aamapi30.dll”, tzn. biblioteki interfejsu programowania aplikacji (API) AssetCenter. Pozycja ta jest przydatna przy uruchamianiu serwera WWW AssetCenter, gdy plik „aamapi30.dll” nie znajduje się w folderze systemowym Windows NT, ani w folderze pliku „amw3.exe”.
FetchingArraySize	Liczba linii pobieranych przez pakiet przy wykonywaniu instrukcji SQL. Wartość domyślna: 80.
OdbcLockingTime	W przypadku bazy danych aplikacji Microsoft SQL Server lub Sybase SQL Anywhere określa czas, po którym rekord jest uznawany za zablokowany przez innego użytkownika. Jednostka: sekundy. Wartość domyślna: 60. Ostrzeżenie: Jeśli wartość ta będzie zbyt mała, import może być przerywany na obciążonych serwerach.
OldStyleCatalog	W przypadku bazy danych Oracle, pozycja ta umożliwia wymuszenie widoku „Tab” zamiast domyślnie używanego widoku „All_Catalog”. Pozycja ta może być ustawiona na jedną z dwóch wartości: 1: używany jest widok „Tab”. 0: używany jest widok „All_Catalog”.
PostConnectSql	Podaje ciąg SQL uruchamiany natychmiast po połączeniu z bazą danych. Przykład zastosowania: Aby otworzyć bazę danych SQL Anywhere 5.0 z wersją wykonawczą SQL Anywhere 5.5, należy dodać następującą linię: PostConnectSql=SET OPTION NON_KEYWORDS='LOGIN' Na końcu każdej sekcji opisującej bazę danych wcześniejszą niż wersja 3.0.

