

Peregrine

AssetCenter

Zastosowania
zaawansowane



© Copyright 2002 Peregrine Systems, Inc.

Wszelkie prawa zastrzeżone.

Informacje zawarte w niniejszym dokumencie są własnością Peregrine Systems, Incorporated i mogą być używane lub ujawniane jedynie za pisemną zgodą firmy. Zabrania się reprodukcji niniejszego podręcznika lub dowolnej jego części bez uprzedniego uzyskania pisemnej zgody firmy Peregrine Systems, Inc.. W dokumentacji wymienione są liczne marki produktów. Większość z nich to znaki towarowe zastrzeżone przez ich właścicieli.

Peregrine Systems ® i AssetCenter ® są znakami towarowymi Peregrine Systems, Inc.

Oprogramowanie opisane w niniejszym podręczniku jest dostarczane w ramach umowy licencyjnej pomiędzy Peregrine Systems, Inc. a użytkownikiem końcowym i musi być używane z zachowaniem warunków umowy. Informacje zawarte w tym dokumencie mogą zostać zmienione bez uprzedzenia i nie wiążą w żaden sposób Peregrine Systems, Inc.. Skontaktuj się z działem obsługi klienta Peregrine Systems, Inc., aby otrzymać informacje na temat daty ostatniej wersji niniejszego dokumentu.

Nazwy firm i osób wymienione 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.

Informacje techniczne o produkcie lub dokumentację na temat licencjonowanego produktu można uzyskać w dziale Obsługi klienta Peregrine Systems, Inc. pod następującym adresem: support@peregrine.com.

Wszelkie informacje lub sugestie dotyczące niniejszego dokumentu prosimy kierować do Działu publikacji technicznych Peregrine Systems, Inc. pod następujący adres: doc_comments@peregrine.com.

Niniejsze wydanie dotyczy wersji 4.2.0 programu objętego umową licencyjną.

AssetCenter

Peregrine Systems, Inc.
Worldwide Corporate Campus and Executive Briefing Center
3611 Valley Centre Drive San Diego, CA 92130
Tel 800.638.5231 or 858.481.5000
Fax 858.481.1751
www.peregrine.com



Spis treści

Rozdział 1. Spisy	19
Spisy dowolne	19
Wartości spisów	20
Spisy otwarte	21
Spisy zamknięte	21
Spisy systemowe	21
Wartości w spisach systemowych	22
Rozdział 2. Zapis historii	23
Utworzenie rekordu	24
Modyfikacja pola w tabeli lub łącza 1 (na przykład: Użytkownik zasobu)	25
Dodawanie łącza n do innej tabeli (na przykład: Zasoby ujęte w umowach)	25
Usunięcie łącza n do innej tabeli	25
Modyfikacja łącza n do innej tabeli	26
Zachowywanie historii cech skojarzonych z rekordami	26
Dodawanie cechy	26
Usuwanie cechy	27
Modyfikowanie wartości cechy	27
Tworzenie, usuwanie lub modyfikowanie linii historii	27

Tworzenie linii historii	28
Rozdział 3. Zapytania w AQL	29
Przedstawienie	29
AQL	29
Zapytania w AssetCenter	31
Zalecenia dotyczące pisania zapytań AQL	32
Przedstawienie połączeń AQL	33
Przyczyna i użyteczność rekordów klucza głównego 0	34
Użycie wartości NULL	36
Wyrażenie Self	37
Funkcja currentUser	38
Spisy systemowe	39
Tabele hierarchiczne	40
Uproszczona notacja AQL	41
Sortowania i indeksy	43
Przykład	43
W jaki sposób wymusić indeksy	43
Porządek sortowania	44
Ostrzeżenia	45
Edytor zapytań	46
Omówienie	46
Uzyskiwanie dostępu do edytora zapytań	46
Tworzenie zapytania przy użyciu edytora zapytań	48
Pola używane w zapytaniach	50
Pisanie wyrażeń	50
Stałe	51
Składnia AQL	51
Konwencje	51
Składnia zapytań	52
Elementy zapytania	53
Warunek FROM	59
Warunek WHERE	60
Warunek GROUP BY	61
Warunek HAVING	62
Warunek ORDER BY	63
Warunek INSERT	63
Warunek UPDATE	64
Warunek DUPLICATE	65
Warunek DELETE	65
Odwołanie funkcji AQL	65
Funkcje AQL typu "Ogółem"	66
Funkcje AQL typu "Ciąg znaków"	67

Funkcje AQL typu "Data"	67
Funkcje AQL typu "Numeryczne"	69
Funkcje AQL typu "Test"	70
Przykłady zapytań	70
Aby porównać pole w głównej tabeli z wartością	71
Aby porównać łącze w głównej tabeli z innym łączem	71
Aby porównać łącze w głównej tabeli z wartością	71
Aby porównać według pola w tabeli połączonej z główną tabelą	72
Tabele hierarchiczne	72
Zapytanie łączące dwa warunki	72
Porównania pola z liczbami, datami lub tekstem	73
Zapytanie dotyczące cechy	73
Aby wyszukać rekordy zgodne z wyrażeniem	73
Aby wyszukać pole, które nie jest wypełnione	73
Aby wyszukiwać brak łącza	74
Zapytanie z aliasem	75
Rozdział 4. Formularze	77
Definicja formularza	77
Instalacja prekonfigurowanych formularzy w roboczej bazie danych	78
Import formularzy podczas tworzenia bazy danych	78
Import formularzy do istniejącej bazy danych	78
Tworzenie formularzy	79
Informacje podstawowe	79
Edytowanie formularzy i obiektów	80
Tekst stały	80
Wzory	81
Listy	81
Obrazy	81
Właściwości obiektów w formularzach	82
Położenie i rozmiar	82
Właściwości	82
Siatka wzorca	85
Ustawienia strony formularza	85
Nie można przenieść obiektów z obszarów nagłówka i stopki do głównego obszaru strony.	86
Identyfikowanie formularzy związanych z danym modulem	87
Rozdział 5. Raporty Crystal	89
Działanie i instalacja generatora raportów	89
Funkcjonowanie ogólne	89
Instalacja pełnej wersji lub runtime Crystal Reports	91

Instalacja prekonfigurowanych raportów Crystal Reports w roboczej bazie danych	91
Szczegóły raportu	93
Plik	93
W jaki sposób zmodyfikować raport Seagate Crystal	94
Statystyka Crystal Reports	94
Właściwość	94
Przycisk Automatyczne odświeżanie	95
Przycisk Zoom	95
Tworzenie raportu szczegółów	95
Przykład wykorzystania	95
Konfigurowanie raportów w Crystal Reports	95
Drukowanie raportu	97
Identyfikowanie raportów Crystal związanych z danym modulem	98

Rozdział 6. Działania 99

Definiowanie działań	99
Domena funkcjonalna	100
Tworzenie działania	101
Typy działań	101
Metoda ogólna	105
Wypełnianie zakładki DDE	106
Wypełnianie zakładki Poczta	108
Przykłady działań	110
Przykład działania typu "Wykonywalne"	111
Przykład działania typu "DDE"	111
Przykład działania typu Poczta	115
Przykład działania typu Skrypt	116
Używanie zmiennych	118
Specyfika motoru bazy danych Sybase SQL Anywhere	118
Testowanie działania	119
Przycisk Oblicz	119
Przycisk Wykonaj	119
Wykonywanie działania	119
Zaznaczenie wielokrotne w listach	120
Działania typu "Kreator"	120

Rozdział 7. Poczta 121

Omówienie poczty	121
W jaki sposób wysyłać wiadomości	122
W jaki sposób sprawdzać wiadomości	122
Potwierdzenie przyjęcia	123

Rozdział 8. Workflow	125
Definicje dotyczące workflow	125
Workflow	126
Czynność Workflow	127
Zdarzenie Workflow	127
Przejście workflow	127
Zadanie Workflow	128
Osoba przypisana do czynności workflow	128
Definicja grupy wykonawczej workflow	128
Omówienie	128
Główne tabele związane z workflow	130
Używanie graficznego edytora workflow	132
Czynności	132
Zdarzenia	133
Przejścia	134
Dodatkowa funkcjonalność	134
Jak realizować workflow	134
Przykład zastosowania workflow w akceptacji zapotrzebowania	135
Cel	135
Wymagania wstępne	137
Tworzenie czynności	142
Ustawianie parametrów dla zdarzeń utworzonych równocześnie z czynnościami	147
Tworzenie zdarzenia początkowego	148
Tworzenie przejść	149
Przykład aktywowania instancji workflow	149
Kontekst workflow	152
Definiowanie kontekstu instancji workflow	152
Obiekt odwołania instancji workflow	153
Ograniczanie instancji workflow w toku dla danego obiektu	153
Role workflow	154
Typ roli workflow	155
Definiowanie osoby przypisanej do czynności	156
Czynności workflow	156
Czynności typu Pytanie	157
Czynność typu Działanie użytkownika	158
Czynność typu Działanie automatyczne	159
Czynności typu Test/ Skrypt	160
Czynność Początek	161
Szablony czynności	161
Inicjowanie czynności	162
Zadania	162
Tworzenie zadań	162

Czynność typu działanie automatycznie lub test/ skrypt	163
Wyświetlanie listy zadań w toku	163
Wykonywanie zadania użytkownika	164
Przypisywanie zadań użytkownika	165
Administrowanie zadaniem workflow	165
Zdarzenia	165
Zdarzenie systemowe	165
Zdarzenie alarmowe	166
Zdarzenie użytkownika	166
Ogólne warunki aktywacji	169
Przetwarzanie zdarzenia	170
Zastosowanie: Realizacja synchronicznego schematu workflow	175
Zdarzenie końcowe	176
Przejścia workflow	177
Alarmy workflow i limity czasu	178
Limit czasu	178
Alarmy workflow	179
Grupy wykonawcze schematów workflow	180
Śledzenie workflow	180
Usuwanie zakończonych instancji workflow	181
Konieczność usuwania zakończonych instancji workflow	181
Automatyczne usuwanie zakończonych instancji workflow	181
Rozdział 9. Eksport danych i tworzenie widoków SQL	187
Definicje skryptu i zapytania eksportu	187
Skrypty eksportu	187
Zapytania eksportowe	188
Eksport danych z bazy danych AssetCenter	188
Eksport danych za pomocą skryptu eksportu	188
Eksport danych za pomocą polecenia menu kontekstowego Eksportuj listę	189
Zarządzanie widokami SQL w bazie danych AssetCenter	190
Zalecenia	190
Definiowanie skryptu eksportu	191
Metodologia	191
Definiowanie zapytań eksportowych	192
Format wyjściowy skryptu eksportu	195
Działania dotyczące widoków SQL	196
Wykonywanie skryptu eksportu	197
Wykonywanie skryptu eksportu z programu AssetCenter Export	197
Wykonywanie skryptów eksportowych w systemie DOS	198
Rozdział 10. Skrypty	201

Definicja skryptu	201
Omówienie	201
Informacje o tej wersji języka Basic	202
Notacja dostępu do danych	203
Zastosowanie skryptów	203
Wprowadzenie do funkcji	204
Definicja funkcji	204
Funkcje wbudowane i programowalne	205
Typy funkcji i parametrów	207
Klasyfikacja funkcji języka Basic	209
Pierwszy krok pisania skryptów	209
Przykładowy scenariusz	209
Krok 1: Utwórz cechę "Podręcznik"	210
Krok 2: Otwórz okno edycji	210
Krok 3: Analiza i definicja algorytmu	210
Krok 4: Napisz skrypt w języku Basic	211
Krok 5: Test skryptu Basic	211
Biblioteka skryptów	212
Pojęcia	212
Tworzenie biblioteki skryptów	212
Wywoływanie skryptu zapisanego w bibliotece skryptów	213
Wskazówki i ostrzeżenia	214
Uwaga na funkcje programowalne	214
Format stałych typu "Data+Czas" w skryptach	214
Format stałych typu "Czas trwania" w skryptach	215
Dostęp do spisu systemowego w trybie odczytu i zapisu	215
Łącze wirtualne "CurrentUser"	216
Komentarze do skryptu Basic	217
Inicjowanie komunikatu o błędzie	218
Pierwszy przykład	218
Opis problemu	218
Krok 1: Analiza i definicja algorytmu	219
Krok 2: Utworzenie skryptu Basic	220
Krok 3: Test skryptu Basic	220
Drugi przykład	221
Opis problemu	221
Krok 1: Analiza i definicja algorytmu	221
Krok 2: Utworzenie skryptu Basic	222
Krok 3: Test skryptu Basic	222
Rozdział 11. Kalendarze	223
Omówienie kalendarzy	223
Omówienie kalendarzy	223

Używanie kalendarzy	224
Wpływ kalendarzy na niektóre zakresy funkcjonalności	224
Metodologia tworzenia kalendarza	224
Opis tworzenia kalendarza	225
Wprowadzenie informacji ogólnych	225
Wypełnianie zakładki Harmonogram	225
Wypełnianie zakładki Wyjątki	227
Sprawdzanie kalendarza	230
Rozdział 12. Strefy czasowe	231
Dlaczego zarządzać strefami czasowymi?	231
Przykład	232
Wprowadzanie stref czasowych	232
Tworzenie stref czasowych	233
Zarządzanie strefą czasową	233
Format pola 'Czas letni'	233
Wartości argumentu <Year>	234
Wartości argumentu <DaylightInfo>	234
Przykład zarządzanie czasem letnim	237
Zarządzanie strefami czasowymi w programie AssetCenter Server	240
Testy do wykonania	240
Częstotliwość testu	241
Skutki dla różnych operacji	241
Tworzenie bazy danych	242
Połączenie z bazą danych	243
Import i eksport	245
Rozdział 13. Pola wyliczane	247
Definicja pola wyliczanego	247
Użyteczność pól wyliczanych	248
Tworzenie pola wyliczanego	248
Przedmowa	249
Metodologia tworzenia	251
Używanie pól wyliczanych	254
Używanie pola wyliczanego w konfiguracji listy	254
Filtrowanie rekordów tabeli	255
Odwoływanie się do pola wyliczanego	255
Rozdział 14. Kreatory	257
Zastosowane konwencje	257
Definicje	258
Twip	258

Kontrolka	258
Węzeł	259
Obiekt	259
Obiekt nadrzędny i obiekt podrzędny	259
Pełna nazwa obiektu	260
Zmienna	260
Przejsście	261
Model struktury	261
Model strony kreatora	263
Uwagi ogólne	263
Ogólna struktura i składnia	263
Właściwości węzła	264
Model deklaracyjny	265
Definicja stałej jako wartości właściwości	265
Odwoływanie się do właściwości	266
Definiowanie skryptu jako wartości właściwości	266
Metody stosowane we właściwościach	267
Właściwość typu tabela	268
Używanie zmiennych globalnych CurrentTable i CurrentSelection	269
Sekwencjonowanie kreatorów	270
Wykonanie	270
Parametry	270
Funkcje języka Basic	270
Definicja węzła 'Root'	271
Składnia węzła 'Root'	271
Właściwości węzła 'Root'	272
Węzły podrzędne węzła 'Root'	278
Definicja węzła 'Page'	279
Składnia węzła 'Page'	279
Właściwości węzła 'Page'	280
Węzły podrzędne węzła 'Page'	281
Definicja węzła 'Transition'	282
Składnia węzła 'Transition'	282
Właściwości węzła 'Transition'	282
Specyfika węzła 'Transition'	284
Dlaczego przejścia są definiowane w węźle "Root"?	284
Definicja węzła 'Finish'	284
Definicja węzła 'Start'	286
Definicja węzła 'Timer'	287
Definicja węzłów typu Long i String	288
Definicja węzła 'Control'	289
Ogólna składnia węzła typu "Control"	289
Typy kontrolek i ich właściwości	289
Wspólne właściwości kontrolek	289

Kontrolka CheckBox	294
Kontrolka ComboBox	294
Kontrolka OptionButtons	295
Kontrolka ListBox	296
Kontrolka Label	301
Kontrolka ProgressBar	301
Kontrolka CommandButton	302
Kontrolka DBListBox	303
Kontrolka DBQueryBox	307
Kontrolka DBEdit	310
Kontrolka DBTable	311
Kontrolka DBPath	311
Kontrolka LinkEdit	312
Kontrolka TextBox	314
Kontrolka CHART	315
Kontrolka FILEEDIT	317
Kontrolka TICKEDIT	317
Kontrolka CALENDAR	318
Kontrolka TIMESPANEDIT	318
Kontrolka NUMBOX	318
Kontrolka COMBOEDIT	319
Kontrolka DATETIMEEDIT	319
Przykład tworzenia kreatora	320
Krok 1: Analiza potrzeb	320
Krok 2: Określenie organizacji kreatora	321
Krok 3: Zapis struktury kreatora za pomocą języka skryptowego	323
Korzystanie z edytora graficznego	327
Omówienie interfejsu	327
Tworzenie nowego węzła	329
Edycja właściwości węzła	330
Kompilacja, uruchamianie i usuwanie błędów z kreatora	331
Często zadawane pytania	331
Pytanie	331
Odpowiedź	331
Pytanie	331
Odpowiedź	332
Pytanie	332
Odpowiedź	332
Pytanie	332
Odpowiedź	332
Pytanie	333
Odpowiedź	333
Pytanie	333
Odpowiedź	333

Pytanie	334
Odpowiedź	334
Pytanie	334
Odpowiedź	334
Pytanie	334
Odpowiedź	335
Pytanie	335
Odpowiedź	335
Rozdział 15. Zawiadomienia	337
Definicja zawiadomienia	337
Ogólne funkcjonowanie zawiadomień	338
Tworzenie zawiadomienia	338
Odczyt zawiadomienia	338
Priorytet zawiadomienia	338
Treść rozsyłanego zawiadomienia	338
Lista odbiorców zawiadomienia	339
Pole wyboru Do wszystkich grup pracowników (Nazwa SQL : bAllGroups)	339
Pole wyboru Z uwzględnieniem podgrup (Nazwa SQL : bChildGrps)	339
Wyświetlanie zawiadomienia	339
Aktywacja rolety zawiadomień.	340

Spis rysunków

1.1. Spisy - okno	20
3.1. Edytor zapytań - tryby tworzenia	46
4.1. Formularze - nagłówki	86
6.1. Działanie typu "wykonywalne" - okno szczegółów	111
6.2. "Wzórlistu.Doc"	112
6.3. Działanie typu "poczta" z obiektem odwołania - okno szczeółów	116
7.1. Poczta - funkcjonowanie ogólne	122
8.1. Workflow - schemat uproszczony	126
8.2. Workflow AssetCenter - widok ogólny	130
8.3. Workflow - główne tabele umożliwiające zdefiniowanie schematu	131
8.4. Workflow - główne tabele związane z instancją będącą w toku	131
8.5. Workflow - model zatwierdzania zapotrzebowań	136
8.6. Workflow - schemat zatwierdzania zapotrzebowań	149
8.7. Zakładka Parametry działania typu Baza danych	168
8.8. Przykład synchronicznego schematu workflow	175
8.9. Przykład asynchronicznego schematu workflow	176
8.10. Schemat workflow ze zdarzeniem końcowym	177
10.1. Skrypt - kompilator skryptu	206
11.1. Kalendarze - zakładka Harmonogramy	226

11.2. Kalendarz - zakładka Podgląd	230
14.1. Kreatory - model struktury	262
14.2. Kreator - przykład organizacji	322
14.3. Przycisk polecenia wykonywania i usuwania błędów . .	328

Spis tabel

1.1. Wartości w spisach systemowych	22
3.1. AQL- konwencje dotyczące składni	51
3.2. AQL - operatory logiczne	56
3.3. AQL- operatory porównania	56
3.4. AQL - funkcje typu Ogółem	66
3.5. AQL - funkcje typu Ciąg znaków	67
3.6. AQL - funkcje typu Data	67
3.7. AQL - przykłady funkcji typu Data	68
3.8. AQL - funkcje typu numerycznego	69
3.9. AQL - funkcje typu Test	70
8.1. Przypadki ograniczenia instancji workflow	169
8.2. Sposoby przetwarzania zdarzenia	170
8.3. Sposoby przetwarzania zdarzenia	173
10.1. Funkcje/parametry - typy	207
13.1. Typy pól wyliczanych	249
14.1. Zastosowane konwencje	257
14.2. Właściwości logiczne węzła "Root" (głównego)	272
14.3. Właściwości fizyczne węzła 'Root'	274
14.4. Węzły podrzędne węzła "Root" (głównego)	278
14.5. Właściwości logiczne węzła "Page" (strona)	280
14.6. Fizyczne właściwości węzła "Page" (strona)	280
14.7. Węzły podrzędne węzła "Page" (strona)	281

14.8. Logiczne właściwości węzła "Transition"	282
14.9. Właściwości logiczne węzła "Finish"	284
14.10. Właściwość fizyczna węzła 'Finish'	285
14.11. Właściwość logiczna węzła "Start"	286
14.12. Właściwość logiczna węzła "Timer"	287
14.13. Właściwość logiczna węzła typu Long lub String	288
14.14. Właściwości logiczne wspólne dla wszystkich kontroltek	289
14.15. Właściwości fizyczne wspólne dla wszystkich kontroltek	291
14.16. Właściwość kontrolki "CHECKBOX" (pole wyboru)	294
14.17. Właściwości fizyczne kontrolki "COMBOBOX" (lista kombinowana)	295
14.18. Właściwości fizycznie kontroltek "OPTIONBUTTONS" (przyciski opcji)	296
14.19. Właściwości fizyczne kontroltek "LISTBOX" (pole listy)	297
14.20. Metody kontrolki "LISTBOX"	299
14.21. Obowiązkowa logiczna właściwość kontrolki "LISTBOX" (lista)	300
14.22. Właściwości fizyczne kontrolki "LABEL"	301
14.23. Właściwości fizyczne kontrolka "PROGRESSBAR"	302
14.24. Właściwości fizyczne kontroltek "COMMANDBUTTON"	302
14.25. Właściwości fizyczne kontrolka "DBLISTBOX"	303
14.26. Właściwości fizyczne kontrolki "DBQUERYBOX"	307
14.27. Właściwości fizyczne kontrolki "DBEDIT" w trybie "Normal"	310
14.28. Obowiązkowa właściwość logiczna kontrolki "DBPATH"	312
14.29. Właściwość logiczna kontrolki "LINKEDIT"	312
14.30. Właściwości fizyczne kontrolki "TEXTBOX"	314
14.31. Właściwości logiczne kontrolki "CHART"	315
14.32. Właściwości fizyczne kontrolki "CHART"	315
14.33. Właściwości kontrolki "FILEEDIT"	317
14.34. Właściwości kontrolki TICKEDIT	317
14.35. Właściwości kontrolki NUMBOX	318
14.36. Właściwości kontrolki COMBOEDIT	319
14.37. Właściwości kontrolki	319
15.1. Przyciski na rolecie zawiadomień.	340

1 | Spisy

ROZDZIAŁ

Spis jest listą wartości proponowanych przez AssetCenter do wypełnienia niektórych pól (pól standardowych na ekranie szczegółów, lub wartości cechy), np. jakości, położenia, kraju, marki.

Pozwala to standaryzować wartości w tych polach i ułatwia wprowadzanie danych.

Wartości są wyświetlane na "liście rozwijanej". Aby przypisać wartość do pola, należy po prostu wybrać prawidłową wartość z listy.

AssetCenter zarządza dwoma typami spisów:

- Spisami dowolnymi
- Spisami systemowymi

Spisy dowolne




Administrator AssetCenter może uzyskać dostęp do spisów dowolnych przy użyciu polecenia menu Administracja/ Spisy.

Istnieją dwa typy spisów dowolnych:

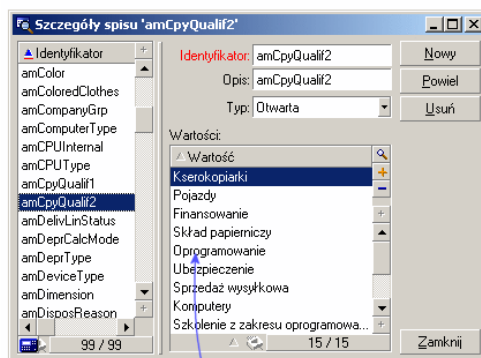
- Spisy, które można tworzyć samodzielnie. Można łączyć je z cechami, ale nie można dołączać ich do pól: wyłącznie oprogramowanie może przypisać spis do pola.
- Spisy, które są połączone z polami bazy danych. Spisy te mogą również być dołączone do cech. Jeśli użytkownik usunie spis tego typu, lub zmieni jej nazwę, AssetCenter ponownie utworzy spis (bez skojarzonych wartości) stosując jej pierwotną nazwę, w chwili, gdy ten spis będzie wymagana do utworzenia rekordu zawierającego pole, które powinno być połączone z tą listą.

Wartości spisów

Lista "wartości" wyświetlana w szczegółach spisu zawiera wartości, które będą proponowane, gdy użytkownik będzie wypełniał pole skojarzone z tym spisem.

Administrator może usuwać, modyfikować lub dodawać wartości za pomocą przycisków ,  oraz  znajdujących się po prawej stronie listy.

Rysunek 1.1. Spisy - okno



Lista wartości proponowanych przy wypełnianiu pola.

Spisy otwarte

Pole **Typ** (nazwa SQL: seType) w szczegółach tych spisów jest ustawione jako **Otwarta**.

Użytkownicy AssetCenter mogą wprowadzać inne wartości, niż proponowane w liście.

Jeśli użytkownik wprowadza nową wartość, jest ona dodawana do listy wartości w spisie (ta lista wartości jest współdzielona przez wszystkich użytkowników). Pojawia się komunikat z prośbą o potwierdzenie utworzenia.

Spisy zamknięte

Pole **Typ** (nazwa SQL: seType) szczegółów dla tych spisów jest ustawione jako **Zamknięta**.

Użytkownicy AssetCenter wprowadzają wyłącznie wartości zawarte w tej liście.



Uwaga:

Po kliknięciu przycisku **Utwórz** lub **Modyfikuj** w oknie zarządzania spisami, wszystkie modyfikacje, które administrator definiuje dla spisów, zostaną przechowywane w bazie danych. Modyfikacje wprowadzone do spisu są uwzględniane na poziomie klienta AssetCenter dopiero po następnym połączeniu z bazą danych, które następuje po danej modyfikacji.

Spisy systemowe

Wartości spisu systemowego zdefiniowane są przez AssetCenter. Nie mogą one być dostosowywane przez administratora, ani użytkownika.

Spisy te nie mogą być edytowane poprzez polecenie menu **Administracja/Spisy**.

Wartości w spisach systemowych

Wartości wyświetlane są różne od wartości przechowywanych w bazie danych.

Wartości wyświetlane są różne od wartości przechowywanych w bazie danych.

Przykład pola **Przypisanie** (nazwa SQL: seAssignment) w szczegółach zasobu:

Tabela 1.1. Wartości w spisach systemowych

Wartość przechowywana w bazie danych	Wartość wyświetlana
0	W użyciu
1	W zapasie
2	Zasób wycofany
3	W trakcie dostawy

Istnieje kilka sposobów, aby uzyskać dostęp do wartości spisu systemowego:

- Przy użyciu pomocy kontekstowej dla pola wypełnianego przez spis systemowy.
- Poprzez program AssetCenter Database Administrator.
- Za pomocą pliku **Database.txt** opisującego strukturę bazy danych.

2 | Zapis historii

ROZDZIAŁ

Można rejestrować wszystkie modyfikacje wprowadzane do dowolnych pól lub łączy w bazie danych. Za każdym razem, gdy tworzysz, modyfikujesz lub usuwasz wartość w polu, dla którego zachowywane jest historia, AssetCenter tworzy pozycję historii w zakładce Historia ekranu.

W tym celu należy określić, że dla danego pola lub łącza zachowywana jest historia. Aby to zrobić:

- 1 Z menu kontekstowego, wybierz **Konfiguruj obiekt**.
- 2 Przejdź do zakładki **Ogólne** na ekranie konfiguracji.
- 3 Ustaw pole **Historia** na **Tak**.
- 4 Kliknij przycisk **OK**, aby potwierdzić.

Uwaga:

Wszystkie modyfikacje dotyczące historii są zapisywane w bazie danych w momencie kliknięcia przycisku **OK** w oknie ustawień indywidualnych bazy danych. Można również określić, czy zachowywana jest historia dla pola lub łącza, za pomocą programu AssetCenter Database Administrator.

Gdy dla pola lub łącza zachowywana jest historia, jest ona dostępna dla wszystkich użytkowników AssetCenter.

W momencie, gdy dla przynajmniej jednego pola lub łącza zachowywana jest historia, w oknie szczegółów rekordu dla tej tabeli pojawia się zakładka **Historia**. Są w niej zachowywane "linie historii", które opisują szczegółowo modyfikacje wprowadzone w polu lub łączu.

Linie historii zawierają kilka informacji:

- **Zmod. w dniu** (nazwa SQL: dtLastModif): Data wprowadzenia modyfikacji.
- **Autor** (nazwa SQL: Author): Osoba, która wykonała modyfikację (login, nazwisko i imię).
- **Pole** (nazwa SQL: Field): Nazwa pola, które zostało zmodyfikowane (krótki opis).
- **Poprzednia wartość** (nazwa SQL: PreviousVal): Poprzednia wartość zmodyfikowanego pola (za wyjątkiem pól typu "komentarz")
- **Nowa wartość** (nazwa SQL: NewVal): Nowa wartość zmodyfikowanego pola (za wyjątkiem pól typu "komentarz"). Domyślnie, pole to nie jest wyświetlane na liście. Aby je wyświetlić, kliknij listę prawym przyciskiem myszy, a następnie wybierz **Konfiguracja listy**.

 **Uwaga:**

Jeśli importujesz bazę danych ze starszej wersji AssetCenter, pole **Nowa wartość** będzie puste dla linii historii.

-
- **Poprzed. komentarze** (nazwa SQL: memPreviousCmt): Poprzednia wartość pól typu "komentarz". Pola typu "komentarz" nie są przetwarzane w taki sam sposób jak inne pola, ponieważ są one w odmienny sposób przechowywane w bazie danych (maks. rozmiar: 32767 znaków).

Zależnie od okoliczności, AssetCenter funkcjonuje w odmienny sposób:

Utworzenie rekordu

Utworzenia rekordów są rejestrowane, jeśli użytkownik polecił programowi AssetCenter, aby śledzić historię wszystkich modyfikacji wprowadzonych do pola identyfikacyjnego odpowiadającego kluczowi głównemu tabeli.

AssetCenter rejestruje:

- **Zmod. w dniu:** datę utworzenia
- **Autor:** Autora utworzenia
- **Pole:** "Utworzenie"
- **Poprzednia wartość:** "Utworzenie"

Modyfikacja pola w tabeli lub łącza 1 (na przykład: Użytkownik zasobu)

AssetCenter rejestruje:

- **Zmod. w dniu:** Datę wprowadzenia modyfikacji
- **Autor:** Autora modyfikacji
- **Pole:** Nazwę zmodyfikowanego pola
- **Poprzednia wartość:** Poprzednią wartość zmodyfikowanego pola
- **Nowa wartość:** Nową wartość zmodyfikowanego pola

Dodawanie łącza n do innej tabeli (na przykład: Zasoby ujęte w umowach)

AssetCenter rejestruje:

- **Zmod. w dniu:** Datę dodania
- **Autor:** Autora dodania
- **Pole:** Nazwę łącza
- **Poprzednia wartość:** Odwołania związanego rekordu, który został dodany
- **Nowa wartość:** Nową wartość zmodyfikowanego łącza

Usunięcie łącza n do innej tabeli

AssetCenter rejestruje:

- **Zmod. w dniu:** Datę usunięcia
- **Autor:** Autora usunięcia

- **Pole:** Odwołania związanego rekordu, który został usunięty
- **Poprzednia wartość:** Odwołania związanego rekordu, który został usunięty
- **Nowa wartość:** Nową wartość zmodyfikowanego łącza (pusta)

Modyfikacja łącza n do innej tabeli

AssetCenter nie rejestruje modyfikacji wprowadzanych do łącza. Aby zachować ślad, należy usunąć przestarzałe łącze, a następnie dodać nowe łącze.

Zachowywanie historii cech skojarzonych z rekordami

W AssetCenter można zachowywać historię cech, dokładnie tak, jak dla każdego innego pola w bazie danych. Historia cechy dotyczy:

- Dodawania cech
- Usuwania cech
- Zmian wartości cechy

W historii zachowywanych jest kilka typów działań:

Dodawanie cechy

Dodania nowych cech są rejestrowane, jeśli pole **Zachowuj historię** (nazwa SQL: seKeepHistory) dla cechy jest ustawione jako **Tak**, oraz jeśli pole wyboru **Zachowuj historię tworząc zapis główny**. (nazwa SQL: bCreationHistory) jest zaznaczone.

AssetCenter rejestruje:

- **Zmod. w dniu** (nazwa SQL: dtLastModif): datę dodania cechy.
- **Autor** (nazwa SQL: Author): osobę, która dodała cechę.
- **Poprzednia wartość:** "Utworzenie"
- **Pole** (nazwa SQL: fieldName): nazwę SQL cechy.

Usuwanie cechy

Usuwanie cech są rejestrowane, jeśli pole **Zachowuj historię** dla cechy jest ustawione jako **Tak**.

AssetCenter rejestruje:

- **Zmod. w dniu:** datę usunięcia cechy.
- **Autor:** osobę, która usunęła cechę.
- **Pole:** nazwę SQL cechy.
- **Poprzednia wartość:** "Usunięcie cechy ('wartość cechy')".
- **Nowa wartość:** Nową wartość cechy (pusta)

Modyfikowanie wartości cechy

Modyfikacje cech są rejestrowane, jeśli pole **Zachowuj historię** (nazwa SQL: seKeepHistory) dla cechy jest ustawione na **Tak**.

AssetCenter rejestruje:

- **Zmod. w dniu:** datę zmodyfikowania cechy.
- **Autor:** osobę, która zmodyfikowała cechę.
- **Pole:** nazwę SQL cechy.
- **Poprzednia wartość:** poprzednią wartość cechy.
- **Nowa wartość:** Nowa wartość zmodyfikowanej cechy.

 **Ostrzeżenie:**

Jeśli usuwasz rekord, wszystkie linie historii są również usuwane, albo bezpośrednio, albo przez AssetCenter Server.

Tworzenie, usuwanie lub modyfikowanie linii historii

Nie można zachowywać historii tworzenia linii historii.

Tworzenie linii historii

Aby zainicjować tworzenie linii historii dla cechy, należy ustawić pole **Zachowuj historię** na **Tak**. Aby to zrobić, wybierz zakładkę **Parametry** w szczegółach cechy i kliknij przycisk naprzeciw linii parametrów.

AssetCenter wyświetli ekran ze szczegółami parametrów cechy. Pole **Zachowuj historię** znajduje się w zakładce **Ograniczenia** na tym ekranie.

Gdy pole to jest ustawione na **Tak**, AssetCenter automatycznie tworzy linie historii dla tej cechy. Linie historii mogą być przeglądane w zakładce **Historia** tabeli skojarzonej z tą cechą.

Ostrzeżenie:

Jeśli usuwasz rekord, wszystkie linie historii są również usuwane, albo w czasie usunięcia, albo poprzez AssetCenter Server. Nie można zachowywać historii tworzenia historii.

3 Zapytania w AQL

ROZDZIAŁ

Niniejszy rozdział wyjaśnia, w jaki sposób pisać zapytania w AQL:

Przedstawienie

Podrozdział ten przedstawia język AQL i wymienia miejsca, w których może zajść konieczność użycia zapytań.

AQL

AQL ("Advanced Query Language" - "Zaawansowany Język Zapytań") jest językiem zapytań wykorzystywanym przez AssetCenter do uzyskiwania dostępu do bazy danych AssetCenter. Jest on porównywalny z SQL. Zapytania pisane w AQL są automatycznie tłumaczone na stosowny język SQL używanego motoru bazy danych.

 Uwaga:

Dobra znajomość SQL oraz baz danych jest bardzo przydatna przed bezpośrednim zastosowaniem AQL.

Użyteczność języka AQL

AQL lepiej nadaje się do tworzenia zapytań dotyczących bazy danych AssetCenter niż SQL, z następujących powodów:

Niezależność od bazy danych

Różne motory bazy danych obsługiwane przez AssetCenter używają różnych wersji SQL, które nie są ze sobą zgodne. AQL jest niezależny od wykorzystywanego motoru bazy danych.

W rezultacie, jeśli użytkownik przejdzie do innego motoru bazy danych, jego zapytania napisane w AQL będą działać tak samo.

Na przykład, AQL wykorzystuje ten sam zestaw funkcji bez względu na użyty motor bazy danych.

A zatem, funkcja **Substring** w AQL jest równoważna funkcji **Substr** użytej w systemie SQL Oracle for WorkGroups oraz funkcji **Substring** użytej w systemie SQL Microsoft SQL Server.

Generowanie zoptymalizowanego kodu SQL

AQL generuje kod SQL zoptymalizowany zgodnie z używanym motorem bazy danych.

Jest to szczególnie widoczne w przypadku indeksów. Na przykład, podczas wyszukiwania pełnej nazwy modeli z wymuszeniem stosowania indeksów **Identyfikator modelu (Model_Id)** i **Pełna nazwa (FullName)** należałoby napisać następujące zapytanie AQL:

```
SELECT FIRST_ROWS IModelId, FullName FROM amModel
```

Wygenerowany kod SQL będzie się różnił w zależności od docelowego motoru bazy danych i będzie dla niego zoptymalizowany. A zatem, w kod SQL będzie miał postać:

```
SELECT /*+ FIRST_ROWS INDEX_ASC(M1 Model_Id) */ M1.IModelId, M1.FullName FROM amModel M1
```

Kod Microsoft SQL Server lub Sybase SQL Server:

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

Kod IBM DB2 będzie miał postać:

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

Uproszczony dostęp do bazy danych AssetCenter

AQL upraszcza zarządzanie łączami i połączeniami. Znacznie ułatwia to dostęp do bazy danych podczas pisania zapytań w porównaniu do bezpośredniego korzystania z SQL.

Na dodatek, AQL upraszcza dostęp do cech, pozwalając je używać jako pola bezpośrednie w związanych z nimi tabelach.

AQL ułatwia również wykorzystanie pól wyliczanych.

AQL wobec SQL - porównanie

AQL nie obsługuje poleceń DDL ("Data Definition Language").

AQL zawiera rozszerzenia, które upraszczają obsługę połączeń, cech oraz pól wyliczanych.

Ostrzeżenie:

Nigdy nie należy pisać bezpośrednio do bazy danych AssetCenter przy użyciu instrukcji SQL.

Zapytania w AssetCenter

Zapytania pozwalają łączyć warunki dotyczące informacji w tabeli lub w tabelach związanych.

Można użyć zapytań:

- Aby utworzyć filtry dla list rekordów. W tym przypadku zapytania są, ogólnie rzecz biorąc, proste i oparte na warunku "Where".
- Aby zdefiniować widoki.
- Aby zdefiniować warunki eksportu w module eksportu.

- Aby utworzyć raporty Crystal Reports.
- Aby utworzyć kreatory
- W przypadku używania API AssetCenter.
- Jeżeli AssetCenter jest wykorzystywany jako serwer DDE.

AQL ("Advanced Query Language" - "Zaawansowany Język Zapytań") jest wewnętrznym językiem zapytań AssetCenter przeznaczonym do prostego dostępu do bazy danych AssetCenter.

AssetCenter zawiera edytor umożliwiający określanie zapytań w AQL:

- Albo pośrednio, przy użyciu interfejsu graficznego,
- Albo poprzez napisanie zapytania bezpośrednio w AQL.

 **Ostrzeżenie:**

Aby zilustrować możliwości AQL, przykłady podane w dalszej części wykorzystują całą składnię AQL. W szczególności, wyraźnie pokazane są warunki SELECT, WHERE oraz FROM. Niektóre funkcje, takie jak filtry zapytań (gdzie użytkownik definiuje w zapytaniu AQL wyłącznie warunek WHERE) lub kompilator wyrażeń znacznie ułatwiają użytkownikowi tworzenie zapytań (niektóre warunki są niewidoczne). Przykłady te nie mogą być wykorzystywane dla tych funkcji.

Zalecenia dotyczące pisania zapytań AQL

Zalecane jest przeczytanie tego podrozdziału przez rozpoczęciem pisania zapytań w AQL.

Podrozdział ten opisuje:

- Notację specyficzną dla AQL.
- Specyfikę AQL oraz bazy danych AssetCenter, które mają wpływ na optymalne projektowanie zapytań.

Podrozdział ten uzupełniają podrozdziały [Składnia AQL](#) oraz [Odwołanie funkcji AQL](#).

! Ostrzeżenie:

Zapytania napisane w AQL używają nazw SQL ("SQLName") pól, łączy i tabeli w bazie danych AssetCenter. Zapoznaj się z plikiem Database.txt, który opisuje strukturę bazy danych i zawiera wyczerpującą listę tych nazw.

Plik ten znajduje się w następującym folderze: [Katalog instalacyjny AssetCenter]/doc/infos

Przedstawienie połączeń AQL

Definicja

Połączenie konsoliduje wiele tabel danych w pojedynczym zapytaniu.

Połączenia AQL

Opis bazy danych AssetCenter, oprócz definiowania tabel i pól definiuje łącza pomiędzy tabelami. Pozwala to zautomatyzować tworzenie złączeń na poziomie AQL.

Łącza AQL są wyrażone jako:

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

Upraszczając obsługę złączeń, AQL upraszcza tworzenie większości zapytań używanych w bazie danych AssetCenter.

Przykład

Następujące zapytanie, napisane w AQL, zwraca dla każdego modelu:

- jego **Identyfikator (IModelId)**
- jego **Pełną nazwę (FullName)**
- **Nazwę (Name)** tabeli powiązanej z markami (**amBrand**)

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

Oto to samo zapytanie, napisane w SQL Oracle lub Microsoft SQL Server:

```
SELECT M1.IModelId, M1.FullName, B2.Name FROM amModel M1, amBrand B2 WHERE M1.IBrandId=B2.IBrandId
```

Te dwa złączenia pomiędzy tabelą **Modele (amModel)** i tabelą **Marki (amBrand)** są automatycznie obsługiwane w AQL. Korzystając z graficznego edytora zapytań programu AssetCenter, aby utworzyć stosowny kod AQL, wystarczy kliknąć pola wybranej tabeli lub tabeli związanej na liście o strukturze drzewa.

 **Uwaga:**

Liczba połączeń zewnętrznych jest ograniczona do 1 dla wszystkich systemów poza Oracle i DB2.

W Microsoft SQL Server 7 i MSSQL 2000, aby zapobiec ewentualnym problemom przy wykonywaniu zapytań, można zmodyfikować plik amdb.ini, w szczegółach połączenia, wpisując następujące polecenie:

```
useSQL92Join=1
```

Przyczyna i użyteczność rekordów klucza głównego 0

Rekordy klucza głównego "0"

Model danych w bazie danych AssetCenter ma pewne specyficzności:

- Klucze podstawowy i zewnętrzny każdej tabeli są numeryczne (liczba całkowita 32-bitowa).
- Klucz zewnętrzny, który nie wskazuje na rekord, jest ustawiony jako "0" (a nie "NULL").
- Każda tabela ma pusty rekord, którego klucz główny jest ustawiony jako "0".

Użyteczność

Z tymi rekordami klucza głównego "0", wyniki zapytania stosującego połączenie, nie będące połączeniem zewnętrznym, pomiędzy dwiema danymi tabelami A i B, mogą zawierać rekordy z tabeli A, które nie są połączone z żadnym rzeczywistym rekordem w tabeli B (łącze nie wypełnione). Są one rekordami w tabeli A, które są połączone z rekordem klucza głównego "0"

Przykład:

Następujące zapytanie, napisane w AQL, zwraca dla każdego kodu wewnętrznego elementu repozytorium nazwę jego użytkownika i zarządcy:

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

Element repozytorium, który nie jest przypisany do użytkownika oraz/lub zarządcy, jest wyświetlany w wynikach zapytania. Na poziomie bazy danych, taki element repozytorium jest związany z rekordem klucza głównego "0" tabeli wydziałów i pracowników.

Przyczyna tych specyficzności

Sekcja ta wyjaśnia, dlaczego używane są rekordy klucza głównego "0", podczas gdy zapytanie wykorzystujące zewnętrzne połączenie SQL może wybrać rekordy w tabeli A, które nie są połączone z rekordem w tabeli B. Rekordy klucza głównego "0" pozwalają zrekompensować fakt, że niektóre modele relacyjnych baz danych (RDBM) nie obsługują wielokrotnych połączeń zewnętrznych: Używając rekordów klucza podstawowego "0", zapytania SQL wygenerowane z zapytań AQL nie wykorzystują połączeń zewnętrznych.

Przykład:

Poniższe zapytanie AQL wyszukuje każdy element repozytorium, jego kod wewnętrzny oraz nazwę lokalizacji jego użytkownika. Wyniki będą obejmować elementy repozytorium, które nie mają użytkownika, oraz elementy repozytorium, których użytkownicy nie mają lokalizacji.

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

Jeśli wygenerowany kod SQL używa zewnętrznych połączeń systemu zarządzania bazą danych (DBMS), kod SQL wygenerowany dla systemu Sybase SQL Server będzie następujący:

```
SELECT a.AssetId, l.name FROM amPortfolio a, amEmplDept e, amLocation l WHERE a.UserId = e.EmplDeptId AND e.LocId = l.LocId
```

Kod ten nie jest obsługiwany przez system Sybase SQL Server ponieważ używa on kilku kolejnych zewnętrznych połączeń.

Jednak, ponieważ jest to rekord klucza głównego "0" w tabeli wydziałów i pracowników oraz lokalizacji, nie ma potrzeby, aby używać zewnętrznych połączeń. A zatem, AssetCenter generuje kod SQL bez zewnętrznych połączeń:

```
SELECT l.name FROM amPortfolio a, amEmpDept e, amLocation l WHERE a.l.UserId = e.lEmpDeptId AND e.lLocId = l.lLocId
```

Zapytanie to daje spodziewane wyniki, ponieważ łączy **Użytkownik (User)** i **Lokalizacja (Location)** nadal wskazują na rekord w tabeli wydziałów i pracowników lub lokalizacji (wskazują one na rekord klucza głównego "0", jeśli łącze nie jest wypełnione).

Skutki

- Ważne jest, aby uwzględnić te rekordy w pisanych zapytaniach, szczególnie podczas stosowania funkcji zbiorczych.

Przykład:

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

Jeśli wykonasz powyższe zapytanie, które zlicza liczbę zasobów w tabeli zasobów, rekord klucza głównego "0" jest uwzględniany w wynikach. Dlatego też musisz zmniejszyć wyniki o 1, aby uzyskać rzeczywistą liczbę zasobów w bazie danych.

- Rzadko zachodzi konieczność generowania zewnętrznych połączeń na poziomie systemu zarządzania bazą danych (DBMS).

Uwaga:

Jeśli naprawdę chcesz zarządzać zewnętrznymi połączeniami na poziomie systemu zarządzania bazą danych (DBMS), stosuj operatory AQL "=" oraz "*=".

Użycie wartości NULL

AssetCenter stosuje wartość NULL systemu zarządzania bazą danych (DBMS) wyłącznie w dwóch przypadkach:

- Dla pustego pola typu "tekst".
- Dla nie wypełnionego pola typu "data" lub "data+czas".

AQL pozwala stosować kilka równoważnych składni, tak jak pokazano niżej. Konwertowane są one na równoważny prawidłowy kod SQL motoru bazy danych.

Dla pustych pól typu "Tekst", można użyć dowolnej z następujących składni, ponieważ wartość NULL będzie przechowywana w bazie danych:

WHERE <pole tekstowe> = NULL

WHERE <pole tekstowe> IS NULL

WHERE <pole tekstowe> = "

Dla nie wypełnionych pól typu "data" lub "data+czas", można użyć dowolnej z następujących składni, ponieważ wartość NULL będzie przechowywana w bazie danych:

WHERE <pole data lub data+czas> = NULL

WHERE <pole data lub data+czas> IS NULL

WHERE <pole data lub data+czas> = []

 **Uwaga:**

Kiedy pole typu "Numeryczne" nie jest wypełnione, jego wartość wynosi "0". Podobnie, brak łącza jest opisany jako "Link = 0" lub "foreign key = 0". Przykład: "Location=0" or "ILocaId=0".

Wyrażenie Self

"Self" jest wyrażeniem, które jest równoważne ciągowi opisu dotyczącego tabeli, do której jest stosowane.

Użycie wyrażenia "Self" pozwala uprościć zapytania i uwzględnić wszystkie ustawienia indywidualne bazy danych AssetCenter.

Przykład:

Jeśli ciąg opisu tabeli wydziałów i pracowników jest następujący:

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

Zapytanie AQL:

```
SELECT self FROM amEmplDept
```

Jest równoważne:

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

Funkcja CurrentUser

Funkcja "CurrentUser" pozwala pisać zapytania, które są zależne od osoby połączonej z bazą danych.

Funkcja "CurrentUser" może być użyta jako wyrażenie, na przykład w zapytaniu, lub jako łącze. Musisz wprowadzić to wyrażenie ręcznie, ponieważ nie jest ono oferowane przez edytor zapytań.

Użyta jako "wyrażenie"

Przykład: szukamy wszystkich elementów repozytorium używanych przez pracownika połączonego z bazą danych.

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

Użyta jako "łącze"

"CurrentUser" może być traktowana jako łącze znajdujące się w każdej tabeli, która wskazuje na rekord odpowiadający bieżącemu użytkownikowi w tabeli wydziałów i pracowników.

- W postaci "CurrentUser", funkcja ta wskazuje na rekord odpowiadający bieżącemu użytkownikowi.
- W postaci "CurrentUser.Field", funkcja ta zwraca wartość pola dla bieżącego użytkownika.

Przykład: Kiedy połączony użytkownik inicjuje działanie, istnieje możliwość kontekstowego inicjowania innego działania typu przesyłanie komunikatów, które automatycznie wysyła komunikat ostrzeżenia do

połączonego użytkownika. Wystarczy wypełnić szczegóły tego działania, jak następuje:

Spisy systemowe

Jeśli zapytanie AQL używa spisu systemowego, musisz użyć wartości, które są przechowywane w bazie danych, a nie tych, które są wyświetlone na ekranie.

Przykład:

Następujące zapytanie wybiera te umowy, których pole **Typ** (nazwa SQL: seType) jest ustawione jako **Leasing ramowy**:

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

Pole **Typ** (nazwa SQL: seType) jest spisem systemowym. Wartości przechowywane w bazie danych są następujące:

- 0 dla umowy typu **Inne**
- 1 dla umowy typu **Leasing ramowy**
- 2 dla umowy typu **Harmonogram leasingu**
- 3 dla umowy typu **Ubezpieczenie**
- 4 dla umowy typu **Serwis**

 Uwaga:

Wartości spisu systemowego można odnaleźć poprzez program AssetCenter Database Administrator, lub plik **Database.txt**, który opisuje strukturę bazy danych.

Plik ten znajduje się w następującym folderze: [Katalog instalacyjny AssetCenter]/ doc/infos

Tabele hierarchiczne

Wszystkie tabele hierarchiczne zawierają:

- Pole "FullName".
- Pole "sLvl".

Pola "FullName"

Dla każdego rekordu w tabeli hierarchicznej, pole "FullName" przechowuje wartość pola rekordu, poprzedzoną strukturą drzewa składającą się z wartości pól rekordów nadrzędnych aż do poziomu głównego.

Wartości są oddzielone przez znak "/" bez spacji. Znak ten pojawia się również na początku oraz na końcu struktury drzewa.

Przykłady:

- Dla tabeli zasobów, pole "FullName" przechowuje Kod Wewnętrzny zasobu poprzedzony Kodem Wewnętrznym jego zasobu nadrzędnego, który z kolei jest poprzedzony Kodem Wewnętrznym jego zasoby nadrzędnego, i tak dalej.

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

- W tabeli lokalizacji, pole "FullName" przechowuje nazwę lokalizacji poprzedzoną nazwami lokalizacji nadrzędnych.

```
FullName = '/Paris/Site Tolbiac/Tour A/Floor 5/'
```

Pola "sLvl"

Dla każdego rekordu w tabeli hierarchicznej, pole "sLvl" wskazuje jego poziom w strukturze drzewa.

Poziom główny jest poziomem 0.

Nazwa	Kod...
006 - Zapas surowca	DEMO-L059
Biurowiec Polaris	DEMO-L01
23 piętro	DEMO-L02
007 - Sala konferencyjna	DEMO-L03
008 - Biuro	DEMO-L04
009 - Biuro	DEMO-L05
010 - Biuro	DEMO-L06
011 - Biuro	DEMO-L07
012 - Biuro	DEMO-L08
30 piętro	DEMO-L09
Firmy zewnętrzne	DEMO-LS0

Następujące zapytanie wybiera rekord "Centrala sprzedaży" oraz jego składniki podrzędne:

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/Sales Head Office/Sales/%') AND (sLvl >= 1)
```

Następujące zapytanie wybiera rekord "Centrala sprzedaży", ale nie wybiera jego składników podrzędnych:

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/Sales Head Office/Sales/%') AND (sLvl = 1)
```

Następujące zapytanie wybiera składniki podrzędne rekordu "Centrala sprzedaży", ale nie wybiera samego rekordu "Centrala sprzedaży":

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/Sales Head Office/Sales/%') AND (sLvl > 1)
```

Uproszczona notacja AQL

Sekcja ta wylicza notację, którą można stosować, aby uprościć pisanie zapytań AQL:

Klucze zewnętrzne

W warunkach innych niż SELECT oraz ORDER BY, nazwa SQL łączy, po którym nie następuje kropka, jest równoważna nazwie SQL skojarzonego klucza zewnętrznego.

Przykład: warunek:

```
WHERE location = 0
```

Jest równoważny:

```
WHERE ILocId = 0
```

Gdzie "location" jest nazwą SQL łącza "Lokalizacja" z tabeli wydziałów i pracowników do tabeli lokalizacji, a "ILocId" jest nazwą SQL skojarzonego klucza zewnętrznego w tabeli zasobów.

Ciąg opisu

W warunkach SELECT i ORDER BY, nazwa SQL łącza, po którym nie następuje kropka, jest równoważna złączeniu <nazwa SQL łącza>.self, które z kolei jest równoważne <nazwa SQL łącza>.<Ciąg opisu>.

Przykład:

Jeśli ciąg opisu tabeli wydziałów i pracowników jest następujący:

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

Wtedy zapytanie AQL:

```
SELECT user FROM amPortfolio
```

Jest równoważne zapytaniu:

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

Które samo jest równoważne:

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

Cechy

AQL daje użytkownikowi bezpośredni dostęp do cech tabeli, tak jak gdyby były one bezpośrednimi polami w tabeli. Aby wyszukać wartości cechy dla danej tabeli, wystarczy wpisać nazwę SQL cechy (przedrostek **fv_**).

Przykład: następujące zapytanie wyszukuje wartości cech o nazwie SQL **fv_WorkUnit** dla tabeli Pracowników (**amEmplDept**):

```
SELECT fv_WorkUnit FROM amEmplDept
```

Pola wyliczane

AQL ułatwia użycie pól wyliczanych skojarzonych z tabelą.

Wystarczy wpisać nazwę SQL pola wyliczanego (przedrostek `cf_`).

Sortowania i indeksy

AQL daje możliwość użycia dwóch strategii dla zapytań, które wykorzystują sortowania (warunek ORDER BY):

- Tryb, w którym AssetCenter wymusza indeksy wskazane w zapytaniu, gdy są użyte, i wyświetla wyniki, gdy są pobrane.
- Tryb, w którym AssetCenter nie wymusza indeksów wskazanych w zapytaniu. W tym przypadku, system zarządzania bazą danych (DBMS) określa, w jaki sposób dane są sortowane.

Uwaga:

Wybór spośród tych dwóch różnych metod nie jest dostępny w systemie SQL Anywhere. Motor bazy danych automatycznie wybiera najbardziej odpowiednią metodę.

Przykład

W przypadku następującego zapytania:

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

- Dostęp bez **wymuszania indeksów**: motor bazy danych skanuje tabelę w całości bez stosowania indeksu "Marka" wskazanego w zapytaniu. Wyszukuje on wszystkie pozycje danych spełniające zapytanie, sortuje je według "Marka" i wysyła je do użytkownika. Wyniki będą wyświetlone dopiero po pewnym okresie czasu.
- W drugim przypadku: Motor bazy danych stosuje indeks "Marka" i wyświetla wyniki wtedy, gdy zostaną znalezione. Pierwsze pozycje są zatem pokazane nagle, ale całkowity czas przetwarzania jest dłuższy.

W jaki sposób wymusić indeksy

Sposób wymuszenia wykorzystania indeksów zależy od sposobu tworzenia zapytania.

Poprzez menu Konfiguruj listę

Można skonfigurować typ dostępu do danych dla każdej listy w AssetCenter. Dotyczy to zarówno list głównych, jak i listy zawartej w zakładkach. Aby to zrobić:

- 1 Przejdź do listy, którą chcesz skonfigurować.
- 2 Kliknij prawym przyciskiem myszy.
- 3 Wybierz pozycję **Konfiguracja listy** w menu kontekstowym.
- 4 W zakładce **Kolumny i sortowanie**, zaznacz pole **Wymuś indeksy**, aby użyć indeksów wskazanych w zapytaniu i wyświetlić wyniki wtedy, gdy są generowane; Usuń zaznaczenie tego pola, aby wybrać drugi typ dostępu.

Przy użyciu AQL

Jeśli piszesz zapytanie bezpośrednio w AQL, możesz wymusić użycie indeksów poprzez warunek "FIRST_ROWS".

Przykład:

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

 Uwaga:

Jeżeli sortowanie dotyczy spisów systemowych, na przykład pola **seDataType** w tabeli cech, może się zdarzyć, że sortowanie nie zostanie wykonane prawidłowo.

Porządek sortowania

Porządek sortowania zależy od:

- Motoru bazy danych
- Użycia indeksów

W systemie Oracle

Z wymuszonymi indeksami

- Rekordy NULL nie są wyświetlane.

- Sortowanie jest wykonywane według wartości kodów ASCII, a zatem rozróżniane są znaki pisane dużymi i małymi literami (sortowanie binarne).

Bez wymuszonych indeksów

- Rekordy NULL są wyświetlane.
- System Oracle nie rozróżnia wielkości liter.

Przykład

Sortowanie

Lista początkowa	A B C D a b NULL NULL
Lista z wymuszonymi indeksami	A B C D a b
Lista bez wymuszonych indeksów	NULL NULL A a B b C D

W systemie Microsoft SQL Server lub Sybase SQL Server

Porządek sortowania zależy od parametru ustawionego przy tworzeniu bazy danych. Motory te można skonfigurować, aby rozróżniały wielkość liter, lub aby brały pod uwagę znaki akcentowane, itp.

W systemie Sybase SQL Anywhere

W systemie Sybase SQL Anywhere, nie można wymusić indeksów za pośrednictwem zapytania AQL.

Motor bazy danych określa optymalną metodę wykorzystywaną do uzyskiwania dostępu do danych oraz do ich sortowania.

Ostrzeżenia

Przy złożonych zapytaniach często trudno jest określić, czy bardziej "korzystnie" jest wymuszać indeksy, czy nie. W praktyce, zalecane jest wykonanie tekstów przed podjęciem ostatecznej decyzji.

W szczególności zalecane jest testowanie z, oraz bez wymuszonych indeksów w przypadku listy, która jest filtrowana, zarówno bezpośrednio (przez filtr prosty zapytania), jak i pośrednio (przez ograniczenia dostępu).

Edytor zapytań

AssetCenter zawiera edytor zapytań. Narzędzie to pozwala projektować zapytania i wyświetlać podgląd ich wyników. Jest on przeznaczony przede wszystkim dla administratorów bazy danych lub zaawansowanych użytkowników.

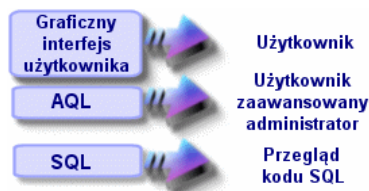
Omówienie

Edytor zapytań umożliwia projektowanie zapytań:

- Albo przy użyciu interfejsu graficznego (wspomagającemu projektowanie zapytań),
- Albo poprzez napisanie zapytania bezpośrednio w AQL.

Bez względu na to, czy użytkownik stosuje metodę graficzną, czy woli pisać bezpośrednio w AQL (oba podejścia są często łączone), widzi on w czasie rzeczywistym kopię swojego zapytania w SQL. Użytkownik nie może jednak pisać swoich zapytań bezpośrednio w SQL.

Rysunek 3.1. Edytor zapytań - tryby tworzenia



Za pomocą edytora zapytań, zaawansowany użytkownik lub administrator może tworzyć, modyfikować lub usuwać zapytania AQL. Zapytania te mogą być użyte w odpowiednim kontekście przez ich autora, lub innych użytkowników.

Uzyskiwanie dostępu do edytora zapytań

Dostęp do edytora zapytań można uzyskać na różne sposoby:

- Przez polecenie menu **Narzędzia/ Zapytania**. Za pomocą tego menu, można tworzyć zapytania na własny użytek, oraz takie, które mogą

być również używane bez ograniczeń przez innych użytkowników. Zapytania mogą być wykonywane:

- Albo bezpośrednio za pośrednictwem okna wyświetlanego przez polecenie menu **Narzędzia/ Zapytania**.
- Albo poprzez użycie "filtru zapytania" podczas wyświetlania głównej tabeli zapytania.
- Za pośrednictwem licznych funkcji AssetCenter, które wywołują zapytania: Ograniczenia dostępu, filtry zapytań, konfiguracje listy, itp.
- Poprzez programy zewnętrzne: AssetCenter Export, itp.

Pokazana wersja edytora zapytań jest w pewnym stopniu uproszczona odpowiednio do kontekstu.

Przykład: Załóżmy, że użytkownik ma następujące zapytanie:

```
SELECT [FIRST_ROWS] <champ>[, <champ>...] FROM <table> [WHERE <clause>] [ORDER BY <clause>]
```

W uproszczonych wersjach edytora zapytań (filtry proste, filtry zapytań, itp.), należy tylko zdefiniować warunek WHERE zapytania. Pozostałe składniki zapytania (tabela początkowa, pola, itp.) są domniemane. Na przykład w przypadku filtra zapytania, tabela jest tą tabelą, do której jest zastosowany filtr, pola i warunki sortowania są kolumnami i warunkami sortowania zdefiniowanymi za pośrednictwem polecenia menu kontekstowego **Konfiguracja listy**. To samo odnosi się do edytora zapytań, do którego uzyskano dostęp poprzez polecenie menu **Narzędzia/ Zapytania**.

A zatem, następujące zapytanie podane w całości:

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

jest zapisane jak następuje, kiedy jest użyte w filtrze zapytania (wyłącznie jeśli warunek WHERE jest jawnie podany) użytym na tabeli modeli:

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

Z drugiej strony, polecenie **Konfiguruj listę** pozwala uzyskać dostęp do bardziej obszernej wersji edytora zapytań:

- Zakładka **Kolumny i sortowanie** definiuje pola, które mają być wyświetlane w kolumnach, oraz warunki sortowania (te warunki sortowania odpowiadają warunkowi ORDER BY).
- Pole **Wymuś indeksy** zastępuje warunek FIRST_ROWS w kodzie SQL.
- Zakładka **Filtr (warunek WHERE)** definiuje warunek "WHERE".

- Tabela jest domniemana.

Tworzenie zapytania przy użyciu edytora zapytań

Aby utworzyć zapytanie za pomocą edytora zapytań, wybierz polecenie menu **Narzędzia/ Zapytania**. Okno, które jest wyświetlone, ma dwie zakładki, **Filtr (warunek WHERE)** oraz **Podgląd**:

- Zakładka **Filtr (warunek WHERE)** jest graficznym interfejsem, który określa warunki zapytania. Definiuje ona elementy warunku WHERE.
- Zakładka **Podgląd** wyświetla transkrypcję zapytania w kodzie SQL i pozwala je testować.

Krok 1: Wypełnij pola w górnej części szczegółów zapytania.

Musisz określić tabelę początkową swojego zapytania.

Jeśli chcesz, aby zapytanie, które tworzysz, mogło być dostępne dla innych użytkowników, usuń zaznaczenie opcji **Nie współdzielone** (nazwa SQL: bPrivate).

 **Uwaga:**

Administrator ma dostęp do wszystkich zapytań przechowywanych w bazie danych, nawet do tych zapytań, które są **Nie współdzielone**.

Po wypełnieniu podstawowych informacji w zapytaniu, kliknij przycisk **Utwórz**, aby móc uzyskać dostęp do zakładek szczegółów zapytania.


Krok 2: Zdefiniuj warunki filtra w zakładce Filtr (warunek WHERE).

Edytor zapytań AssetCenter pozwala definiować warunki, które łączą pola, wyrażenia wyliczane, stałe i operatory.

Możesz zdefiniować jeden, lub większą liczbę warunków filtra.

Aby zdefiniować warunek filtra:


- 1 Z tabeli początkowej, wybierz pole, stałą lub wyrażenie (**Pole 1**), które będziesz porównywać z polem, stałą lub wyrażeniem (**Pole 2**).

- 2 Potwierdź warunek filtra poprzez przeniesienie go do dolnej części ekranu przy użyciu przycisku .
- 3 Potwierdź zapytanie poprzez kliknięcie przycisku **Modyfikuj** w szczegółach zapytania.

Aby zdefiniować kilka warunków filtra połączonych operatorami logicznymi AND oraz OR:

- 1 Utwórz pierwszy warunek filtra, jak wskazano powyżej.
- 2 Zdefiniuj pozostałe warunki filtra i potwierdź je za pomocą przycisków **I** lub **LUB**.
- 3 Potwierdź zapytanie poprzez kliknięcie przycisku **Modyfikuj** w szczegółach zapytania.

 **Uwaga:**


W celu wprowadzenia modyfikacji do wybranych warunków, kliknij przycisk , aby usunąć zawartość okna, lub w przeciwnym razie zmodyfikuj kod AQL bezpośrednio.

 **Uwaga:**

Zamiast w narzędziu graficznym, możesz wprowadzić swoje zapytanie bezpośrednio w AQL w dolnym obszarze zakładki **Filtr (warunek WHERE)**.

Krok 3: Wyświetlanie podglądu wykonania zapytania

Aby przetestować zapytanie i zobaczyć jego transkrypcję w języku SQL:

- 1 Przejdź do zakładki **Podgląd** w szczegółach zapytania.
- 2 Kliknij ikonę : AssetCenter wyświetli podgląd wyników zapytania w postaci listy rekordów. Liczba rekordów spełniających zapytanie jest pokazana w prawej dolnej części okna.

 **Uwaga:**


Kod SQL zawarty w zakładce **Podgląd** nie może być bezpośrednio modyfikowany.

Pola używane w zapytaniach

Podczas definiowania warunków filtru w zapytaniach, można wywoływać:

- Pole w tabeli, której dotyczy zapytanie.
- Pole związane.
- Cechy skojarzone z tabelą.

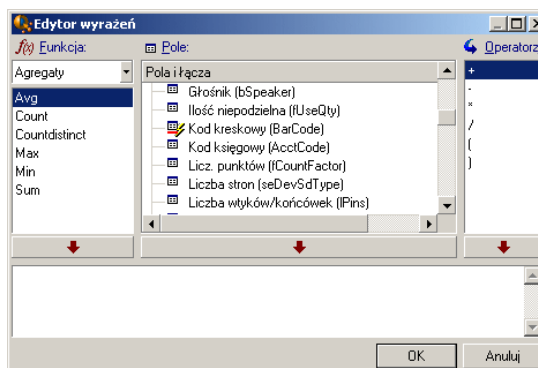
Pisanie wyrażeń

Wyrażenia  pozwalają wykonywać operacje zliczania w zapytaniu. Na przykład, można użyć funkcji "Count" w celu wyliczenia liczby rekordów będących wynikiem zapytania.


Aby napisać wyrażenie, możesz:

- Albo wprowadzić je bezpośrednio w stosownym polu,
- Albo użyć kompilatora wyrażeń AssetCenter.

Aby użyć kompilatora wyrażeń, kliknij przycisk  sąsiadujący z obszarem edycji zakładki **Filtr (warunek WHERE)** szczegółów zapytania.



Kompilator wyrażeń składa się z trzech kolumn:

- Kolumny "Funkcja" wyliczającej istniejące funkcje AQL. Kliknięcie  stosuje filtr do listy funkcji AQL zgodnie z ich typem: "Ogółem", "Ciąg", "Dane", "Numeryczne", "Test".
- Kolumna "Pole" wylicza różne pola, które mogą być użyte w zapytaniu.
- Kolumna "Operatory" wylicza operatory, które mogą być użyte w wyrażeniu.

Aby wstawić "Funkcję", "Pole" lub "Operator" w wyrażeniu:


- 1 Wybierz funkcję, pole lub operator.
- 2 Kliknij .

Gdy wyrażenie jest zdefiniowane, kliknij przycisk **OK**, aby przenieść je do zakładki **Filtr (warunek WHERE)** w szczegółach zapytania.


Stałe

Stałe *x* są wartościami stałymi, które można przypisać do warunku wyboru. Na przykład, jeśli wyszukujesz wszystkie modele o marce **3Com**, przypisujesz wartość stałą **3Com** do pola powiązanego **Brand.Name** (nazwa SQL: Brand) w tabeli modeli.

Aby wybrać stałą:

- 1 Kliknij ikonę .
- 2 Zostaje wyświetlone okno wyboru, które pokazuje wartości obecne w bazie danych dla pola określonego jako warunek wyszukiwania.

 **Uwaga:**

Nawet w przypadku pól typu "Spis", okno wyświetlone poprzez kliknięcie ikony  pokazuje tylko te wartości, które są użyte w bazie danych.

Składnia AQL

Korzystanie z AQL wymaga znajomości języka SQL. Szczegółowy opis składni SQL jest jednak poza zakresem tego podręcznika. W celu uzyskania dalszych informacji, należy sprawdzić odpowiednią dokumentację.

Konwencje

Oto konwencje obowiązujące przy używaniu składni AQL:

Tabela 3.1. AQL- konwencje dotyczące składni

[]	Nawiasy kwadratowe otaczają pozycje opcjonalne. Nie pisz w tych nawiasach.
< >	Nawiasy te otaczają pozycje logiczne. Nie pisz w tych nawiasach.
	Znak " " wskazuje, że wybory wzajemnie się wykluczają.
...	Wielokropek wskazuje, że tekst go poprzedzający może być powtórzony jeden raz lub kilka razy.
FROM	Wyrażenia pisane dużymi literami oznaczają stałe.

Składnia zapytań

Zapytania proste

SELECT [DISTINCT] [FIRST_ROWS] <lista wyboru>

[Warunek FROM]

[Warunek WHERE]

[Warunek GROUP BY]

[Warunek HAVING]

[Warunek ORDER BY]

Zapytania podrzędne

AQL obsługuje stosowanie zapytań podrzędnych zamiast pól.

 Uwaga:

W zapytaniach podrzędnych, instrukcja SELECT autoryzuje tylko pojedyncze wyrażenie.

(SELECT [DISTINCT] <wyrażenie>

[Warunek FROM]

[Warunek WHERE]

[Warunek GROUP BY]

[Warunek HAVING]
)

 **Ostrzeżenie:**

Zapytania podrzędne muszą być zawarte w nawiasach.

Przykład wykorzystania:

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

Zapytania typu UNION

UNION pozwala na grupowanie wyników kilku zapytań:

SELECT <lista wyboru>

[Warunek FROM]

[Warunek WHERE]

[Warunek GROUP BY]

[Warunek HAVING]

[UNION | UNION ALL | INTERSECTS | MINUS]

SELECT <lista wyboru>

[Warunek FROM]

[Warunek WHERE]

[Warunek WHERE]

[Warunek GROUP BY]

[Warunek HAVING]...

[Warunek ORDER BY]

Elementy zapytania

Pola i łącza

Zapytania są związane z polami i łączami w bazie danych AssetCenter.

Możesz wskazać nazwę pola:

- W odwołaniu do tabeli początkowej zapytania. W tym przypadku, nie jest konieczne określanie nazwy tej tabeli:

[Łącze. ...[Łącze.]]<pole>

Przykłady z tabeli **Elementy repozytorium (AmPortfolio)**:

Model
User.Name
User.Location.Name

- Jako odwołanie bezwzględne. W tym przypadku, musisz wskazać nazwę tabeli, do której należy pole:
 - Albo przez zadeklarowanie tabeli w warunku **FROM** i użycie jej nazwy (lub możliwego aliasu):
 - <tabela.[Łącze...]**pole**>
 - <alias.[Łącze...]**pole**>
 - Albo przez nie deklarowanie tabeli w warunku **FROM** i użycie zamiast tego ":":
 - <tabela:[Łącze...]**pole**>
 - <tabela[_alias]:[łącze[_alias]...]**pole**>

Te ostatnie dwie notacje są szczególnie użyteczne, jeśli nie możesz użyć warunku **FROM**.

Na przykład, podczas pisania zapytania w AssetCenter, masz dostęp tylko do warunku **WHERE**. Tabela początkowa zapytania jest ukryta (tabela, do której jest zastosowany filtr, pole **Tabela (TableName)** w szczegółach zapytania, itp.). Może jednak zajść konieczność użycia innych tabel w zapytaniu. W tym przypadku, notacja ":" pozwala to zrealizować.

Stałe

Dla stałych, które mogą być używane w zapytaniach, obowiązuje następująca składnia.

Stałe numeryczne

Kropka jest używana jako separator dziesiętny.

Przykłady:

12

52.23

Stałe typu tekst

Są one zawarte w pojedynczym cudzysłowie.

Przykłady:

'Komputer'

'Monitor'

Stałe typu data lub czas

Stałe typu data lub czas są zawarte pomiędzy znakami #. Ich format musi uwzględniać następujące zasady:

- Lata są wyrażone 4 cyframi.
- Daty są wyrażone jako Rok-Miesiąc-Dzień.
- Czas jest wyrażony jako Godziny-Minuty-Sekundy.
- Jest stosowany zegar 24 godzinny (a nie zegar 12 godzinny z godzinami przed południem lub po południu).
- Separator stosowany w datach to znak "/" lub "-".
- Separator stosowany dla czasu to znak ":".
- Miesiące, dni, godziny, minuty, i sekundy są wyrażone jako liczby 2-cyfrowe.
- Jeśli data i czas są wyrażone łącznie, data zawsze poprzedza czas i są one rozdzielone spacją.

Przykłady:

#rrrr-mm-dd gg:mm:ss#

#rrrr-mm-dd#

#gg:mm:ss#

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

Wyrażenia

Wyrażenia są tworzone przy użyciu:

- Stałych,
- Pól,
- Funkcji,
- Zapytań podrzędnych.

Można łączyć te elementy z operatorami i nawiasami w celu konstruowania złożonych wyrażeń.

Porównywanie wyrażeń przybiera formę:

<wyrażenie> <operator porównania> <wyrażenie>

Wyrażenia logiczne przybierają formę:

<operator porównania> <AND | OR> <operator porównania>

Można stosować nawiasy do łączenia kilku wyrażeń logicznych.

Operatory

Operatory logiczne

Operatory logiczne są używane do łączenia dwóch wyrażeń:

Tabela 3.2. AQL - operatory logiczne

Operator	Znaczenie
AND	Logiczne "AND"
OR	Logiczne "OR"

W celu zoptymalizowania zapytania, czasami mądrze jest unikać operatorów logicznych, jeśli zamiast tego może być użyty operator porównania. Następujący przykład ilustruje, w jaki sposób zoptymalizować filtr zapytania użyty do wybrania elementów repozytorium, których pole **Przypisanie (seAssignment)** jest ustawione jako **Oczekiwanie na dostawę** lub **Zwrot do serwisu**. Wartości tych dwóch elementów spisu systemowego to odpowiednio "3" i "4". Zatem można napisać:

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

Ostatnia wartość spisu systemowego wynosi "4", lepiej napisać zapytanie jak następuje:

```
seAssignment >=3
```

Operatory porównania

Operatory porównania są używane do porównywania dwóch wyrażeń.

Tabela 3.3. AQL- operatory porównania

Operator	Znaczenie
=	Równa się
<>	różne niż
=!	
>	większe niż
<	mniejsze niż
>=	większe lub równe
=<	mniejsze lub równe
=*	prawe połączenie zewnętrzne. Zastosowanie tego operatora jest ograniczone z powodu sposobu, w jaki AQL obsługuje łącza.
*=	lewe połączenie zewnętrzne. Zastosowanie tego operatora jest ograniczone z powodu sposobu, w jaki AQL obsługuje łącza.
LIKE	Działa jak operator = i pozwala również stosować znaki globalne.
NOT LIKE	Dostępne są następujące znaki globalne: "% " zastępuje każdy ciąg znaków. "_ " zastępuje każdy pojedynczy znak. Zależnie od używanego motoru bazy danych (systemy SQL Anywhere, SQL Server oraz Sybase obsługują ten operator, natomiast system Oracle for Workgroups nie obsługuje): [abc...] pozwala definiować listę możliwych znaków (bez spacji pomiędzy możliwymi wartościami). [a-c] pozwala definiować zakres możliwych wartości. DB2 nie obsługuje użycia operatora LIKE X, jeśli X zawiera nazwę SQL kolumny. Dla tego operatora obsługiwane są tylko stałe. Na przykład następujące zapytanie jest nieprawidłowe dla DB2: SELECT COL1, COL2 FROM TABLE1 WHERE COL1 LIKE COL2
IS NULL	Sprawdza, czy wartość pola jest "NULL", czy nie.
IS NOT NULL	Ważna uwaga: AssetCenter autoryzuje wartość "NULL" tylko dla pustych pól tekstowych, oraz dla pól Data lub Data+Czas , które nie są wypełnione.

 Uwaga:

System SQL Anywhere nie może przetwarzać warunków "LIKE X", jeśli X zawiera więcej niż 128 znaków. Jeśli X jest większe niż 128 znaków, zastosowanie zapytania wywołuje komunikat o błędzie ODBC. Problem ten może wystąpić na przykład podczas wyświetlania list w widoku drzewa, ponieważ operacja ta wykorzystuje warunek "LIKE" dotyczący pola "FullName".

Operatory specyficzne dla zapytań podrzędnych

Można porównać wartość z wynikami zapytania podrzędnego przy użyciu następujących operatorów:

- = ANY (zapytanie podrzędne)
- = ALL (zapytanie podrzędne)
- = SOME (zapytanie podrzędne)

Przykład:

- Następujące zapytanie dostarcza listę elementów repozytorium, których marka jest używana w biurze w Sandomierzu:

```
SELECT IModelId, Model.Brand FROM amPortfolio WHERE Model.Brand = ANY (SELECT Model.Brand FROM amPortfolio WHERE Location.FullName = '/Site Lyon')
```

Lista wyboru

Lista wyboru

Lista wyboru składa się z jednego, lub większej liczby wyrażeń rozdzielonych przecinkami:

<wyrażenie> [,<wyrażenie>...]

Każde wyrażenie może być związane z aliasem. Na przykład:

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

Jest to szczególnie użyteczne na poziomie eksportu zapytań, w celu przypisania nazwy do eksportowanych kolumn.

 Uwaga:

Niektóre systemy zarządzania bazą danych (DBMS) ograniczają liczbę wyrażeń w podanej instrukcji SELECT.

Warunek FROM

Warunek **FROM** wskazuje tabelę lub tabele, których dotyczy instrukcja **SELECT**.

AQL autoryzuje wykorzystanie aliasów dla nazw tabel.

Składnia

FROM <nazwa tabeli> [alias tabeli] [, <nazwa tabeli> [alias tabeli>] ...]

Tabela początkowa zapytania

Pierwsza tabela wskazana w warunku **FROM** zapytania jest tabelą początkową tego zapytania.

Jeśli zapytanie wykorzystuje pole, dla którego tabela nie jest określona, AQL przyjmuje, że należy ono do tabeli początkowej tego zapytania.

Warunek AQL **FROM** różni się od warunku o tej samej nazwie w SQL.

Na przykład, w następującym zapytaniu podrzędnym, AQL wyszukuje pole **Kod wewnętrzny (AssetTag)** w tabeli **Zasoby (amAsset)**:

```
SELECT AssetTag FROM amAsset
```

Liczba tabel w zapytaniu

Liczba tabel autoryzowanych w zapytaniu zależy od używanego systemu zarządzania bazą danych (DBMS).

Przykład:

- Oracle: można użyć dowolną ilość tabel.
- System Microsoft SQL Server lub Sybase SQL Server: użytkownik jest ograniczony w zapytaniu do 16 tabel.

! Ostrzeżenie:

Przy zliczaniu liczby tabel w zapytaniu, nie zapomnij uwzględnić tych tabel, które nie są jawnie wymienione, w szczególności, jeśli zapytanie używa łączy. Szukaj również notacji "fv_" (wyszukiwanie wartości cech), która generuje dodatkowe połączenie na poziomie systemu zarządzania bazą danych. Podobnie, notacja "cf_" (pola wyliczane) może generować dodatkowe połączenia.

Przykłady

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

Następujące zapytania są równoważne:

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

Warunek WHERE

Warunek AQL **WHERE** jest równoważny warunkowi **WHERE** w SQL. Określa on warunki wyszukiwania, podając elementy do pobrania z bazy danych. Warunki te można również wyrazić za pomocą warunków **WHERE** lub **HAVING**.

Składnia

WHERE <warunki wyszukiwania>

Redagowanie warunków wyszukiwania

W większości przypadków, należy wpisać te warunki stosując następujący format:

```
<WHERE | HAVING> [NOT] <wyrażenie> <operator porównania> <w
rażenie>
<WHERE | HAVING> [NOT] <wyrażenie logiczne>
<WHERE | HAVING> [NOT] <pole> [NOT] LIKE 'xxxxx'
```

```
<WHERE | HAVING> [NOT] <wyrażenie logiczne> <AND | OR> <wyrażenie logiczne>
```

```
<WHERE | HAVING> [NOT] <pole> IS [NOT] NULL
```

W niektórych przypadkach, może wystąpić potrzeba napisania bardziej złożonych zapytań, takich jak:

```
<WHERE | HAVING> [NOT] EXISTS (<zapytanie podrzędne>)
```

```
<WHERE | HAVING> [NOT] <wyrażenie> [NOT] IN (<lista wartości> | <zapytanie podrzędne>)
```

```
<WHERE | HAVING> [NOT] <wyrażenie> <operator porównawczy> <ANY | ALL> (<zapytanie podrzędne>)
```

Warunek GROUP BY

Warunek AQL **GROUP BY** jest równoważny warunkowi 'GROUP BY' w SQL.

Składnia

GROUP BY <;wyrażenie bez funkcji zbiorczychs> [, <;wyrażenie bez funkcji zbiorczychs>]...

Rada przy redagowaniu

GROUP BY określa podzbiory tabeli. Podzbiory są definiowane w warunku **GROUP BY** przez wyrażenie, które może być na przykład nazwą pola.

Jeśli w liście wyboru instrukcji **SELECT** zawarte są funkcje zbiorcze, **GROUP BY** wyszukuje wartość wynikową dla każdego podzbioru. Te wartości wynikowe mogą być użyte w warunku **HAVING**.

Jeśli zapytanie wykorzystuje warunek **GROUP BY**, każde wyrażenie listy wyboru musi dostarczać pojedynczą wartość dla każdego podzbioru.

GROUP BY - przykłady

Następujące zapytanie podaje liczbę ogółem marek obecnych w bazie danych. Dla każdego zasobu ze skojarzoną marką AssetCenter zwraca instancję dotyczącą marki.

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

Wykorzystując warunek **GROUP BY**, uzyskujemy listę marek oraz liczbę zasobów każdej marki:

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

Warunek HAVING

Warunek AQL **HAVING** jest równoważny warunkowi SQL 'HAVING'.

Składnia

HAVING <Warunki wyszukiwania>

Różnice z warunkiem WHERE

Warunek ten określa warunki wyszukiwania podobnie jak warunek **WHERE**. Te dwa warunki różnią się jednak następująco:

- Warunek **HAVING** określa ograniczenia, które mają być zastosowane do funkcji zbiorczych w liście wyboru. Ograniczenia wpływają na liczbę linii wynikowych, lecz nie wpływają na wyliczenia związane z funkcjami zbiorczymi.
- Jeśli zapytanie używa warunku **WHERE**, warunki wyszukiwania ograniczają linie podlegające funkcjom obliczeń zbiorczych, ale nie wpływają na linie wynikowe.

Przykłady

Przykład zapytania, gdzie warunek **WHERE** jest równoważny warunkowi **HAVING**:

Następujące zapytanie zwraca listę marek, których nazwa rozpoczyna się od litery po literze **B**, oraz liczbę zasobów każdej z tych marek:

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

To samo zapytanie można również wyrazić przy użyciu warunku **WHERE**:

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

Przykład zapytania używającego warunku **HAVING**:

Warunek **HAVING** umożliwia użycie funkcji zbiorczych (takich jak **Count**). Takiej możliwości nie daje warunek **WHERE**. A zatem, następujące zapytanie wyszukuje wszystkie marki reprezentowane przez więcej niż jeden zasób:

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

Warunek ORDER BY

Warunek AQL **ORDER BY** jest równoważny warunkowi SQL 'ORDER BY'.

Pozycje mogą być sortowane:

- W porządku rosnącym **ASC**. Jest to domyślny porządek sortowania.
- W porządku malejącym: **DESC**.

Składnia

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

Warunek INSERT

Warunek ten umożliwia dodanie jednego lub kilku rekordów do tabeli bazy danych.

Składnia

INSERT INTO <Nazwa tabeli> [alias tabeli] (<Nazwa pola> [, <Nazwa pola>]...) **VALUES** (<wyrażenie> [, wyrażenie]...) | **zapytanie podrzędne AQL**)

Warunek ten jest zawarty w API AssetCenter o nazwie AmDbExecAql.

Więcej informacji na temat API AssetCenter można znaleźć w podręczniku **Programmer's Reference**, w rozdziale **Reference**.

Przykład

Warunek **INSERT** może umożliwić uproszczenie kodu kreatora

Dodatkowe informacje nt. dostawy:

Kod kreatora nie stosującego warunku INSERT

```

hrAlarm = AmCreateRecord("amDateAlarm")
    lErr = AmSetFieldLongValue(hrAlarm, "bSecondLevel", 0)
    lErr = AmSetFieldLongValue(hrAlarm, "dtTrig1", AmGetFieldLongValue(hrAsset, 2)-IDaysBefore*86400)
    lErr = AmSetFieldLongValue(hrAlarm, "lAction1Id", lActionId)
    lErr = AmSetFieldLongValue(hrAlarm, "lMonitObjId", lAstId)
    lErr = AmSetFieldStringValue(hrAlarm, "MonitoredField", "dWarrEnd")
    lErr = AmSetFieldStringValue(hrAlarm, "MonitoredTable", "amAsset")
    lErr = AmSetFieldLongValue(hrAlarm, "sDaysBefore1", IDaysBefore)
    lErr = AmInsertRecord(hrAlarm)

```

Kod kreatora stosującego warunek INSERT

```

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

```

Warunek UPDATE

Warunek ten umożliwia zaktualizowanie jednego lub kilku pól rekordu w tabeli bazy danych.

Składnia

UPDATE <Nazwa tabeli> [alias tabeli] **SET** (<nazwa pola> [, <nazwa pola> ...]) [Warunek FROM] [Warunek WHERE]

Przykład

Warunek **UPDATE** może umożliwić uproszczenie kodu działania, które wywołuje polecenie:

Kod działania nie stosującego warunek UPDATE

```
hr = AmGetRecordFromMainId("amPOrder", [IPordId])
  lErr = AmSetFieldLongValue(hr, "seStatus", "$(IDS_POSTATUS_OR
DERED)")
  lErr = AmUpdateRecord(hr)
```

Kod działania stosującego warunek UPDATE

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

Warunek DUPLICATE

Warunek ten umożliwia powielenie istniejącego rekordu w tabeli bazy danych.

Funkcja ta jest właściwa dla AssetCenter.

Więcej informacji na ten temat można znaleźć w podręczniku **Ergonomia**, rozdział **Operacje dotyczące rekordów**, podrozdział **Powielanie rekordu**.

Składnia

DUPLICATE <Nazwa tabeli> [alias tabeli] **SET** (<nazwa pola> [, <nazwa> ...]) [Warunek FROM] [Warunek WHERE]

Warunek DELETE

Warunek ten umożliwia usunięcie pól rekordu w tabeli bazy danych.

Składnia

DELETE [Warunek FROM] [Warunek WHERE]

Odwołanie funkcji AQL

Następujące funkcje AQL mogą być użyte w zapytaniach i wzorach:

- Funkcje AQL typu "Ogółem"
- Funkcje AQL typu "Ciąg znaków"
- Funkcje AQL typu "Data"
- Funkcje AQL typu "Numeryczne"
- Funkcje AQL typu "Test"

 Uwaga:

Możesz również używać wewnętrznych funkcji SQL swojego systemu zarządzania bazą danych. W tym przypadku, kod wynikowy nie jest przenośny.

Funkcje AQL typu "Ogółem"

Tabela 3.4. AQL - funkcje typu Ogółem

Funkcja	Opis
Avg(<kolumna>)	Zwraca średnią wartość kolumny typu "liczba". Zwraca "0", jeśli kolumna nie zawiera żadnych rekordów.
Count(<kolumna>)	Zlicza w kolumnie wartości nie będące wartościami pustymi.
Countdistinct(<kolumna>)	Zlicza w kolumnie różne wartości nie będące wartościami pustymi.
Max(<kolumna>)	Zwraca wartość maksymalną w kolumnie typu "liczba", "ciąg znaków" lub "data". Jeśli kolumna nie zawiera żadnych rekordów, zwraca "0" (kolumna typu "liczba"), "pusty ciąg znaków" (kolumna typu "ciąg znaków"), lub "pusta data" (kolumna typu "data").
Min(<kolumna>)	Zwraca wartość minimalną w kolumnie typu "liczba", "ciąg znaków" lub "data". Jeśli kolumna nie zawiera żadnych rekordów, zwraca "0" (kolumna typu "liczba"), "pusty ciąg znaków" (kolumna typu "ciąg znaków"), lub "pusta data" (kolumna typu "data").
Sum(<kolumna>)	Zwraca sumę wartości kolumny typu "liczba". Zwraca "0", jeśli kolumna nie zawiera żadnych rekordów.

Funkcje te używają warunków "GROUP BY" oraz "HAVING" łącznie.

Funkcje AQL typu "Ciąg znaków"

Tabela 3.5. AQL - funkcje typu Ciąg znaków

Funkcja	Opis
Ascii(<Ciąg znaków>)	Zwraca wartość ASCII pierwszego znaku <ciągu>.
Char(<n>)	Zwraca znak o kodzie ASCII "n".
Left(<Ciąg znaków>, <n>)	Zwraca "n" pierwszych znaków <ciągu>
Lower(<Ciąg znaków>)	Zwraca <ciąg> w małych literach.
Ltrim(<Ciąg znaków>)	Usuwa spacje po lewej stronie <ciągu>.
Right(<Ciąg znaków>, <n>)	Zwraca "n" ostatnich znaków <ciągu>.
Rtrim(<Ciąg znaków>)	Usuwa spacje po prawej stronie <ciągu>.
Substring(<Ciąg znaków>, <n1>, <n2>)	Pobiera podciąg znaków rozpoczynając od znaku "n1" w <ciągu> i o długości "n2" (1 znak <ciągu> jest traktowany jako znak numer 1).
Upper(<Ciąg znaków>)	Zwraca <ciąg> w dużych literach.

Funkcje AQL typu "Data"

Tabela 3.6. AQL - funkcje typu Data

Funkcja	Opis
Year(<date>)	Zwraca liczbę oznaczającą rok dla pola typu "data" lub "data i czas" (np.: 2000).
Month(<date>)	Zwraca numer miesiąca dla pola typu "data" lub "data i czas" (1=styczeń, ..., 12=grudzień).
Day(<date>)	Zwraca numer dnia w miesiącu dla pola typu "data" lub "data i czas" (1-31).
DayOfYear(<date>)	Zwraca numer dnia w roku dla pola typu "data" lub "data i czas" (1-366).

Funkcja	Opis
WeekDay(<data>)	Zwraca numer dnia w tygodniu dla pola typu "data" lub "data i czas". Numer ten zależy od tego, jak jest skonfigurowany serwer. Na przykład, domyślna konfiguracja w systemie Sybase lub Microsoft SQL Server to (1=niedziela, 2=poniedziałek, ..., 7=sobota). Domyślna konfiguracja w systemie Oracle to (1=poniedziałek, ..., 7=niedziela).
Hour(<godzina>)	Zwraca liczbę godzin w dniu dla pola typu "czas" lub "data i czas" (0-23).
Minute(<godzina>)	Zwraca liczbę minut dla pola typu "czas" lub "data i czas" (0-59).
Second(<godzina>)	Zwraca liczbę sekund dla pola typu "czas" lub "data i czas" (0-59).
Getdate()	Zwraca bieżącą datę systemową serwera.
AddDays(<data>, <liczba>)	Dodaje daną liczbę dni do pola typu "data" lub "data i czas".
AddHours(<data>, <liczba>)	Dodaje daną liczbę godzin do pola typu "data" lub "data i czas".
AddMinutes(<data>, <liczba>)	Dodaje daną liczbę minut do pola typu "data" lub "data i czas" type field.
AddSeconds(<data>, <liczba>)	Dodaje daną liczbę sekund do pola typu "data" lub "data i czas".
DaysDiff(<data1>, <data2>)	Liczba dni pomiędzy datami data1 i data2 (liczba "zmiennoprzecinkowa" z miejscami po przecinku).
HoursDiff(<data1>, <data2>)	Liczba godzin pomiędzy datami data1 i data2 (liczba "zmiennoprzecinkowa" z miejscami po przecinku).
MinutesDiff(<data1>, <data2>)	Liczba minut pomiędzy datami data1 i data2 (liczba "zmiennoprzecinkowa" z miejscami po przecinku).
SecondsDiff(<data1>, <data2>)	Liczba sekund pomiędzy datami data1 i data2 (liczba "zmiennoprzecinkowa" z miejscami po przecinku).
DbToLocalDate(<data>)	Konwertuje datę wyrażoną w strefie czasowej serwera bazy danych na datę wyrażoną w strefie czasowej zdefiniowanej na poziomie komputera klienta.
LocalToDbDate(<data>)	Konwertuje datę wyrażoną w strefie czasowej komputera klienta na datę wyrażoną w strefie czasowej serwera bazy danych.

Tabela 3.7. AQL - przykłady funkcji typu Data

Opis	Język zapytań AssetCenter
Wszystkie rekordy zmodyfikowane w trakcie ubiegłego tygodnia.	AddDays(dtLastModif,7)>=Getdate()
Wszystkie zlecenia pracy zgłoszone w ciągu ostatniej godziny.	HoursDiff(Getdate(), dtNotif) <= 1 lub HoursDiff(Getdate(), dtNotif) <= 1
Wszystkie zlecenia pracy zamknięte w ciągu ostatnich 30 minut.	MinutesDiff(Getdate(), dtActualFixed) <= 30 lub AddMinutes(dtActualFixed, 30) >= Getdate()

Następujące zapytanie wymienia zlecenia pracy skutecznie zrealizowane i załatwione w trakcie tego samego dnia. Strefa czasowa komputera klienta jest uwzględniona:

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

Następujące zapytanie wymienia wszystkie zlecenia pracy, które zostały dziś skutecznie rozpoczęte:

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

Funkcje AQL typu "Numeryczne"

Tabela 3.8. AQL - funkcje typu numerycznego

Funkcja	Opis
Abs(<Liczba>)	Zwraca wartość bezwzględną "liczby".
Ceil(<Nombre>)	Zwraca najmniejszą liczbę całkowitą większą lub równą niż "liczba".
Floor(<Liczba>)	Zwraca największą liczbę całkowitą mniejszą lub równą niż "liczba".
Mod(<a>,)	Zwraca resztę dzielenia "a" przez "b" (a = qb + r, gdzie q to liczba całkowita i 0 < r < q).

Funkcja	Opis
Round(<a>, <n>)	Zaokrągla "a" do "n" miejsc po przecinku
Trunc(<a>, <n>)	Obcina "a" do "n" miejsc po przecinku.

Przykłady zastosowania:

Abs (2.516) = 2.

Ceil (2.516) = 3.

Floor (2.516) = 2.

Mod (6,4) = 2.

Round (31.16, 1) = 31.20.

Round (31.16, 0) = 31.00.

Round (31.16, -1) = 30.00.

Trunc (31.16, 1) = 31.1.

Funkcje AQL typu "Test"

Tabela 3.9. AQL - funkcje typu Test

Funkcja	Opis
IsNull(<a>,)	Jeśli "a" jest "Null", zamienia "a" na "b". Typy "a" i "b" muszą być zgodne.

Przykłady zapytań

Każdy z następujących przykładów dotyczy specyficznego aspektu projektowania zapytań. Przykłady te można modyfikować lub łączyć, aby wykorzystać je jako podstawę swoich własnych zapytań.

Przykłady te podają pełną składnię zapytania. Aby przetestować je bez zmian, należy użyć programu AssetCenter Export. Trzeba będzie zmodyfikować składnię tych przykładów, aby użyć ich na przykład w filtrze zapytania.

A zatem, następujące zapytanie podane w całości:

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

należy zapisać następująco, jeśli jest użyte w filtrze zapytania (tylko warunek WHERE jest jawnie podany) zastosowanym do tabeli zasobów:

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

Dalsze przykłady zapytań są zawarte w demonstracyjnej bazie danych dostarczonej z AssetCenter.

Uwaga:

Aby obejrzeć transkrypcję zapytania w odpowiadającym kodzie SQL systemu zarządzania bazą danych, którego używasz, wyświetl zakładkę **Podgląd** w szczegółach zapytania.

Aby porównać pole w głównej tabeli z wartością

Przykład: wszystkie elementy repozytorium o marce "Compaq".

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

Aby porównać łącze w głównej tabeli z innym łączem

Przykład: wszystkie elementy repozytorium o tej samej lokalizacji, jak ich zasób nadrzędny.

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

Aby porównać łącze w głównej tabeli z wartością

Przykład: Wszystkie wydziały i pracownicy z bezpośrednim łączem do "Przedstawicielstwa w Tarnowie".

```
SELECT Self FROM amEmplDept WHERE Parent.Name = 'Przedstawicielstwo w Tarnowie'
```

Aby porównać według pola w tabeli połączonej z główną tabelą

Przykład: wszystkie zasoby, które mają taką samą nazwę lokalizacji, jak ich lokalizacja nadrzędna.

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

Tabele hierarchiczne

Wykorzystanie pola "FullName"

Przykład: Wszystkie lokalizacje podrzędne lokalizacji o nazwie "Biurowiec Polaris":

```
SELECT Self FROM amLocation WHERE FullName LIKE '/Biurowiec Polaris/%'
```

Wykorzystanie pól "FullName" i "sLvl"

Zapytania dotyczące tabel hierarchicznych często wykorzystują pola "FullName" i "sLvl".

Przykład: Wszystkie lokalizacje podrzędne lokalizacji "Biurowiec Polaris", o poziomie hierarchii niższym niż 3.

Wartość hierarchiczna głównego poziomu struktury drzewa jest równa "0".

```
SELECT Self FROM amLocation WHERE (FullName LIKE '/Biurowiec Polaris/%') AND (sLvl < 3)
```

Należy zwrócić szczególną uwagę na znaki "/", które pojawiają się na początku oraz na końcu pełnych nazw.

Zapytanie łączące dwa warunki

Przykład: Wszyscy pracownicy na stanowisku "Specjalista d.s. kontaktów z klientem" i umiejscowienie w "Biurowie w Tarnowie".


```
SELECT Self FROM amEmplDEpt WHERE (Title = 'Specjalista d.s. konta
któw z klientem') AND (Location.Name = 'Biuro w Tarnowie')
```

Porównania pola z liczbami, datami lub tekstem

Przykład: wszystkie zlecenia pracy zrealizowane pomiędzy 1 stycznia 2003 i 31 grudnia 2003.

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

Zapytanie dotyczące cechy

Przykład: wszystkie elementy repozytorium, których cecha o nazwie SQL `fv_Size` pokazuje wartość większą lub równą 150 cm.

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

Aby wyszukać rekordy zgodne z wyrażeniem

Przykład: Wszystkie zasoby, których cena nabycia jest równa najwyższej cenie nabycia zawartej bazie danych. Zwróć uwagę, że zapytanie podrzędne jest użyte w zapytaniu głównym w celu zidentyfikowania tej maksymalnej ceny.

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

Aby wyszukać pole, które nie jest wypełnione

Przykład: Wszyscy pracownicy bez numeru telefonu. Zwróć uwagę, że pusty ciąg jest reprezentowany przez dwa pojedyncze cudzysłowy ('').

```
SELECT Self FROM amEmplDEpt WHERE Phone=""
```

Aby wyszukiwać brak łącza

Przypadek łącza 1

Przykład: wszystkie elementy repozytorium, które nie zostały przypisane do użytkownika. Zwróć uwagę, że brak łącza jest oznaczony przez "0".

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

Przypadek łączy n

Przykład: wszystkie modele, które nie mają zasobów powiązanych.

```
SELECT self FROM amModel WHERE NOT ( EXISTS (SELECT A1.IAstl  
d FROM amAsset A1 WHERE A1.IModelId = amModel.IModelId))
```

Zapytanie to skanuje tabelę modeli, pobiera kolejno każdy model i porównuje z liczbą 0 zasobów należących do tego modelu.

Przykład łączący sprawdzenie dotyczące łącza 1 oraz łącza n

Przykład: wszystkie modele bez modelu nadrzędnego, ani bez modelu podrzędnego:

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

Zapytanie to wykonuje:

- Sprawdzenie łącza 1 ("Parent = 0"), aby wybrać te modele, które nie mają modelu nadrzędnego.
- Sprawdzenie łącza n ("0 = (SELECT COUNT(a.IModelId) FROM amModel a WHERE a.IParentId = IModelId)", aby wybrać te modele, które nie mają modeli podrzędnych. Sprawdzenie łącza n pobiera każdy model, wybiera jego identyfikator "IModelId", i zlicza wszystkie modele, których identyfikator "IParentId" jest równy "IModelId".

Przykład drugi

Wszystkie modele bez modelu podrzędnego o naturze "Komputer".

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

 **Uwaga:**

W przypadku stosowania tego zapytania w AssetCenter Export, wyświetla się komunikat o błędzie. Komunikat można zignorować. Zapytanie funkcjonuje prawidłowo.

Zapytanie z aliasem

Przykład: Wszyscy pracownicy, którzy wzięli udział w programie szkoleniowym 'Peregrine' oraz w programie szkoleniowym 'Baza danych'.

Tabela początkowa: Tabela wydziałów i pracowników.

Zapytanie jest następujące:

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

Alias, wyrażone jako "Szkolenie_1" oraz "Szkolenie_2", pozwalają zdefiniować dwa warunki dotyczące dwóch różnych rekordów związanych łączem "Szkolenie".

Gdyby napisać:

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

Zostaliby wybrani wszyscy pracownicy, którzy brali udział w obu kursach szkoleniowych, zarówno o jednej, jak i o drugiej nazwie.

Gdyby napisać:

```
SELECT Self FROM amEmplDept WHERE (Trainings.Title = 'Peregrine')
OR (Trainings.Title = 'Baza danych')
```

Zostaliby wybrani wszyscy pracownicy, którzy brali udział w jednym kursie szkoleniowym, o jednej z tych dwóch nazw.

4 Formularze

ROZDZIAŁ

Niniejszy rozdział wyjaśnia, w jaki sposób projektować formularze za pomocą AssetCenter.

Lista formularzy jest dostępna w menu **Narzędzia/ Przetwarzanie raportów/ Formularze**.

Definicja formularza

Formularz jest wzorem dokumentu, który pozwala drukować dane. W odróżnieniu od raportów Seagate Crystal Reports, formularze są tworzone bezpośrednio w AssetCenter.

Instalacja prekonfigurowanych formularzy w roboczej bazie danych

Wraz z AssetCenter są dostarczane prekonfigurowane formularze. Są one już zainstalowane w demonstracyjnej bazie danych. Należy je natomiast dodać do operacyjnej bazy danych przy jej tworzeniu lub aktualizacji.

Import formularzy podczas tworzenia bazy danych

Aby zaimportować formularze podczas tworzenia bazy danych

- 1 Uruchom AssetCenter Database Administrator.
- 2 Wybierz menu **Plik/ Otwórz**.
- 3 Wybierz opcję **Otwórz plik opisowy bazy danych - utwórz nową bazę danych**.
- 4 Wybierz plik gbbase.dbb, znajdujący się w folderze podrzędnym config folderu instalacyjnego oprogramowania AssetCenter.
- 5 Wybierz menu **Działanie/ Utwórz bazę danych**.
- 6 Wypełnij wszystkie pola umożliwiające utworzenie bazy danych przy użyciu podręcznika **Administracja**, rozdział **Tworzenie bazy danych AssetCenter**.
- 7 Wybierz opcję **Import dodatkowych danych**.
- 8 W ramce **Dane do zaimportowania** wybierz **Formularze**.
- 9 Kliknij przycisk **Utwórz**, aby zatwierdzić utworzenie bazy danych.

Import formularzy do istniejącej bazy danych

Aby zaimportować formularze do istniejącej bazy danych:

- 1 Uruchom AssetCenter Database Administrator.
- 2 Wybierz menu **Plik/ Otwórz**.
- 3 Wybierz opcję **Otwórz plik opisowy bazy danych - utwórz nową bazę danych**.

- 4 Wybierz plik gbbase.dbb, znajdujący się w folderze podrzędnym config folderu instalacyjnego oprogramowania AssetCenter.
- 5 Wybierz menu **Działanie/ Utwórz bazę danych**.
- 6 Wybierz bazę danych w polu **Baza danych**.
- 7 Wybierz opcje **Tworzenie bazy danych i Tworzenie danych systemowych**.
- 8 Wybierz opcję **Import dodatkowych danych**.
- 9 W ramce **Dane do zaimportowania** wybierz **Formularze**.
- 10 Kliknij przycisk **Utwórz**, aby zatwierdzić import formularzy.

Tworzenie formularzy

Lista formularzy jest dostępna w menu **Narzędzia/ Przetwarzanie raportów/ Formularze**.

Informacje podstawowe

- 1 Wprowadź nazwę formularza
- 2 Wybierz typ formularza: (lista lub szczegóły)
Oba typy mogą zawierać tekst i predefiniowane obrazy
Różnice pomiędzy tymi dwoma typami:
 - lista: pozwala wydrukować listę rekordów, tak jak jest ona wyświetlona na aktywnej liście (zgodnie z kolumnami, które lista zawiera, oraz zastosowanymi filtrami).
 - lista: pozwala wydrukować listę rekordów, tak jak jest ona wyświetlona na aktywnej liście (zgodnie z kolumnami, które lista zawiera, oraz zastosowanymi filtrami).
- 3 Wybierz główną tabelę dla formularza.

Ostrzeżenie:






Pole **Tabela** (nazwa SQL: TableName) pozwala programowi AssetCenter proponować tylko te formularze, które odnoszą się do danej listy.

Edytowanie formularzy i obiektów

Aby przeprowadzić edycję formularza, należy zdefiniować obiekty i umieścić je na stronie.

Aby wstawić nowy obiekt na stronie:

- 1 Wybierz zakładkę Formularz.
- 2 Kliknij ikonę obiektu po lewej stronie strony.

Ikona	Funkcja
	Aby wybrać obiekt w formularzu, na przykład w celu jego modyfikacji.
	Aby dodać tekst stały oraz zmienne niezależne od rekordów, które są drukowane (na przykład, bieżąca data).
	Aby dodać obraz.
	Aby wstawić wzór zawierający wartości pola oraz ciągi tekstu stałego.
	Aby wstawić listę rekordów. Narzędzia te pozwalają umiejscowić listę na stronie. Dla formularzy szczegółów, pozwalają również zdefiniować tabelę związaną zawierającą rekordy, oraz listę pól do wydrukowania.

- 3 Umieść wskaźnik myszy na stronie.
- 4 Kliknij lewy przycisk myszy.
- 5 Zakreśl ramkę myszą: Ramka ta określa miejsce zarezerwowane dla obiektu.
- 6 Kliknij podwójnie pozycję obiektu. Zostanie wyświetlone okno opisujące właściwości obiektu.
- 7 Zdefiniuj właściwości obiektu.
- 8 Kliknij przycisk **Modyfikuj**.

W formularzu można wstawiać następujące obiekty:

Tekst stały

Dotyczy to tekstu, który jest niezależny od drukowanych rekordów. Można łączyć dowolny rodzaj znaków, jak również zmienne:

\$D: Drukowana data

\$U: nazwisko użytkownika AssetCenter, który drukuje formularz.

\$C: Numer strony.

\$N: Liczba ogółem drukowanych stron.

 **Ostrzeżenie:**

Ważne: Nie umieszczaj cudzysłowów wokół tekstu.

Przykład:

Dokument wydrukowany \$D przez \$U

Wzory

Wzory są dostępne tylko w formularzach typu "szczegóły".

Wzory łączą ze sobą:

- wartości pól bazy danych AssetCenter.
- tekst stały otoczony cudzysłowami.

Przykład:

"Zasob: " CodInt " / " / Marka

Wzory nie pozwalają wykonywać obliczeń.

Listy

- Formularze typu "lista": Można dołączyć tylko jedną listę. Po wybraniu polecenia menu **Plik/ Drukuj** lista ta będzie zastąpiona listą w aktywnym oknie.
- Formularze typu "lista": Można dołączyć tylko jedną listę. Po wybraniu polecenia menu **Plik/ Drukuj** lista ta będzie zastąpiona listą w aktywnym oknie.

Obrazy

Można wstawiać obrazy (np. logo).

 Uwaga:

Przy edycji formularza typu "lista" nie można wybrać pól do wyświetlenia. AssetCenter drukuje pola, które pojawiają na listach się w postaci kolumn. Gdy nakreślasz formularz typu "Szczegóły", możesz pojedynczo wybierać pola do wydrukowania.

Właściwości obiektów w formularzach

Położenie i rozmiar

Aby zmienić położenie i rozmiar obiektu, wystarczy po prostu przeciągnąć obiekt lub jego krawędzie.

 Uwaga:

Możesz przenosić i zmieniać rozmiar kilku obiektów równocześnie, poprzez ich jednoczesne zaznaczenie. Jeśli przeciągniesz myszą ramkę obszaru zaznaczenia wokół kilku obiektów, wszystkie obiekty w ramce będą wybrane; możesz też pojedynczo wybierać obiekty, przytrzymując wciśnięty przycisk SHIFT.

Właściwości

Gdy podwójnie klikniesz obiekt, lub użyjesz polecenia menu **Formularze wewnętrzne/ Właściwości**, zostanie wyświetlona lista właściwości wybranego obiektu.

Lista właściwości ma dwie kolumny: Pierwsza kolumna zawiera nazwy właściwości, natomiast druga pozwala edytować wartości właściwości.

Aby zmodyfikować właściwość, kliknij na drugiej kolumnie lewym przyciskiem myszy.

Proste właściwości mogą być edytowane bezpośrednio (tekst, wzory, listy, kolory tła, kolory tekstu, wyrównanie tekstu, wyrównanie obiektu). Dla złożonych właściwości (czcionka dla tekstu lub wzoru, ramka, zawartość listy, obraz) wyświetlane są dodatkowe okna.

Tekst

Wprowadź swój tekst bezpośrednio.

Wzór

Aby pomóc sobie przy tworzeniu wzorów, możesz wyświetlić listę rozwijaną z drzewem wyświetlającym pola, które są dostępne i zgodne z typem wybranego formularza. Kliknięcie węzła w drzewie zastępuje bieżące zaznaczenie polem wybranym na liście. Możesz wstawiać tekst pomiędzy polami pod warunkiem, że dołączasz cudzysłowy.

Kolor tła, kolor tekstu

Można wybierać spośród 16 dostępnych kolorów.

Wyrównanie tekstu


Określa ono, w jaki sposób tekst jest wyrównywany wewnątrz ramki. Lista rozwijana wyświetla różne rodzaje wyrównania (Wyśrodkowany, Wyrównanie do lewej, Wyrównanie do prawej).

Wyrównanie strony


Określa poziome wyrównanie tekstu na stronie. Okno kontekstowe pozwala wybrać wyrównanie:

- **Do lewej.**
- **Do prawej.**
- **Wyśrodkowane.**
- **Wyrównanie względne:** W tym przypadku, obiekt zachowuje położenie, które zostało określone na ekranie tworzenia formularza.

Czcionka

Aby wybrać zestaw znaków obiektu oraz jego rozmiar, kliknij  obok edytowanej komórki.

Ramka

Aby dodać ramkę wokół obiektów, kliknij przycisk  w komórce edycji. Zostanie wyświetlony ekran konfiguracji dla ramki.

Styl **3D** nadaje ramce efekty uwypuklenia.

Dla ramek bez stylu 3D możesz wybrać, które krawędzie będą wyświetlone, ich kolor oraz grubość.

Obraz

Aby wstawić obraz:

- 1 Kliknij przycisk w komórce edycji.
- 2 Wybierz plik graficzny na ekranie "Wybierz obraz".

Łącze w liście

Aby wybrać listę, która ma być wyświetlana w formularzu, użyj listy rozwijanej. Kliknij żądaną listę (na przykład listę zasobów używanych przez pracownika).

Zawartość listy

 Uwaga:

Zawartość listy można konfigurować tylko podczas edytowania formularza szczegółów.

Aby zdefiniować zawartość listy, kliknij przycisk w komórce edycji. Zostanie wyświetlone okno konfiguracji. Pokazane są wszystkie kolumny na liście.

Ekran ten pozwala zdefiniować:

- tytuł każdej kolumny,
- wzór stosowany do określania zawartości kolumny,
- rozmiar kolumny (procent zajęty przez kolumnę)
- czcionkę tytułu i zawartości każdej kolumny,
- wyrównanie tytułu oraz zawartości każdej kolumny,
- separatory pionowy i poziomy.

Aby usunąć kolumnę z listy, użyj przycisku "Delete".

Aby wstawić kolumnę do listy, wykonaj edycję ostatniej linii listy.

Każdą komórkę można edytować stosując te same zasady, jak dla listy właściwości.

Siatka wzorca

Siatka wzorca składa się z linii pionowych i poziomych, które pokrywają tło ekranu.

Polecenie menu kontekstowego **Formularz/ Siatka** pozwala:

- pokazywać i ukrywać siatkę wzorca.
- określać odstęp pomiędzy liniami w siatce.

Pokazane są tylko punkty przecięcia linii poziomych i pionowych. Odstęp pomiędzy liniami wyznacza dokładność, z którą można umieszczać obiekty na stronie.

Ustawienia strony formularza

Menu kontekstowe **Formularz/ Ustawienia strony** pozwala określić:

- Format strony.
- Układ **Pionowa** lub **Pozioma**.
- Marginesy dokumentu.
- Nagłówek i stopkę.

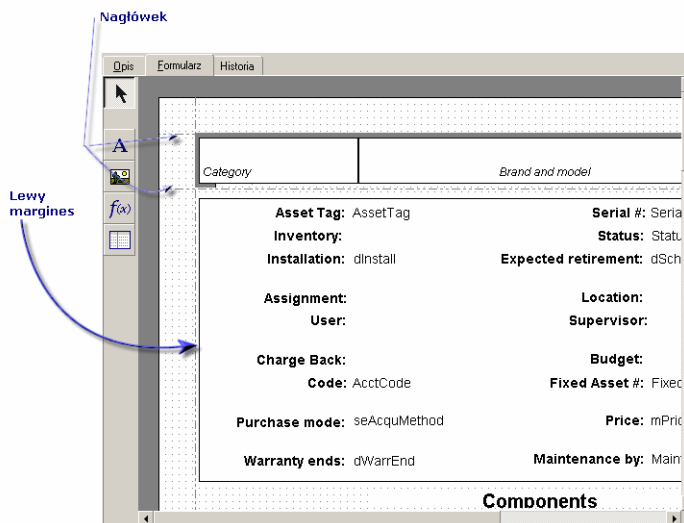
Uwaga:

Marginesy dokumentu, obszary stopki i nagłówka mogą być modyfikowane bezpośrednio w obszarze edycji. Po prostu przeciągnij znacznik marginesu oraz granice obszaru dla nagłówka i stopki, które są pokazane za pomocą linii kropkowanych.

Aby wstawić tekst w nagłówkach i stopkach strony:

- 1 Wybierz polecenie menu **Formularz/ Ustawienia strony**.
- 2 Zaznacz pola **Nagłówek** i **Stopka**.
- 3 Kliknij przycisk **OK**.
- 4 Przejdź w formularzu do obszaru nagłówka lub stopki (obszary nagłówka i stopki są pokazane przy użyciu linii kropkowanych).
- 5 Wstaw obiekty tutaj, lub przenieś obiekty z głównego obszaru strony.
- 6 Zatwierdź modyfikacje klikając przycisk **Modyfikuj**.

Rysunek 4.1. Formularze - nagłówki



Uwaga:

Nie można przenieść obiektów z obszarów nagłówka i stopki do głównego obszaru strony.

Nie można przenieść obiektów z obszarów nagłówka i stopki do głównego obszaru strony.

Aby regularnie tworzyć raporty zaleca się:

- 1 utworzyć "widok" z odpowiednimi parametrami.
- 2 skojarzyć ten widok z formularzem.

Widok pozwala zdefiniować:

- Warunki sortowania.
- Filtr do zastosowania oraz wartości tego filtru.
- Listę widocznych kolumn.

Formularz pozwala zorganizować układ strony raportu.

Aby wydrukować raport:

- 1 Wyświetl uprzednio utworzony widok (polecenie menu **Narzędzia/Widoki**).
- 2 Wydrukuj posługując się wyświetlonym widokiem (menu **Plik/Drukuj**): wybierz "Typ" drukowania i właściwy "Formularz".

Identyfikowanie formularzy związanych z danym modulem

Aby zidentyfikować formularze związane z danym modulem:

- 1 Uruchom AssetCenter.
- 2 Wyświetl listę formularzy (**Narzędzia/Przetwarzanie raportów/Formularze**).
- 3 Kliknij listę prawym przyciskiem myszki.
- 4 Wybierz menu **Konfiguruj listę**.
- 5 Dodaj łącze **Domena** (Domain) do kolumn listy.
- 6 Kliknij przycisk **OK**.
- 7 Przesortuj listę w kolumnie **Domena**.
- 8 Formularze modułu identyfikuje nazwa domeny.
Przykład: **/Zakupy/Formularze/**.

5 | Raporty Crystal

ROZDZIAŁ

Rozdział ten wyjaśnia, w jaki sposób drukować raporty za pomocą AssetCenter.

Lista raportów jest dostępna w menu Narzędzia/ Przetwarzanie raportów/ Raporty.

Działanie i instalacja generatora raportów

Funkcjonowanie ogólne

AssetCenter wykorzystuje do tworzenia raportów oprogramowanie Crystal Reports. Raporty te mają rozszerzenie pliku **.rpt**.

 Uwaga:

Lista z komentarzami na temat dostępnych raportów jest dostarczana w pliku reports.txt w folderze podrzędnym datakit\standard\reports folderu instalacyjnym oprogramowania AssetCenter.

Raporty Crystal są przechowywane w folderze \datakit\standard\reports\rpt.

Nie należy modyfikować struktury pliku reports.txt, ponieważ jest on wykorzystywany przez skrypt importu. Możesz natomiast, jeżeli chcesz zaimportować jedynie niektóre raporty do bazy eksploatacyjnej, usunąć **pełne** wiersze z tego pliku przed wykonaniem skryptu importu. Możesz również dodać własne raporty.

Nie musisz posiadać programu Crystal Reports, aby drukować istniejące raporty.

Jeśli zaznaczysz odpowiednią opcję podczas instalacji, wraz z AssetCenter zainstalowana zostanie ograniczona wersja Crystal Reports.

Ta ograniczona wersja jest wystarczająca do przeglądania i drukowania istniejących raportów z bieżącymi danymi z bazy danych AssetCenter.

Musisz posiadać program Crystal Reports, aby modyfikować istniejące raporty lub tworzyć nowe raporty.

AssetCenter oczywiście nie umożliwia tworzenia tych raportów bezpośrednio.

W tym celu należy nabyć i zainstalować pełną wersję Crystal Reports. Ograniczenia, których należy przestrzegać, są opisane w poniższej tabeli:

Obsługiwane wersje Crystal Reports	Wersje 7.x et 8.0
	Wersja japońska: wersja 7.0

Obsługiwane języki Crystal Reports

- Francuski
- Angielski
- Niemiecki
- Hiszpański
- Włoski
- Japoński
- Polski

Możliwe jest stosowanie różnych języków dla Crystal Reports i AssetCenter.

Sterowniki ODBC obsługiwane przez Crystal Reports

Sterownik ODBC AssetCenter.

Instalacja pełnej wersji lub runtime Crystal Reports

Program instalacyjny oprogramowania AssetCenter stwarza możliwość zainstalowania wersji runtime Crystal Reports, wystarczającej do wydrukowania istniejących raportów. Wystarczy zaznaczyć właściwą opcję podczas instalacji oprogramowania AssetCenter.

Jeżeli została już zainstalowana pełna wersja Crystal Reports, albo masz zamiar ją zainstalować, nie ma potrzeby instalowania wersji runtime Crystal Reports przez AssetCenter .

Instalacja prekonfigurowanych raportów Crystal Reports w roboczej bazie danych.

Wraz z AssetCenter są dostarczane raporty prekonfigurowane. Są one już zainstalowane w demonstracyjnej bazie danych. Należy je natomiast dodać do bazy eksploatacyjnej.

 **Uwaga:**

Raporty prekonfigurowane nie są dostarczane z następującymi wersjami oprogramowania AssetCenter:

- Wersją japońską
- Wersją polską

W celu dodania raportów jeden po drugim:

- 1 Uruchom AssetCenter
- 2 Otwórz roboczą bazę danych
- 3 Otwórz menu **Narzędzia/ Przetwarzanie raportów/ Raporty**
- 4 Utwórz nowy raport.
- 5 Wybierz zakładkę **Plik** w szczegółach raportu.
- 6 Kliknij przycisk **Import**.
- 7 W folderze podrzędnym config folderu instalacyjnego oprogramowania AssetCenter wybierz plik o rozszerzeniu .rpt zgodnie z potrzebami.

Import raportów przy tworzeniu nowej bazy danych

Aby zaimportować raporty podczas tworzenia bazy danych:

- 1 Uruchom AssetCenter Database Administrator.
- 2 Wybierz menu **Plik/ Otwórz**.
- 3 Wybierz opcję **Otwórz plik opisowy bazy danych - utwórz nową bazę danych**.
- 4 Wybierz plik gbbase.dbb, znajdujący się w folderze podrzędnym config folderu instalacyjnego oprogramowania AssetCenter.
- 5 Wybierz menu **Działanie/ Utwórz bazę danych**.
- 6 Wypełnij wszystkie pola umożliwiające utworzenie bazy danych przy użyciu podręcznika **Administracja**, rozdział **Tworzenie bazy danych AssetCenter**.
- 7 Wybierz opcję **Import dodatkowych danych**.
- 8 W ramce **Dane do zaimportowania** wybierz **Raporty Crystal Reports**.
- 9 Kliknij przycisk **Utwórz**, aby zatwierdzić utworzenie bazy danych.

Import raportów do istniejącej bazy danych

Aby zaimportować raporty do istniejącej bazy danych:

- 1 Uruchom AssetCenter Database Administrator.
- 2 Wybierz menu **Plik/ Otwórz**.
- 3 Wybierz opcję **Otwórz plik opisowy bazy danych - utwórz nową bazę danych**.
- 4 Wybierz plik gbbase.dbb, znajdujący się w folderze podrzędnym config folderu instalacyjnego oprogramowania AssetCenter.

- 5 Wybierz menu **Działanie/ Utwórz bazę danych**.
- 6 Wybierz bazę danych w polu **Baza danych**.
- 7 Wybierz opcje **Tworzenie bazy danych i Tworzenie danych systemowych**.
- 8 Wybierz opcję **Import dodatkowych danych**.
- 9 W ramce **Dane do zaimportowania** wybierz **Raporty Crystal Reports**.
- 10 Kliknij przycisk **Utwórz**, aby zatwierdzić import raportów.

Szczegóły raportu

Użyj polecenia menu **Plik/ Raporty**, aby wyświetlić listę raportów. Szczegóły raportu w AssetCenter składają się z następujących informacji:

Plik

Nie można edytować tego pola bezpośrednio. Wskazuje ono nazwę pliku raportu (z jego rozszerzeniem oraz względną ścieżką jego folderu), który był zaimportowany przy użyciu przycisku **Importuj**.

Następujące przyciski umożliwiają pracę z raportami:

- **Importuj**: ten przycisk w szczegółach raportu pozwala zaimportować (pierwszy raz - aby utworzyć raport, później - aby modyfikować raport) raport zewnętrzny. Raporty zewnętrzne mają rozszerzenie pliku **.rpt**. Importowanie raportu zewnętrznego aktualizuje pole **Plik** (nazwa SQL: FileName) w szczegółach raportu AssetCenter.
- **Eksportuj**: ten przycisk w szczegółach raportu pozwala utworzyć plik **.rpt** z raportu zawartego w bazie danych AssetCenter. Domyślnie okno dialogowe, które zostaje otwarte, proponuje nazwę tego pliku zawartą w polu **Plik**. Można ją zmienić. Zmiana nazwy umożliwia modyfikowanie raportu przy użyciu zewnętrznego programu do tworzenia raportów.
- **Podgląd**: Przycisk ten, dostępny przez polecenie menu **Plik/ Drukuj**, pozwala wyświetlić podgląd raportu na ekranie przed jego wydrukowaniem.
- **Drukuj**: przycisk ten, dostępny przez polecenie menu **Plik/ Drukuj**, pozwala wydrukować raport.

 Uwaga:

Po naciśnięciu przycisku **Podgląd** lub **Drukuj**, AssetCenter tworzy plik tymczasowy z raportu w bazie danych. Plik ten jest przetwarzany przez motor wydruku Crystal Reports. Plik tymczasowy jest natychmiast po tym usuwany. Dane wyświetlone lub wydrukowane są danymi aktualnie przechowywanymi w bazie danych.

W jaki sposób zmodyfikować raport Seagate Crystal

Aby zmodyfikować raport zawarty w bazie danych AssetCenter, musisz posiadać program Crystal Reports.

Zastosuj następującą procedurę:

- 1 Wyświetl szczegóły raportu przy użyciu polecenia menu **Plik/Raporty**.
- 2 Kliknij przycisk **Eksportuj**, aby utworzyć plik **.rpt**.
- 3 Zmodyfikuj raport **.rpt** za pomocą Crystal Reports i zapisz go.
- 4 Wyświetl ponownie szczegóły raportu za pomocą polecenia menu **Narzędzia/Przetwarzanie raportów/Raporty**.
- 5 Zimportuj plik **.rpt**, aby go zaktualizować i zmodyfikuj rekord.

Statystyka Crystal Reports


Aby wyświetlić raporty Crystal, które są automatycznie aktualizowane, użyj polecenia menu **Narzędzia/Przetwarzanie raportów/Statystyki Crystal Reports**.

Możesz wyświetlić te same raporty przy użyciu polecenia menu **Narzędzia/Przetwarzanie raportów/Raporty**.

Właściwość


Wskaż właściwość raportu do wyświetlenia. Pole po prawej stronie tego pola umożliwia wybór danego raportu. Dostępne raporty zależą od "Właściwości", która została wybrana.

Przycisk Automatyczne odświeżanie

Przycisk jest przedstawiony w postaci ikony .

- Kliknij ten przycisk, aby odświeżyć raport.
- Kliknij ten przycisk prawym przyciskiem myszy, aby określić częstotliwość automatycznego odświeżania raportów.

Przycisk Zoom

Przycisk jest przedstawiony w postaci ikony .

Zmienia współczynnik powiększenia (3 poziomy).

Tworzenie raportu szczegółów

"Raport szczegółów" jest raportem, który drukuje informacje szczegółowe dotyczące jednego lub większej liczby rekordów wybranych na liście.

Przykład wykorzystania

- 1 Wyświetl listę zasobów
- 2 Wybierz zasób.
- 3 Wybierz polecenie menu **Plik/ Drukuj**.
- 4 Ustaw pole **Typ** jako "Raport szczegółów (Crystal Reports)".
- 5 Wybierz raport.
- 6 Drukuj.

Procedura ta drukuje raport szczegółów dla każdego wybranego rekordu.

Konfigurowanie raportów w Crystal Reports

Aby uzyskać raport szczegółów, zastosuj się do poniższej procedury (przykład wzięty z Crystal Reports Professional 5.0):

- 1 Użyj polecenia menu **Wstaw/ Pole wzoru**, aby utworzyć pole wzoru. Jego nazwa musi respektować poniższą składnię:

`<SQL name of the table for which the report is contextual>Id`

 Uwaga:

Musisz uwzględniać wielkość liter nazw SQL tabel.

Na przykład, aby utworzyć raport kontekstowy dotyczący tabeli zasobów, wzór jest następujący:

```
amAssetId
```

 Uwaga:

Nie pomył składni nazwy pola wzoru z nazwą SQL pola klucza głównego. Na przykład klucz podstawowy tabeli zasobów to "lAstId". Wzór "CurrentUserId" (respektuj wielkość liter) umożliwia identyfikację użytkownika drukującego raport. Przy drukowaniu, wzór ten przyjmuje wartość numeru identyfikatora (tj. wartość pola o nazwie SQL: "lEmplDeptId" dla bieżącej nazwy login) użytkownika połączonego z bazą danych AssetCenter.

Jeśli chcesz zobaczyć wynik raportu dla danego rekordu w tabeli kontekstowej, edytuj pole wzoru i wprowadź klucz główny tabeli dla istniejącego rekordu w bazie danych AssetCenter.

Przykład:

```
512
```

 Uwaga:

Pole wzoru edytuj w oknie, które jest automatycznie wyświetlane po potwierdzeniu nazwy nowego pola wzoru. Jeśli pole wzoru już istnieje, kliknij przycisk **Edit**, aby je edytować.

- 2 Użyj polecenia menu **Raport/ Edytuj wybrany wzór/ Rekord**, aby edytować wzór wyboru. Używana jest następująca składnia:

```
{< SQL name of the context table>.<SQL name of the field used as primary key>} = @<Formula Field name>}
```

Wielkość liter zastosowana dla nazw SQL tabel i pól nie ma znaczenia. Przykład:

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


Stosując powyższą procedurę, AssetCenter automatycznie identyfikuje raport jako kontekstowy, gdy jest on importowany do bazy danych.

Zobaczysz to, gdy wykonasz co następuje:

- 1 Otwórz listę raportów przy użyciu polecenia menu **Plik/ Raporty**.
- 2 Utwórz nowy raport.
- 3 Zaimportuj plik Crystal Reports (rozszerzenie **.rpt**) przez kliknięcie przycisku **Importuj**.
- 4 W chwili dodania tego pliku zauważysz, że pole **Tabela** (nazwa SQL: TableName) pokazuje nazwę SQL tabeli kontekstowej. Jeśli to nie nastąpi, zweryfikuj pole wzoru oraz wzór wyboru w Crystal Report.

Drukowanie raportu

Raporty szczegółowe

Uwaga:

Informacje na temat raportów szczegółowych i sposobu ich tworzenia za pomocą Crystal Reports można znaleźć w niniejszym podręczniku w rozdziale **Raporty Crystal**, podrozdział **Tworzenie raportu szczegółów**.

Aby wydrukować raport szczegółowy w AssetCenter:

- 1 Wyświetl listę rekordów, dla których chcesz wydrukować szczegóły (na przykład **Repozytorium/ Zasoby i partie**).
- 2 Zaznacz wszystkie rekordy, które zostaną objęte drukowaniem.
- 3 Wybierz menu **Plik/ Drukuj**.
- 4 Nadaj wartość **Raport szczegółów** (Crystal Reports) polu **Typ**.
- 5 Wybierz raport.
- 6 Kliknij przycisk **Drukuj**.

Procedura ta drukuje raport szczegółów dla każdego wybranego rekordu.

Raporty niekontekstowe

 Uwaga:

Raporty niekontekstowe są to raporty zawierające listy i wykresy. W przeciwieństwie do **raportów szczegółowych** nie mają one tabeli kontekstowej.

Aby wydrukować raport niekontekstowy w AssetCenter:

- 1 Wybierz menu **Plik/ Drukuj**.
- 2 Nadaj wartość **Raport niekontekstowy (Crystal Reports)** polu **Typ**.
- 3 Wybierz raport.
- 4 Kliknij przycisk **Drukuj**.

Procedura ta drukuje raport niekontekstowy dotyczący listy lub wykresu.

Identyfikowanie raportów Crystal związanych z danym modulem

Aby zidentyfikować raporty Crystal związane z danym modulem:

- 1 Uruchom AssetCenter.
- 2 Wyświetl listę raportów (**Narzędzia/Przetwarzanie raportów/Raporty**).
- 3 Kliknij listę prawym przyciskiem myszki.
- 4 Wybierz menu **Konfiguruj listę**.
- 5 Dodaj łącze **Domena [Domain]** do kolumn listy.
- 6 Kliknij przycisk **OK**.
- 7 Przesortuj listę w kolumnie **Domena**.
- 8 Raporty modułu identyfikuje nazwa domeny.

Przykład: **Zarządzanie repozytorium/ Informatyka/ Raporty**.

6 | Działania

ROZDZIAŁ

Niniejszy rozdział wyjaśnia, w jaki sposób definiować działania za pomocą AssetCenter.

Użyj polecenia menu **Narzędzia/ Działania/ Edytuj** w celu tworzenia działań.

Działania mogą być wykonywane przez polecenie menu **Narzędzia/ Działania** lub listę kontekstową "Działania" na pasku narzędzi.

Definiowanie działań

Działanie pozwala na częściowe automatyzowanie zadań wykonywanych w bazie danych AssetCenter.

Można wyróżnić kilka typów działań:

- Wykonywalny,
- DDE
- Poczta,
- Skrypt: modyfikowanie obiektu w obrębie bazy danych AssetCenter,
- Kreator,

- Drukowanie.
- Wdrożenie
- Działanie

Aby działania mogły być wykonywane za pomocą wyboru z listy działań, muszą one zostać uprzednio zdefiniowane.

 **Uwaga:**

Można określić domenę dla działania i dzięki temu uporządkować funkcje, za pomocą domen funkcjonalnych.

Domena funkcjonalna

AssetCenter umożliwia definiowanie domen, gromadzących funkcjonalności oprogramowania. Domyślnie niektóre domeny funkcjonalne są dostarczane wraz z oprogramowaniem. Odpowiadają one modułom, które można aktywować lub dezaktywować za pomocą menu Plik/ Aktywacja modułów.

Domeny funkcjonalne są używane w celu tworzenia i porządkowania informacji wyświetlanych w panelu funkcji i ulubionych. I tak, domena funkcjonalna wybrana dla działania pojawi się w panelu funkcji i ulubionych pod tytułem domeny funkcjonalnej.

 **Uwaga:**

Zawartość panelu funkcji i ulubionych jest organizowana i modyfikowana w zależności od kontekstu. Jeżeli działanie jest kontekstowe (na przykład jest wykonane, gdy ekran pracowników jest otwarty), pojawi się w panelu funkcji i ulubionych jedynie wtedy, gdy kontekst bieżący jest zgodny z jego kontekstem (na przykład tabela pracowników).

Aby zdefiniować domenę funkcjonalną:

- 1 Kliknij przycisk **Nowy**,
- 2 Wybierz menu **Administracja/ Domeny funkcjonalne**
- 3 Nadaj **Nazwę** domenie funkcjonalnej. Nazwa ta zostanie wyświetlona w panelu funkcji i ulubionych. Domyślnie AssetCenter nadaje domenie funkcjonalnej **Nazwę SQL**. Wartość tę można zmodyfikować.
- 4 Ewentualnie wybierz **Domene nadrzędną** dla domeny funkcjonalnej.

5 Zakończ tworzenie klikając przycisk **Utwórz**.

Tworzenie działania

Podrozdział ten opisuje, w jaki sposób utworzyć działanie:

- Typy działań
- Metoda ogólna
- Wypełnianie zakładki DDE
- Wypełnianie zakładki Messaging

Typy działań

AssetCenter pozwala definiować kilka typów działań.

 **Uwaga:**

AssetCenter pozwala definiować jedynie działania typu **Wykonywalny**, **Poczta** lub **Drukowanie**. Działania typu **DDE**, **Skrypt**, **Kreator** są zarezerwowane i można je jedynie wykonywać.

Działania typu Wykonywalny

Działanie typu **Wykonywalny** powoduje, że zostaje uruchomiony program.

Uruchamia ono aplikacje typu **.exe**, **.com**, **.bat**, lub **.pif**. Można także odwoływać się do dowolnego typu dokumentu pod warunkiem, że jego rozszerzenie jest skojarzone z aplikacją w menadżerze plików.

Działania typu DDE

Działanie typu **DDE** wysyła żądanie DDE do aplikacji serwera DDE (lub aplikacji zgodnej z DDE) potrafiącej obsługiwać żądania DDE.

DDE oznacza "Dynamic Data Exchange" (Dynamiczna wymiana danych); wyznacza ono metodę dla dynamicznego wymienia informacji pomiędzy programami. AssetCenter używa poleceń DDE wykonywanych przez inną aplikację.

Przykład: Przez polecenie DDE można zgłosić żądanie do programu Microsoft Word, aby otworzył plik o określonej nazwie i z podaną zawartością.

Działania typu Poczta

AssetCenter pozwala definiować i uruchamiać działania typu **Poczta**.

- Za pośrednictwem wewnętrznego systemu poczty AssetCenter.
- Za pośrednictwem zewnętrznego standardowego systemu poczty VIM (np. Lotus Notes, Lotus cc:Mail).
- Za pośrednictwem zewnętrznego standardowego systemu poczty MAPI (np. Microsoft Exchange, Microsoft Outlook).
- Za pośrednictwem zewnętrznego standardowego systemu poczty SMTP.

Ostrzeżenie:

Możesz wysyłać wiadomości jedynie za pośrednictwem tych systemów poczty, z którymi masz możliwość nawiązywania połączenia.

Aby wysłać wiadomość zgodną ze standardem VIM, MAPI lub SMTP, AssetCenter wykorzystuje:

- Pola **Konto** (nazwa SQL: MailLogin) oraz **Hasło** (nazwa SQL: MailPassword) zakładki **Poczta** szczegółów pracownika, który utworzył bazę danych AssetCenter (tabela wydziałów i pracowników), aby zidentyfikować nadawcę tej wiadomości,
- Pole **E-mail** (nazwa SQL: EMail) w zakładce **Ogólne** szczegółów pracownika, aby zidentyfikować adresata tej wiadomości.

Aby wysłać wiadomość za pośrednictwem wewnętrznego systemu poczty, AssetCenter wykorzystuje pola **Login** oraz **Hasło** w zakładce **Profil** szczegółów zarówno nadawcy, jak i adresata.

Uwaga:

Wewnętrzny adres pocztowy użytkownika AssetCenter jest taki sam, jak **Login**.

! Ostrzeżenie:

Administrator musi utworzyć użytkownika o nazwie "Admin" i wypełnić pola **Konto**, **E-mail** oraz **Hasło**, aby skorzystać z zewnętrznego systemu poczty i mieć pewność, że AssetCenter Server funkcjonuje prawidłowo.

Działania typu Skrypt

Działania typu **Skrypt** umożliwiają wykonanie dowolnej operacji w bazie danych AssetCenter. Dają one zaawansowanym użytkownikom rozległą kontrolę bazy danych, pozwalając im wykonywać operacje, które nie mogą być wykonywane przy użyciu innych typów działań, a w szczególności:

- Tworzenie rekordów
- Usuwanie rekordów
- Powielanie rekordów
- Modyfikowanie jednego lub większej liczby obiektów w bazie danych AssetCenter, np. wszystkich rekordów w tabeli, polu lub łączu.

Operacje wykonywane przez ten typ działania są opisane przez skrypt Basic; pozwala to wykorzystywać złożone funkcje podobne do występujących w interfejsach API AssetCenter.

✍ Uwaga:

Złożoność funkcji, które mogą być użyte w działaniu typu **Skrypt**, skojarzona ze zdolnością do wprowadzania głębokich zmian w bazie danych, czyni ten typ działania potencjalnie niebezpiecznym dla integralności bazy danych. Dlatego też powinien on być wykorzystywany wyłącznie przez zaawansowanych użytkowników.

Do zmiany wartości obiektu bazy danych używane są różne funkcje, zależnie od kontekstu działania:

- Jeśli działanie nie ma kontekstu, musisz stosować funkcje pochodzące z interfejsów API programu AssetCenter, takie jak **AmSetFieldStringValue()** lub **AmSetFieldLongValue()**.
- Jeśli działanie ma tabelę jako kontekst, możesz użyć funkcji **Set()**; ma ona następującą składnię:

```
Set [<Łącze.Łącze.Pole>]=<Wartość>
```

Działania typu "Kreator"

Kreatory są przeznaczone do przeprowadzania użytkownika krok po kroku przez złożone lub powtarzające się zadania w AssetCenter. Kreatory są projektowane za pośrednictwem dedykowanego języka programowania.

Uwaga:

Kreatory są działaniami złożonymi. Więcej informacji można znaleźć w niniejszym podręczniku, w rozdziale **Kreatory**.

Działania typu Drukowanie

Działanie typu **Drukowanie** pozwala na wydrukowanie raportu lub formularza.

Dla tego typu działań należy wypełnić pola:

- pole **Typ** (seFormType): pozwala na określenie typu dokumentu do wydrukowania (raportu lub formularza).
 - pole **Raport** lub **Formularz**: pozwala na określenie używanego typu raportu lub formularza.
-

Uwaga:

Kontekst działania jest określony przez kontekst zdefiniowany dla raportu lub formularza.

Działania typu wdrożenie

Działanie typu **Wdrożenie** pozwala na uruchomienie wdrożenia na komputerach podanych w tabeli komputerów.

Jest to działanie kontekstowe w tabeli komputerów

Dla tego typu działań należy wypełnić pola:

- **Workflow wdrożenia**: wybierz workflow wdrożenia, które utworzyłeś lub zaimportowałeś.
- **serwer wdrożenia**: wybierz aplikację serwera wdrożenia.

Więcej informacji na temat implementacji i konfiguracji wdrożenia znajduje się w podręczniku **Desktop Administration**.

Działania typu działanie

Działanie typu **Działanie** pozwala na wykonanie innego działania w zależności od wpisanego zapytania.

Dla tego typu działań należy wypełnić pola:

- pole **Działanie do wykonania**: pozwala na podanie działania do uruchomienia.
- pole **Zapytanie dot. zaznaczenia**: pozwala na wpisanie skryptu, który poda kontekst stosowania działania.

Metoda ogólna

Aby utworzyć działanie:

- 1 Wybierz **Narzędzia/ Działania/ Edytuj**.
- 2 Kliknij przycisk **Nowy**.
- 3 Wprowadź nazwę dla działania.
- 4 W polu **Typ** (nazwa SQL: seActionType), określ typ działania, które chcesz utworzyć. Wybrany typ działania decyduje o wyświetlaniu jednej z następujących zakładek:
 - **Wykonywalny**.
 - **DDE**.
 - **Poczta**.
 - **Skrypt**.
 - **Kreator**.
 - **Drukowanie**.
- 5 Opcjonalnie, wypełnij pole **Nazwa SQL** (nazwa SQL: SQLName) szczegółów działania. Nazwa SQL jest unikalnym sposobem, aby zidentyfikować działanie; jest ona w szczególności używana podczas wykonywania działania za pośrednictwem polecenia DDE (w przypadkach, gdzie AssetCenter jest wykorzystywany jako serwer poleceń DDE).

Uwaga:

Jeśli nie wypełnisz pola **Nazwa SQL**, AssetCenter zrobi to automatycznie poprzez wygenerowanie standardowej nazwy SQL.

- 6 Wypełnij pole **Kontekst** (nazwa SQL: ContextTable):

- Jeśli wybierzesz tabelę z listy rozwijanej, działanie jest kontekstowe: będzie ono proponowane tylko przy wyświetlaniu listy rekordów w tej tabeli, lub szczegółów jednego z tych rekordów.
 - Jeśli działanie nie zależy od tabeli, wybierz opcję (**Brak tabeli**) u góry listy rozwijanej.
- 7 Ewentualnie wypełnij pole **Domena**, które umożliwia podanie domeny funkcjonalnej, do której należy działanie. To pod tą nazwą działanie pojawia się w panelu funkcji i ulubionych.
 - 8 Opcjonalnie, dołącz ikonę do działania:
Aby to zrobić, użyj kwadratu znajdującego się u góry, po lewej stronie ekranu szczegółów działania. Obraz pojawi się następnie na liście kontekstowej "Działania" na pasku narzędzi. Aktywna ikona na tej liście (ikona domyślnie wyświetlana na ekranie) jest ikoną dla ostatniego działania wykonanego za pomocą paska narzędzi.
 - 9 Wypełnij pola w zakładce **Opis** oraz pola w zakładce specyficznej dla typu działania, które chcesz utworzyć.
 - 10 Kliknij przycisk **Utwórz**.
-

Uwaga:

Administrator AssetCenter widzi wszystkie działania, bez względu na to czy są one, czy też nie są współdzielone, a także bez względu na to, czy były one, czy też nie były utworzone przez administratora.

Wypełnianie zakładki DDE

Informacje dotyczące szczególnego działania typu DDE znajdują się w zakładce DDE szczegółów tego działania.

Zakładka ta jest wyświetlana wyłącznie, jeśli użytkownik przypisał wartość **DDE** do pola **Typ** (nazwa SQL: seActionType) w informacjach podstawowych dla działania.

Mechanizmy DDE są oparte na "usługach" dostarczanych przez oprogramowanie. Aby wykonać mechanizmy DDE, musisz zdefiniować "temat" wskazujący kontekst, w którym powinny być wykonywane "polecenia".

Dlatego też musisz wskazać:

- W polu **Usługa** (nazwa SQL: DDEService), nazwę usługi DDE dostarczanej przez plik wykonywalny, który chcesz wywołać. Zwykle jest to unikalna usługa pliku wykonywalnego. Zapoznaj się z dokumentacją pliku wykonywalnego, aby uzyskać listę usług, które on dostarcza.
- W polu **Temat** (nazwa SQL: DDETopic), kontekst, w którym działanie powinno być wykonywane.
- W polu **Polecenie** (nazwa SQL: DDECommand), polecenia, które ma wykonać zewnętrzna aplikacja.
W przypadku programu Word, polecenie może być poleceniem WordBasic lub Visual Basic.
Jeśli usługa DDE wywołanej aplikacji pozwala na to, możesz umieścić kilka poleceń obok siebie.
Musisz zastosować się do składni wymaganej przez zewnętrzną aplikację.
- Jeśli usługa nie istnieje, wskaż w polu **Plik** (nazwa SQL: ActionFile) plik używany do uruchamiania aplikacji, która uaktywnia usługę. Jest to główna aplikacja, która odpowiada na polecenia DDE.

Ważna uwaga

Polecenia wysyłane do zewnętrznej aplikacji są otoczone nawiasami kwadratowymi ("[" , "]"). Na przykład (wykorzystując program Microsoft Word):

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

- Jeśli działanie jest kontekstowe, możesz użyć zmiennych, aby odwołać się do wartości pola w bazie danych. Ponieważ zmienne te są również otoczone nawiasami kwadratowymi, AssetCenter nie potrafi samodzielnie rozróżniać poleceń i zmiennych. Musisz zidentyfikować polecenia przez użycie przed nawiasami kwadratowymi przedrostka w postaci znaku ukośnika "\". A zatem poprzedni przykład jest zapisany (w przypadku działania kontekstowego):

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

Możesz łączyć polecenia i zmienne, jak pokazano poniżej (w tym przypadku kontekst jest tabelą zasobów):

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

- Jeśli działanie nie jest kontekstowe, problem ten nie występuje. Tekst otoczony nawiasami kwadratowymi mimo to traktowany jest jako polecenia do wysłania do aplikacji zewnętrznej.

Wypełnianie zakładki Poczta

Informacje dotyczące działania typu Poczta znajdują się w zakładce Poczta szczegółów działania.

Zakładka ta jest wyświetlana wyłącznie, jeśli pole **Typ** (Nazwa SQL: seActionType) w informacjach podstawowych działania zostało ustawione jako **Poczta**.

Ostrzeżenie:

Aby system funkcjonował poprawnie, do zmiennej PATH systemu użytkownika musi być włączony folder zawierający VIM DLL (VIM32.DLL) oraz MAPI DLL (MAPI32.DLL).

Do czego służy pole Obiekt odwołania (Nazwa SQL: RefObject)?

Pole to jest używane do wybrania łącza z tabeli wybranej w polu **Kontekst**. Pole to jest używane tylko dla wiadomości wysyłanych za pośrednictwem wewnętrznego systemu poczty AssetCenter. Pozwala ono bezpośrednio uzyskać dostęp do obiektu, który zainicjował wysłanie wiadomości, poprzez proste kliknięcie przycisku **Odsyłacz do obiektu** w szczegółach wiadomości. Jeśli obiekt odwołania jest bezpośrednio rekordem inicjującym działanie, nie wypełniasz pola **Odsyłacz do obiektu** (nazwa SQL: RefObject).

W jaki sposób otrzymać potwierdzenie

Jeśli chcesz, aby nadawca wiadomości otrzymywał potwierdzenie za pośrednictwem swojej zwykłej usługi pocztowej, zaznacz pole **Potwierdzenie** (nazwa SQL: bAcknowledgment).

Potwierdzenie to zostanie wysłane na adres określony przez pole **E-mail** (nazwa SQL: EMail) w zakładce **Ogólne** pracownika, który otworzył bazę danych AssetCenter (w tabeli wydziałów i pracowników).

 **Uwaga:**

Nie można otrzymać potwierdzenia dla wiadomości wysłanej za pośrednictwem wewnętrznego systemu poczty AssetCenter, ani za pośrednictwem systemu poczty MAPI lub SMTP.

W jaki sposób określić adres

Oto różne sposoby określania adresu:

Adres w formie <Motor poczty>:<Adres poczty>

<Motor poczty> może być następujący:

- AM: aby wymusić wykorzystanie wewnętrznego systemu poczty AssetCenter.
- MAPI: aby wymusić wykorzystanie standardowego systemu poczty MAPI (np. Internet Mail, Microsoft Outlook).
- VIM: aby wymusić wykorzystanie standardowego systemu poczty VIM (np. Lotus Notes).
- SMTP: aby wymusić wykorzystanie standardowego systemu poczty SMTP (standard internetowy).

<Adres poczty> ma zwykłą formę odpowiadającą wybranemu systemowi poczty. Wewnętrzne adresy pocztowe są takie same jak nazwy "Login".

Przykłady adresów:

- AM:Admin
- MAPI:KatarzynaBiernat@taltek.com
- VIM:Katarzyna Biernat / TALTEK
- SMTP: kbiernat@taltek.com

Adres w formie <login AssetCenter>

W tym przypadku, używanym systemem poczty będzie system wskazany w polu **E-mail** (nazwa SQL: EMail) zakładki **Ogólne** szczegółów

pracownika, którego **Login** (zakładka **Profil** w szczegółach pracownika) jest określony w adresie.

Jeśli pole **E-mail** nie jest wypełnione, wiadomość jest wysyłana za pośrednictwem wewnętrznego systemu poczty.

Przykład:

- 1 Wiadomość wysłana jest do następujących nazw login AssetCenter: "Katarzyna", "Gerard" oraz "Filip".
- 2 Pola **E-mail** pokazują "MAPI:KatarzynaBiernat@taltek.com" dla "Katarzyna" oraz "VIM:Gerard Colombo / Taltek" dla "Gerard". Pole **E-mail** "Filipa" jest puste.
- 3 Jeśli nadawca ma konto MAPI, wiadomość zostanie wysłana do "Katarzyna" za pośrednictwem MAPI, a do dwóch pozostałych adresatów za pośrednictwem wewnętrznego systemu poczty AssetCenter.
- 4 Jeśli nadawca ma konto VIM, wiadomość zostanie wysłana do "Gerard" za pośrednictwem VIM, a do dwóch pozostałych adresatów za pośrednictwem wewnętrznego systemu poczty AssetCenter.

Adres ze zmiennymi kontekstowymi

Jeśli działanie jest kontekstowe, możesz zastosować zmienne pomiędzy nawiasami []. Zmienne te wywołują wartości pól w bazie danych AssetCenter.

Na przykład: aby wysłać wiadomość do użytkownika zasobu wybranego w tabeli zasobów, możesz użyć [User.Email].

Przykłady działań

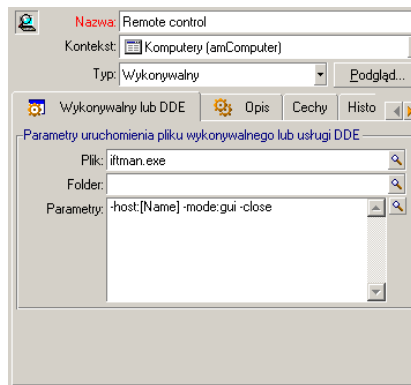
Podrozdział ten dostarcza przykłady działań AssetCenter:

- Przykład działania typu "Wykonywalne"
- Przykład działania typu "DDE"
- Przykład działania typu "Poczta"
- Przykład działania typu "Skrypt"

Przykład działania typu "Wykonywalne"

Następujący ekran opisuje działanie niekontekstowe, które uruchamia AssetCenter Server i łączy się z bazą danych **acdemo**:

Rysunek 6.1. Działanie typu "wykonywalne" - okno szczegółów



Przykład działania typu "DDE"

Istnieją liczne zastosowania działań typu **DDE**:

- Wstawianie danych AssetCenter do skoroszytu Microsoft Excel.
- Wstawianie informacji związanych ze zleceniem zakupu do pakietu oprogramowania księgowego.
- Automatyczne wysyłanie potwierdzenia zgłoszenia lub zamknięcia problemu..
- Automatyczne wysyłanie zapotrzebowania na zlecenie pracy.
- Itp.

Sekcja ta opisuje proste działanie **DDE**.

Cel działania

Działanie to wysyła potwierdzenie zapotrzebowania zakupu.

Działanie to jest inicjowane ze szczegółów zapotrzebowania zakupu.


```

in the back end
Documents.Open ("LettreType.doc") 'Open letter pattern
Documents("LettreType.doc").Activate

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

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

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

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

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

```

```

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

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

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

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

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

```

Krok 1: Utworzenie działania AssetCenter

Aby utworzyć działanie AssetCenter:

- 1 Otwórz tabelę działań (polecenie menu **Narzędzia/Działania/Edytuj**).
- 2 Kliknij przycisk **Nowy**, aby utworzyć nowe działanie.
- 3 Wypełnij pole **Kontekst** (nazwa SQL: ContextTable), aby pokazywało ono tabelę zapotrzebowań zakupu.
- 4 Ustaw pole Typ jako **DDE**.

- 5 Wypełnij zakładkę **DDE**, aby zdefiniować łącze DDE pomiędzy programami AssetCenter i Microsoft Word 7.

W zakładce DDE:

- 1 Pole **Usługa** (nazwa SQL: DDEService) jest ustawione jako "Winword".
- 2 Pole **Temat** (nazwa SQL: DDETopic) jest ustawione jako "System".
- 3 Ramka **Parametry uruchomienia usługi** pokazuje **Winword.exe** oraz ścieżkę dostępu.
- 4 Pole **Polecenie** (nazwa SQL: DDECommand) określa makro, które ma być uruchamiane, oraz jego parametry:

```
\[MyMacro.ImprimeLettreType "[Requester.MrMrs]", "[Requester.First Name]", "[Requester.Name]", "[Requester.Location.Address1]", "[Requester.Location.Address2]", "[Requester.Location.ZIP]", "[Requester.Location.City]", "[ReqNumber]"\]
```

Kliknij przycisku **Utwórz**, aby potwierdzić utworzenie działania.

Krok 2: Uruchamianie działania

Aby uruchomić działanie:

- 1 Otwórz tabelę zapotrzebowań zakupu.
- 2 Wybierz zapotrzebowanie zakupu.
- 3 Uruchom działanie przez polecenie menu **Narzędzia/ Działania**.

Po uruchomieniu działania:

- 1 Zostanie uruchomiony program Microsoft Word, który wczyta **LetterTemplate.doc**.
- 2 Do listu zostaną wstawione szczegóły osoby kontaktowej oraz numer zapotrzebowania.
- 3 List zostanie wydrukowany.

Przykład działania typu Poczta

Wysyłasz wiadomość z listy zasobów, aby podać datę upływu terminu harmonogramu leasingu użytkownikowi zasobu. Sposobem nabycia zasobu powinna być dzierżawa i powinien on być powiązany z harmonogramem leasingu (zakładka Nab.). Aby obiektem odwołania był harmonogram leasingu, skonfiguruj szczegóły działania jak następuje:

Rysunek 6.3. Działanie typu "poczta" z obiektem odwołania - okno szczegółów

The screenshot shows a configuration window for an action named 'Wygaśnięcie'. The 'Kontekst' is 'Zasoby (amAsset)' and the 'Typ' is 'Poczta'. The 'Odsyłać do obiektu' is 'AcquContract'. The 'Priorytet' is 'Normalny'. The 'Do' field contains '[PortfolioItem.User.Email]'. The 'Temat' is 'Wygaśnięcie umowy'. The 'Wiadomość' field contains 'Umowa [AcquContract.cf_self] wygasa w dniu [AcquContract.dEnd]'.

Przykład działania typu Skrypt


Utworzenie działania typu **Skrypt** w zasadzie wymaga napisania skryptu Basic, który modyfikuje bazę danych AssetCenter.

Uwaga:

Użycie funkcji specyficznych dla tych działań jest autoryzowane w tych skryptach. Indeks tych funkcji znajduje się w podręczniku AssetCenter zatytułowanym "Programmer's Reference", rozdział "Index of functions by field of application", podrozdział "Built-in functions".

Przedmowa

Aby przygotować się do utworzenia działania, zastosuj się do poniższych kroków:

- 1 Wybierz polecenie menu **Narzędzia/Działania/Edytuj** i kliknij przycisk **Nowy** na ekranie szczegółów działania.
- 2 Przypisz nazwę do działania, które zamierzasz utworzyć, np. "Test", i ustaw pole **Typ** (nazwa SQL: seActionType) jako **Skrypt**. Nie wybieraj kontekstu dla działania. Kliknij przycisk **Utwórz**.
- 3 W zakładce **Skrypt**, kliknij przycisk , aby wyświetlić okno kompilatora skryptu. Funkcja programowalna, o nazwie **Success()**, używana dla tych działań nie wymaga jawnego kodu powrotu. W następnym

przykładzie utworzymy nowy rekord w tabeli natur w oparciu o informacje zawarte w poniższej tabeli:

Etykieta pola	Nazwa SQL pola	Wartość pola
Nazwa	Name	Minikomputer
Tworzy	seBasis	Element repozytorium
Zakładka Połączenie możliwe	bIsCnxClient	Pole jest zaznaczone

Pisanie skryptu

Wprowadź następujący przykład:

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

 **Uwaga:**

Działanie to tworzy żadaną naturę bez żadnej interwencji użytkownika.

Demonstracja funkcji "Set()"

Teraz utworzymy taką samą naturę z działania typu **Skrypt**, poprzez określenie tabeli natur jako kontekstu dla działania. W tym przypadku piszemy skrypt jak następuje:

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

 **Uwaga:**

Aby wykonać to działanie, użytkownik musi utworzyć tabelę natur i nacisnąć przycisk **Nowy**. Po wykonaniu skryptu, użytkownik musi również kliknąć przycisk **Utwórz**, aby zatwierdzić utworzenie.

Wskazówka


Jeśli chcesz unieważnić wykonanie działania w skrypcie, po prostu ustaw wartość kodu powrotu jako inną niż 0 (na przykład 12001). Wartość ta jest traktowana jako kod błędu. Następne polecenie przerywa działanie i anuluje wszystkie już dokonane zmiany:

```
RetVal=12001
```

Używanie zmiennych

W zakładkach **Wykonywalny**, **DDE**, lub **Poczta** szczegółów działania kontekstowego, można stosować zmienne, które odwołują się do zawartości pól, cech lub pól wyliczanych w bazie danych.

Używają one formy **[Łącze.Łącze.Pole]**.

Aby uzyskać pomoc przy wprowadzaniu tych zmiennych, kliknij lupę  po prawej stronie pola, które ma być wypełnione.

Wszystko, co nie jest zawarte wewnątrz nawiasów klamrowych [] jest traktowane jako tekst.

Na przykład: **[Łącze.Łącze.Pole].doc**, wywołuje wartość pola **Pole** w tabeli połączonej z główną tabelą przechodząc przez łącza **Łącze.Łącze**.

Ostrzeżenie:

Aby zmienne działały, pole **Kontekst** szczegółów działania musi pokazywać tabelę w AssetCenter, a użytkownik przed wykonaniem działania musi wybrać rekord na liście rekordów tej tabeli.

Specyfika motoru bazy danych Sybase SQL Anywhere

W przypadku korzystania z systemu Sybase SQL Anywhere jako motoru bazy danych AssetCenter, nie można zapisać "{d" ani "m" na początku pola w działaniu.

Jeśli pole w działaniu musi rozpoczynać się od "{d" lub "m", zalecane jest poprzedzanie tych ciągów spacją.

Testowanie działania

Aby przetestować działanie przy jego tworzeniu, użyj przycisku **Podgląd** znajdującego się w prawym górnym rogu szczegółów działania, które ma być przetestowane.

Przycisk Oblicz





Po wybraniu kontekstu, kliknij przycisk **Oblicz**. Wypełnia on pola w zakładkach **Wykonywalny**, **DDE**, lub **Poczta**. Sprawdź, czy zostały prawidłowo pobrane zmienne z rekordu wybranego w polu **Kontekst** (ContextTable).

Przycisk Wykonaj

Przycisk ten pozwala wykonać działanie bezpośrednio z tego ekranu.

Wykonywanie działania

Działania można uruchamiać na kilka sposobów:

- Za pomocą menu kontekstowego  dostępnego na pasku narzędzi:
 - Przycisk  zastąpiony jest przez ikonę określającą ostatnio wykonane na danej stacji roboczej działanie, o ile ikona taka istnieje. Jeśli jakieś działanie było już wcześniej wykonywane, można kliknąć ikonę  lub zastępującą ją ikonę w celu ponowienia działania.
 - Przycisk  wyświetla listę dostępnych działań.
 - Aby dołączyć listę kontekstową do paska narzędzi, należy zastosować polecenie menu Narzędzia/ Dostosuj pasek narzędzi, stanowiącą część kategorii "Narzędzia".
- Za pomocą polecenia menu Narzędzia/ Działania: kliknij żądane działanie spośród dostępnych w menu podrzędnym.
- Za pomocą przycisku **Podgląd** znajdującego się w prawym górnym rogu szczegółowego opisu działania.

- Jeśli dane działanie jest kontekstowe, należy określić jego Kontekst (nazwa SQL: ContextTable) wybierając rekord w tabeli odniesienia dla danego działania.
- Kliknij przycisk **Wykonaj**, aby wykonać działanie.
- Z menu kontekstowego (dostępnego poprzez kliknięcie prawym przyciskiem myszki). Jeśli dla otwartej aktualnie tabeli dostępne jest co najmniej jedno działanie, w menu kontekstowym pojawia się pozycja Działania.

Zaznaczenie wielokrotne w listach

W obrębie listy zaznaczyć można szereg rekordów, a następnie zastosować działanie w stosunku do nich wszystkich.

Można w ten sposób na przykład zaznaczyć kilka zasobów i wysłać komunikat do ich użytkowników.

Działania typu "Kreator"

Kreatory złożone są z szeregu stron. Każda ze stron wyświetla informacje lub wymaga wprowadzenia informacji przez użytkownika, na przykład dokonania wyboru lub wpisania danych.

Poruszanie się pomiędzy poszczególnymi stronami kreatora jest proste:

- Po odpowiednim wypełnieniu strony można przejść do następnej strony (w zależności od możliwości dokonania takiego przejścia), klikając przycisk **Dalej**. Przycisk ten nie jest dostępny w przypadku ostatniej strony.
- Zawsze można powrócić do wcześniejszych stron, aby wprowadzić ewentualne poprawki, klikając przycisk **Cofnij**.
- Można także w każdej chwili wykonać działanie końcowe kreatora klikając przycisk **Zakończ**. Jeśli kreator nie posiada informacji wystarczających do wykonania działania, zostanie wyświetlona odpowiednia strona.

Można również anulować pracę kreatora (a co za tym idzie, związane z nim działanie) poprzez kliknięcie przycisku **Anuluj**.

7 | Poczta

ROZDZIAŁ

AssetCenter daje możliwość zarządzania dwoma typami wiadomości:

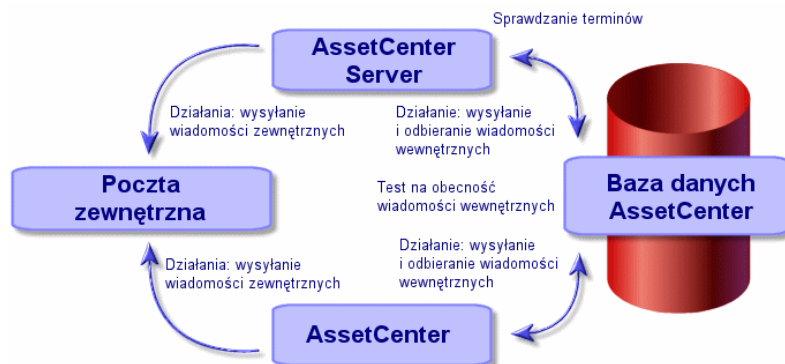
- Wiadomościami wysłanymi z AssetCenter do bazy danych AssetCenter za pośrednictwem wewnętrznego systemu poczty.
- Wiadomościami utworzonymi w AssetCenter i wysłanymi za pośrednictwem zewnętrznego systemu poczty.

Omówienie poczty

AssetCenter zarządza wysyłaniem wiadomości stosujących protokoły typu:

- AM (AssetCenter)
- SMTP
- MAPI
- VIM

Przy pobieraniu AssetCenter zarządza jedynie wiadomościami typu AM (AssetCenter)

Rysunek 7.1. Poczta - funkcjonowanie ogólne

W jaki sposób wysyłać wiadomości

Wiadomości są generowane za pośrednictwem działań typu **Poczta**.

Należy je utworzyć przed wysłaniem wiadomości.

Działanie jest inicjowane na różne sposoby:

- Ręcznie, poprzez wybór działania na liście podanej przez polecenie menu **Narzędzia/ Działania**.
- Automatycznie, za pośrednictwem programu AssetCenter Server.
- Automatycznie przez AssetCenter.

Tworzenie działań typu Poczta jest opisane w niniejszym podręczniku w rozdziale **Działania**, podrozdział **Tworzenie działania**, część **Wypełnianie zakładki Poczta**.

W jaki sposób sprawdzać wiadomości

Sprawdzanie wiadomości wysłanych do wewnętrznego systemu poczty

Agent sprawdza nadejście nowych wiadomości wewnętrznych i informuje użytkowników AssetCenter o obecności nowych wiadomości.

Wiadomości te mogą być sprawdzane:

- Przy użyciu polecenia menu **Narzędzia/ Wiadomości**.
- Z okien komunikatów, które informują o obecności nowych wiadomości.

Sprawdzanie wiadomości wysłanych do zewnętrznego systemu poczty

Adresat tych wiadomości może je sprawdzać w zwykły sposób przy użyciu klienta zewnętrznego systemu poczty.

Potwierdzenie przyjęcia

Uwaga:

Nie można otrzymać potwierdzenia dla wiadomości wysłanej za pośrednictwem wewnętrznego systemu poczty AssetCenter, ani za pośrednictwem systemu poczty MAPI lub SMTP.

Aby otrzymywać potwierdzenie, zaznacz pole wyboru **Potwierdzenie odbioru** (nazwa SQL: bAcknowledgment) w szczegółach działania typu **Poczta**.

Potwierdzenie odbioru zostanie wysłane na adres określony przez pole **E-mail** (nazwa SQL: EMail) w zakładce **Ogólne** pracownika, który otworzył bazę danych AssetCenter (w tabeli wydziałów i pracowników).

8 Workflow

ROZDZIAŁ

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

Definicje dotyczące workflow

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.
- itp.

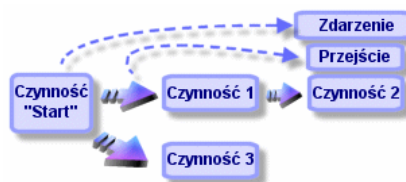
AssetCenter pozwala definiować schematy workflow i zarządzać ich przebiegiem.

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.

Rysunek 8.1. Workflow - schemat uproszczony



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 AssetCenter Server.
- 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.

Omówienie

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 AssetCenter Server lub agentów AssetCenter.

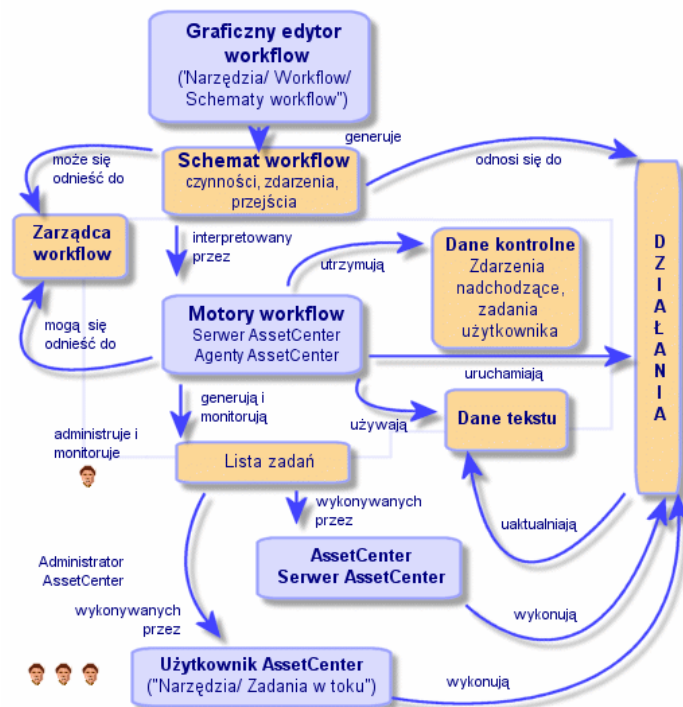
W reakcji na zdarzenia, motory workflow inicjują i monitorują instancje workflow:

- 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:

Rysunek 8.2. Workflow AssetCenter - widok ogólny

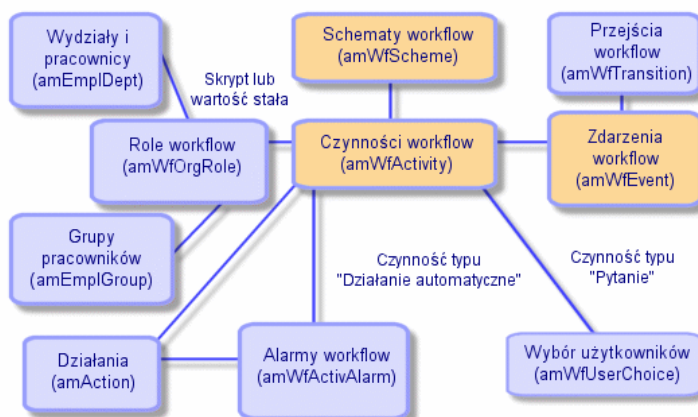


Przetwarzanie instancji workflow różni się w zależności od właściwości czynności i zdarzeń oraz sposobu w jaki skonfigurowano sposób ich kontrolowania przez motory workflow.

Główne tabele związane z workflow

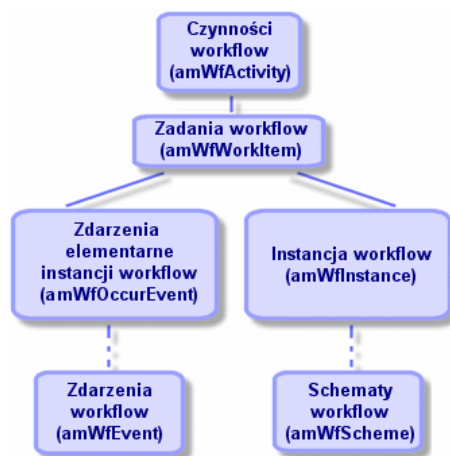
Na poniższym diagramie przedstawiono główne tabele związane z workflow i łączy pomiędzy nimi. Tabele są oznaczane przez etykietę i nazwę SQL:

Rysunek 8.3. Workflow - główne tabele umożliwiające zdefiniowanie schematu



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

Rysunek 8.4. Workflow - główne tabele związane z instancją 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 naciskając lewy przycisk myszki, a następnie naciśnij przycisk "Delete".
- Albo wybierz zdarzenie i wybierz polecenie **Usuń** z menu kontekstowego.
- 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.

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

- 1 Kliknij zdarzenie początkowe, aby wybrać przejście.
- 2 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:


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

Aby zmodyfikować źródło i/ lub cel przejścia:

- 1 Wybierz przejście.
- 2 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 AssetCenter umożliwia zautomatyzowanie procedur stosowanych w firmie. W tym celu należy:

- 1 Analiza procedur przedsiębiorstwa, które mają ulec formalizacji.

- 2 Utworzenie:
 - 1 Roli workflow.
 - 2 Działań.
- 3 Utworzenie schematów workflow, dla których należy zdefiniować:
 - 1 Czynności, zdarzenia i przejścia.
 - 2 Alarmy
- 4 Definiowanie odpowiednich grup wykonawczych workflow.
Powiązanie każdego schematu workflow z grupą wykonawczą workflow.
- 5 Uruchomienie programu AssetCenter Server na jednej lub kilku stacjach roboczych. Dla każdej instancji AssetCenter Server, zdefiniowanie grup wykonawczych workflow, które będą monitorowane przez AssetCenter Server oraz parametrów monitorowania.

Przykład zastosowania workflow w akceptacji zapotrzebowania

W tej sekcji podano prosty przykład workflow.

- 1 Cel
- 2 Wymagania wstępne
- 3 Tworzenie czynności
- 4 Ustawianie parametrów dla zdarzeń utworzonych równocześnie z czynnościami
- 5 Tworzenie zdarzenia początkowego
- 6 Tworzenie przejść
- 7 Przykład aktywowania instancji workflow

Cel

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

Rysunek 8.5. Workflow - model zatwierdzania zapotrzebowań



Etapy schematu workflow są następujące:

- 1 Instancja workflow uruchamia się, za każdym razem, gdy istnieje zapotrzebowanie zakupu do zatwierdzenia, to znaczy gdy pole **St. zapotrz.** (nazwa SQL: seStatus) w szczegółach zlecenia zakupu ma wartość **W oczekiwaniu na zatwierdzenie**.
- 2 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.
- 3 Jeśli osoba odpowiedzialna zaakceptuje zapotrzebowanie pod względem technicznym, następnym etapem jest akceptacja finansowa. W przeciwnym wypadku zapotrzebowanie jest odrzucane (3b).
- 4 Akceptacja finansowa polega na przedstawieniu zapotrzebowania kontrolerowi finansowemu firmy, Gerardowi Colombo. Jest on również informowany za pomocą poczty i programowany jest alarm przypominający.
- 5 Jeśli kontroler finansowy zaakceptuje zapotrzebowanie zakupu, jest ono zatwierdzone. W przeciwnym wypadku zapotrzebowanie jest odrzucane (5b).
- 6 Jeśli zapotrzebowanie zakupu zostało zatwierdzone, AssetCenter ustawia pole **St. zapotrz.** w szczegółach zapotrzebowania zakupu jako **Zatwierdzone**.

Jeśli zapotrzebowanie zakupu zostało odrzucone, AssetCenter ustawia pole **St. zapotrz.** (nazwa SQL: seStatus) w szczegółach zlecenia zakupu jako **Odrzucone**.

Wymagania wstępne

Należy połączyć się z bazą danych za pomocą nazwy logowania **Admin** i skonfigurować pocztę (patrz rozdział **Poczta** niniejszego podręcznika).

Tworzenie osób przypisanych do workflow

Osoby przypisane do tego schematu workflow to:

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

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 przycisk **Nowy**.

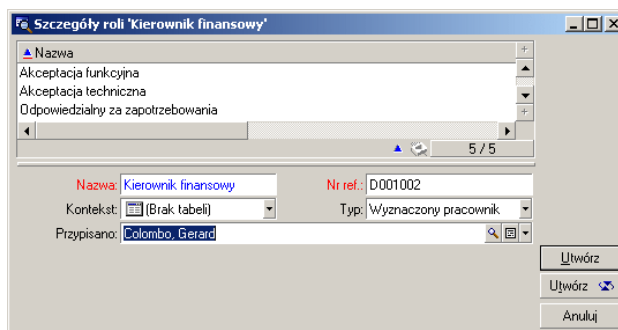
Kierownik wydziału osoby zgłaszającej zapotrzebowanie

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

The screenshot shows a software interface for defining a role. The title bar reads "Szczegóły roli 'Odpowiedzialny za zapotrzebowania'". A table lists roles with columns for "Nazwa" and "Kontekst". The selected row is "Odpowiedzialny za zapotrzebowania" with context "amRequest". Below the table, there are input fields: "Nazwa" (Odpowiedzialny za zapotrzebowania), "Nr ref.:" (D001001), "Kontekst:" (Zapotrzebowania (amRequest)), and "Typ:" (Pracownik wyliczony). A "Skrypt:" field contains the script "RetVal=[Requester.Parent.Supervisor]". On the right side, there are buttons for "Nowy", "Powiel", "Usuń", and "Zamknij".

Kontroler finansowy

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



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

- Nazwa:** Kierownik finansowy
- Nr ref.:** D001002
- Kontekst:** (Brak tabeli)
- Typ:** Wyznaczony pracownik
- Przypisano:** Colombo, Gerard

At the bottom right, there are three buttons: "Utwórz", "Utwórz" (with a refresh icon), and "Anuluj".

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 web form for 'Prośba o zatwierdzenie funkcjonalne'. The form includes the following fields and options:

- Nazwa:** Prośba o zatwierdzenie funkcjonalne
- Kontekst:** Zadania workflow (amW/WorkflowItem)
- Typ:** Poczta
- Buttons:** Poczta, Opis, Cechy, Historia, Dokumenty
- Odsyłać do obiektu:** (empty dropdown)
- Priorytet:** Normalny
- Potwierdzenie odbioru:** (checkbox, unchecked)
- Do:** [Assignee.E Mail]
- DW:** (empty)
- Udwr:** (empty)
- Temat:** Zatwierdzenie zapotrzebowań
- Wiadomość:** Zatwierdź lub odrzuć zapotrzebowanie zakupu.

Przesłanie zapotrzebowania do akceptacji finansowej do kontrolera finansowego

Działanie to jest używane w procesie akceptacji finansowej. Powoduje ono wysłanie wiadomości do osoby odpowiedzialnej za akceptację finansową o potrzebie sprawdzenia zapotrzebowania:

The screenshot shows a web form for 'Prośba o zatwierdzenie finansowe'. The form includes the following fields and options:

- Nazwa:** Prośba o zatwierdzenie finansowe
- Kontekst:** Zadania workflow (amW/WorkflowItem)
- Typ:** Poczta
- Buttons:** Poczta, Opis, Cechy, Historia, Dokumenty
- Odsyłać do obiektu:** (empty dropdown)
- Priorytet:** Normalny
- Potwierdzenie odbioru:** (checkbox, unchecked)
- Do:** [Assignee.E Mail]
- DW:** (empty)
- Udwr:** (empty)
- Temat:** Zatwierdzenie zapotrzebowań
- Wiadomość:** Zatwierdź lub odrzuć zapotrzebowanie zakupu.

Akceptacja zlecenia zakupu

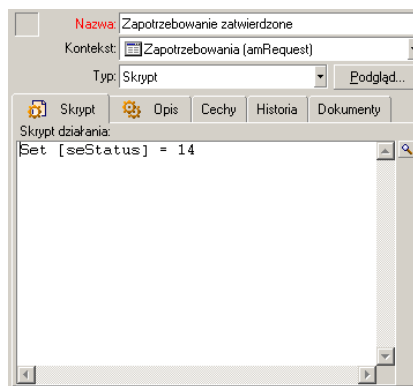
To działanie jest używane na poziomie czynności **Zapotrzebowanie zatwierdzone**, które zostanie utworzone później.

Ustawia ono pole **St. zapotrz.** (nazwa SQL: seStatus) w szczegółach zapotrzebowania zakupu jako **Zatwierdzone**. Jest to działanie typu **Skrypt**.

Pole **St. zapotrz.** jest spisem systemowym. Aby zobaczyć listę jego wartości, wyświetl pomoc dla tego pola:

- 1 Kliknij prawym przyciskiem myszy pole **St. zapotrz.** w szczegółach zapotrzebowania.
- 2 Wybierz polecenie menu **Pomoc dla tego pola** z menu kontekstowego: Wartość wyświetlana jako **Zatwierdzone** jest przechowywana w bazie danych jako **14**.

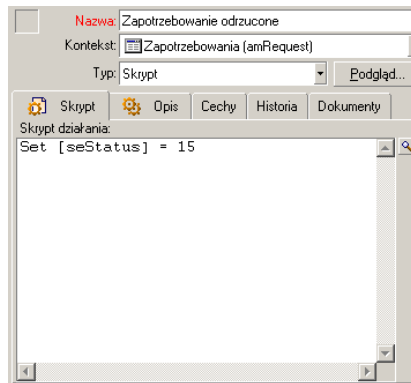
Działanie przedstawia się następująco:



Odrzucenie zapotrzebowania zakupu

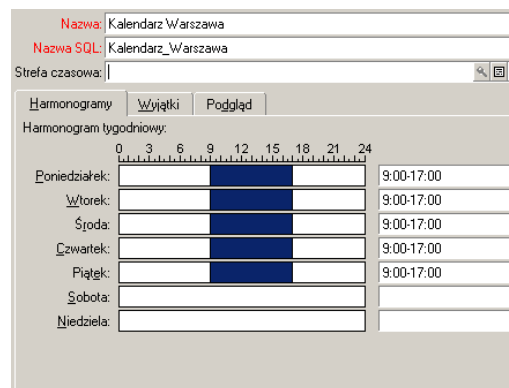
Działanie to jest używane na poziomie czynności **Zapotrzebowanie odrzucone**, która zostanie utworzona.

Jest ono podobne do działania **Zapotrzebowanie zatwierdzone**, z tym, że pole **St. zapotrz.** (nazwa SQL: seStatus) w zakładce **Ogólne** szczegółów zapotrzebowania musi mieć wartość **Odrzucone**.



Tworzenie kalendarza

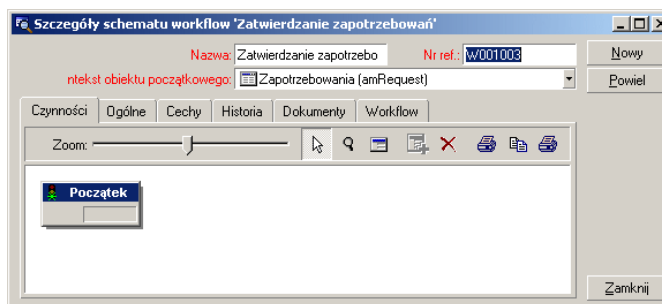
Użyj polecenia menu **Narzędzia/Kalendarze**, aby uzyskać dostęp do listy kalendarzy. Kalendarz jest związany z czynnościami schematu workflow. Umożliwia on określanie terminów końcowych dla instancji schematu workflow:



Przygotowanie schematu workflow

- 1 Uruchom polecenie menu **Narzędzia/Workflow/Schematy workflow**.

- 2 Kliknij przycisk **Nowy**.
- 3 Nadaj nazwę **Zatwierdzanie zlecenia zakupu** dla schematu workflow.
- 4 Wskaż kontekst obiektu początkowego, który będzie odnosił się domyślnie do wszystkich czynności schematu workflow. W naszym przypadku, jest to tabela zapotrzebowań (nazwa SQL: amRequest).
- 5 Kliknij przycisk **Utwórz**: czynność początkowa (**Start**) jest tworzona automatycznie przez AssetCenter w edytorze graficznym w zakładce **Czynności**.



Tworzenie czynności

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

- 1 Umieść kursor poza obszarem obiektu workflow.
- 2 Naciśnij prawy przycisk myszy.
- 3 Wybierz polecenie menu **Dodaj czynność** z menu kontekstowego: Zostaną wyświetlone szczegóły czynności.

Utwórz czynność Akceptacja funkcyjna

- 1 Nadaj nazwę **Akceptacja funkcyjna** czynności.
- 2 Ponieważ czynność polega na przekazaniu zapotrzebowania do kierownika wydziału do akceptacji, wybierz wartość **Pytanie** ze spisu **Typ** (nazwa SQL: seType).
- 3 Pole **Kontekst** (nazwa SQL: ContextTable) w zakładce **Ogólne** nie jest modyfikowane.

Ustaw parametry czynności Akceptacja funkcyjna

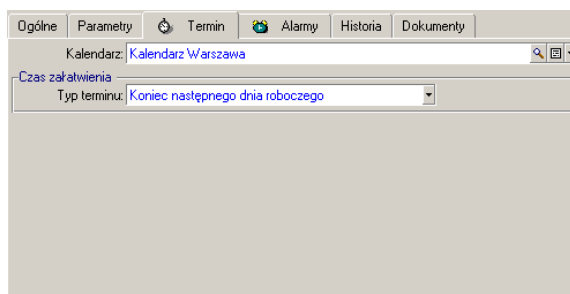
- 1 Wypełnij zakładkę **Parametry** w sposób pokazany poniżej:

- 2 Podaj pytanie, które będzie zadane:
 - 1 Tekst pytania odwołuje się do numeru zapotrzebowania zakupu.
 - 2 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 łącze** lub **Powiel łącze**.
- 3 W polu **Przypisano** (nazwa SQL: Assignee) wskaż, do kogo pytanie jest kierowane. W naszym przypadku osoba przypisana do workflow to kierownik wydziału osoby składającej zapotrzebowanie. Osoba ta została utworzona w tabeli ról workflow podczas czynności wstępnych.
- 4 Aby osoba przypisana do workflow została poinformowana o potrzebie rozpatrzenia zapotrzebowania:
 - 1 Ustaw wartość pola **Zawiadom osobę** (nazwa SQL: bNotifAssignee) jako **Tak**.
 - 2 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.

Definiowanie limitu czasowego dla czynność Akceptacja funkcyjna

W zakładce **Limit czasu** szczegółów czynności:

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



Definiowanie alarmu dla czynności Akceptacja funkcyjna

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 funkcyjnej**:

Można zdefiniować kolejne alarmy za pomocą polecenia **Dodaj łącze** 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).

Następujące czynności są inicjowane bądź bezpośrednio przez AssetCenter, bądź przez AssetCenter Server, w zależności od konfiguracji zdarzeń workflow:

- Jeżeli pole **Przetwarzanie** (nazwa SQL: seProcessingMode) ma wartość **Rejestracja zdarzenia i przetwarzanie natychmiastowe** lub **Przetwarzanie natychmiastowe bez rejestracji**, AssetCenter inicjuje następną czynność.
- Jeżeli pole **Przetwarzanie** (nazwa SQL: seProcessingMode) ma wartość **Rejestracja zdarzenia i przetwarzanie przez serwer**, AssetCenter Server inicjuje następne zdarzenie.

Domyślnie pole **Przetwarzanie** (nazwa SQL: seProcessingMode) w zakładce **Ogólne** szczegółów zdarzeń ma wartość **Rejestracja zdarzenia i przetwarzanie przez serwer**.

Tworzenie czynności Akceptacja finansowa

Czynność ta jest podobna do poprzedniej.

Inna jest osoba przypisana do workflow: w tym przypadku, jest to kontroler finansowy firmy, Gerard Colombo (osoba wyznaczona). Został on utworzony w tabeli ról workflow podczas poprzedniej czynności. Aby go zawiadomić, wybierz działanie **Zapotrzebowanie akceptacji finansowej**.

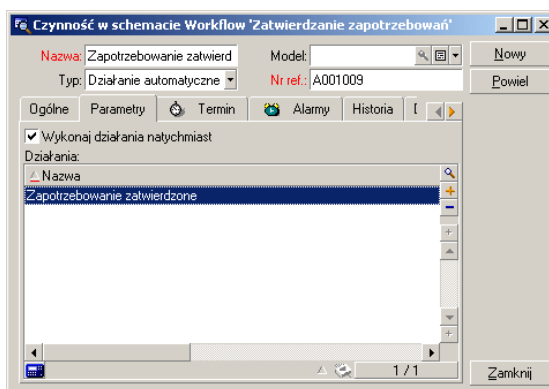
Tworzenie czynności Zapotrzebowanie zatwierdzone

Jeśli zapotrzebowanie pozytywnie przejdzie oba etapy sprawdzenia, jest ono zatwierdzone.

Czynność **Zapotrzebowanie zatwierdzone** jest jednym z możliwych zakończeń schematu workflow.

Czynność ta musi modyfikować szczegóły zapotrzebowania, aby wskazać, że zapotrzebowanie zostało zatwierdzone.

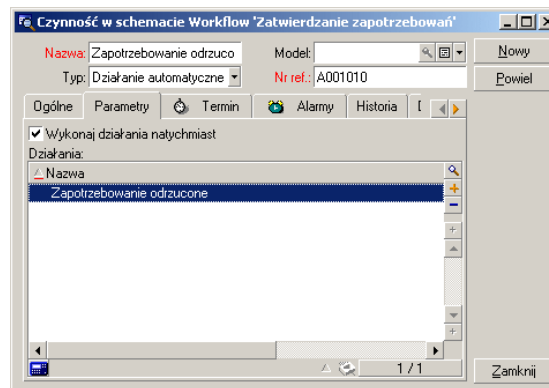
Czynność ta jest typu **Działanie automatyczne**; Działanie do wykonania to **Zapotrzebowanie zatwierdzone** utworzone na początku.



Tworzenie czynności 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.



Ustawianie parametrów dla zdarzeń utworzonych równocześnie z czynnościami

Równocześnie z czynnościami utworzonymi w poprzednich etapach zostały utworzone następujące zdarzenia:

- Czynność **Akceptacja funkcyjna**:
 - Zdarzenie **Akceptacja**
 - Zdarzenie **Odrzucenie**
- Czynność **Akceptacja finansowa**:
 - Zdarzenie **Akceptacja**
 - Zdarzenie **Odrzucenie**
- Czynność **Zapotrzebowanie zatwierdzone**:
 - Zdarzenie **Wykonane**
- Czynność **Zapotrzebowanie odrzucone**:
 - Zdarzenie **Wykonane**

Aby uniezależnić się od aplikacji AssetCenter Server, wybierz jedno zdarzenie po drugim i wykonaj następujące operacje:

- 1 Kliknij prawym przyciskiem myszki nazwę zdarzenia
- 2 Wybierz menu **Szczegóły zdarzenia**
- 3 Wybierz zakładkę **Ogólne**
- 4 W polu **Przetwarzanie** (nazwa SQL: seProcessingMode) wybierz wartość **Rejestracja zdarzenia i przetwarzanie natychmiastowe**
- 5 Kliknij przycisk **Modyfikuj**

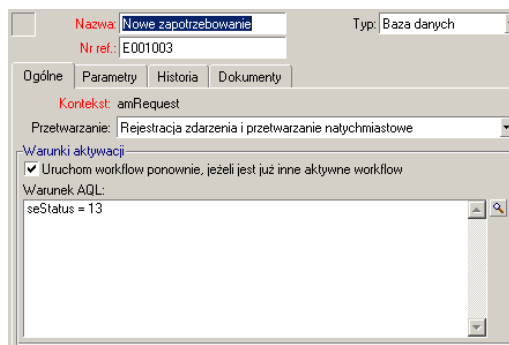
Tworzenie zdarzenia początkowego

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**.

- 1 W naszym przypadku, instancja workflow jest inicjowana, gdy pole **St. zapotrz.** (nazwa SQL: seStatus) w szczegółach zapotrzebowania ma wartość **W oczekiwaniu na zatwierdzenie**.

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



- 2 Ustaw wartość pola **Przetwarz.** (nazwa SQL: seProcessingMode) w zakładce **Ogólne** zdarzenia jako **Rejestracja zdarzenia i przetwarzanie**.

- 3 W zakładce **Parametry** zdarzenia:
 - 1 Zaznacz pole **Aktualizuj** (nazwa SQL: bUpdate).
 - 2 Wpisz w polu **Pola monitorowane** (Nazwa SQL: MonitFields) zawartość pola o nazwie SQL **seStatus**.

Tworzenie przejść

Po utworzeniu czynności, trzeba je połączyć za pomocą przejść.

Aby utworzyć przejście:

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

W tym przypadku należy utworzyć następujące przejścia:

- Od zdarzenia początkowego do czynności **Akceptacja techniczna**.
- Od zdarzenia **Zatwierdź** 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:

Rysunek 8.6. Workflow - schemat zatwierdzania zapotrzebowań



Przykład aktywowania instancji workflow

Teraz trzeba sprawdzić, czy schemat workflow działa prawidłowo.

Aby to zrobić, wykonaj następujące czynności:

- 1 Wypełnianie tabeli wydziałów i pracowników
- 2 Tworzenie zapotrzebowania zakupu do zatwierdzenia
- 3 Kontrolowanie instancji

Wypełnianie 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:

- 1 Utworzyć zgłaszającego zapotrzebowanie o nazwisku **Jerzy Stolorz**, należącego do wydziału **Informatyka**.
- 2 Przypisać **Login Stolorz**, hasło i profil użytkownika umożliwiające wprowadzenie zlecenia zakupu (zakładka **Profil** w szczegółach właściwego rekordu). Można wykorzystać profil **Zgłaszający** demonstracyjnej bazy danych.
- 3 W polu **E-mail** (nazwa SQL: EMail) wpisz wartość **AM: Stolorz**.
- 4 Kierownik wydziału **Informatyka** nazywa się **Filip Czajka**.
- 5 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). Podaj **Login** (nazwa SQL: UserLogin) oraz hasło Filipa Czajki.
- 6 W polu **E-mail** (nazwa SQL: EMail) wpisz wartość **AM: Czajka**.
- 7 Wybierz pracownika **Geralda Colombo**.
- 8 W polu **E-mail** (nazwa SQL: EMail) wpisz wartość **AM: Colombo**, a w polu **Login** (nazwa SQL: UserLogin) - **Colombo**.

Tworzenie zapotrzebowania zakupu do zatwierdzenia

Tworzenie zapotrzebowania zakupu do zatwierdzenia:

- 1 Nawiąż połączenie z demonstracyjną bazą danych używając loginu Jerzy Stolorz.
- 2 Uruchom polecenie menu **Zakupy/ Zapotrzebowania zakupu** AssetCenter.
- 3 Kliknij przycisk **Nowy**.

- 4 Wybierz opcję **Utwórz nowe dostosowane zapotrzebowanie**.
- 5 Kliknij przycisk **OK**.
- 6 W polu **Zlecający** (nazwa SQL: Requester) w zakładce **Ogólne** szczegółów zapotrzebowania, wybierz rekord **Stolorz**.
- 7 Ustaw pole **St. zapotrz.** (nazwa SQL: seStatus) w szczegółach zlecenia zakupu jako **W oczekiwaniu na zatwierdzenie**.
- 8 Potwierdź zapotrzebowanie zakupu: Następuje zdarzenie początkowe, a AssetCenter rejestruje zdarzenie w tabeli zdarzeń elementarnych workflow (nazwa SQL: amWfOccurEvent).

Kontrolowanie instancji

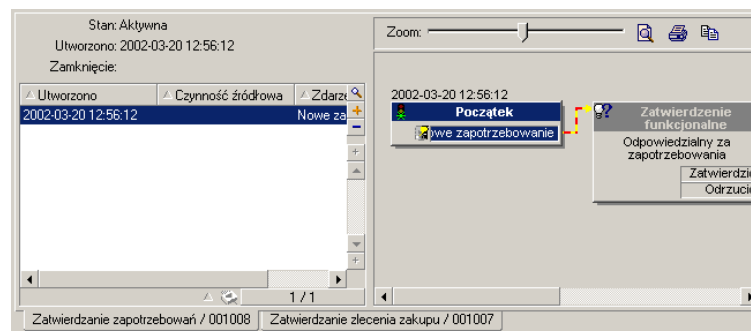
Aby sprawdzić, czy instancja workflow działa prawidłowo, uruchom AssetCenter i nawiąż połączenie z demonstracyjną bazą danych używając loginu Philip Chavez, kierownika wydziału Jérôme'a Carpentera.

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**.



Podgląd zadania do wykonania

- 1 Wybierz polecenie menu **Narzędzia/ Zadania w toku**: kierownik wydziału Filip Czajka może w ten sposób przeglądać szczegóły przypisanego zadania:

W zakładce **Ogólne** wyświetlane jest pytanie zdefiniowane w zakładce **Pytanie** czynności **Zatwierdzenie techniczne**.

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 funkcyjna**, oraz daty utworzenia zadania (tzn. daty aktywacji przejścia). Możesz kliknąć przycisk **Szczegóły**, aby uzyskać dostęp do szczegółów zapotrzebowania tworzącego zadanie.

- 2 Kliknij przycisk **Zatwierdź** lub **Odrzuć**, aby wykonać zdanie. Można również wprowadzić komentarz dotyczący decyzji w zakładce **Ogólne**.

Kontekst workflow

Każda instancja workflow ma swój specyficzny kontekst.

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 Pojedyncza aktywna instancja workflow dla obiektu w zakładce Ogólne schematu workflow.	
		Zatwierdzona	Nie zatwierdzona
Opcja Ponownie zainicjuj instancję workflow jeżeli jest w toku w zakładce Ogólne 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 **Pojedyncza aktywna instancja workflow dla obiektu**, aby dane zapotrzebowanie zakupu podlegało jednemu procesowi akceptacji.

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

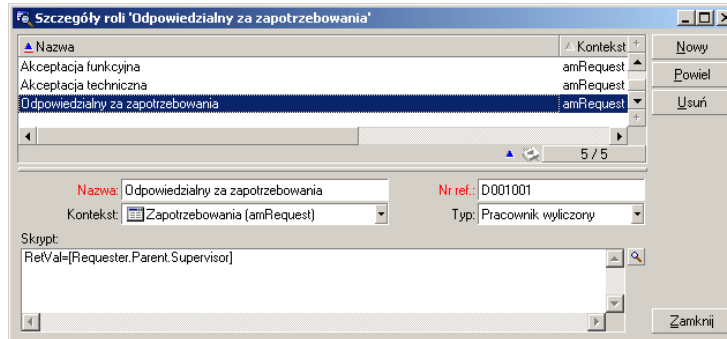
W tym przypadku, osoba przypisana wybierana jest bezpośrednio z tabeli wydziałów i pracowników.

Przykład:

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 workflow

Czynności można podzielić na dwie grupy:

- Wymagające interakcji użytkownika AssetCenter: 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.
- Pytania lub instrukcje.

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:

- 1 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.

- 2 Tekst pytania lub instrukcji do wykonania.
 - 3 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 łącze**, **Powiel łącze** 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 przypisaną 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 AssetCenter Server.

Opis

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

Przykład: w przypadku zmiany lokalizacji zasobów, czynność typu **Działanie automatyczne** automatycznie modyfikuje lokalizację wszystkich zasobów, których zasoby nadrzędne 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

Motor workflow, który aktywuje przejście inicjujące czynność, automatycznie wykonuje działania należące do czynności. W zależności od wybranego trybu przetwarzania dla zdarzenia, które inicjuje przejście działania wykonuje albo AssetCenter Server, albo agent AssetCenter.

- Jeśli zaznaczysz opcję **Wykonaj działania natychmiast** (nazwa SQL: bExecImmediately), motor workflow, który aktywuje przejście inicjujące czynność, natychmiast wykonuje działania należące do czynności.

- W przeciwnym wypadku zadania są wykonywane przez AssetCenter Server podczas kolejnej weryfikacji.

Czynności typu Test/ Skrypt

Czynności te są wykonywane automatycznie przez AssetCenter lub AssetCenter Server.

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 elementy wymienione w liniach zapotrzebowania zakupu są dostępne w zapasie i nie zarezerwowane. Jeśli tak, czynność może zainicjować czynność typu **Pytanie**, która pyta osoby zgłaszające zapotrzebowanie, czy chcą zarezerwować element 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 łącze**, **Powiel łącze** lub **Usuń łącze** z menu kontekstowego.

 Uwaga:

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

 Uwaga:

Każdy wynik automatycznie tworzy zdarzenie wynikowe czynności.

Wykonanie

Motor workflow, który aktywuje przejście inicjujące czynność, automatycznie wykonuje działania należące do czynności. W zależności

od wybranego trybu przetwarzania dla zdarzenia, które inicjuje przejście działania wykonuje albo AssetCenter Server, albo agent AssetCenter.

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

Czynność Początek

Czynność **Start** 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 **Start** 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.



Uwaga:

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 AssetCenter Server), aby czynność została zainicjowana.
- Jeśli czynność może zostać zainicjowana przez kilka przejść:
 - Jeśli **AND** jest warunkiem wejściowym czynności, muszą zostać uaktywnione wszystkie przejścia, aby czynność została zainicjowana.
 - Jeśli **OR** jest warunkiem wejściowym czynności, tylko jedno przejście musi być uaktywnione, aby czynność została zainicjowana.

Uwaga:

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

Zadania

Niniejszy podrozdział wyjaśnia, jak tworzyć i wykonywać zadania 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 zatwierdzana automatycznie:

- Jeżeli czynność jest 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 AssetCenter Server lub agent AssetCenter.

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

Częstotliwość, z którą AssetCenter Server monitoruje funkcje workflow jest zdefiniowana w opcjach AssetCenter Server.

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:

- 1 Kliknij czynność prawym przyciskiem myszy.
- 2 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 AssetCenter, 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 AssetCenter.

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**.

Zdarzenie systemowe

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

Odpowiadają one różnym możliwym wynikom (rezultat) 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".

Zdarzenie 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.

Zdarzenie 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.

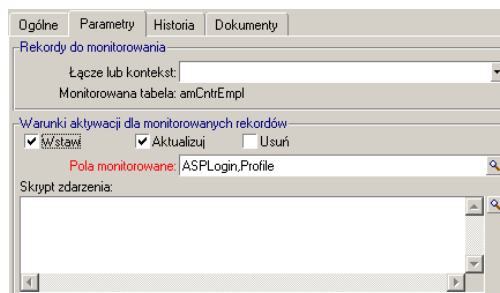
Ostrzeżenie:

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:

Rysunek 8.7. 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", w rozdziale "Workflow" niniejszego podręcznika.

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 o naturze "PC".

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

W tym przypadku, AssetCenter Server inicjuje zdarzenie.

Częstotliwość, z jaką AssetCenter Server inicjuje zdarzenia typu **Okresowo** jest definiowana w polu planowania w zakładce **Parametry** szczegółów zdarzenia.

Sposób przetwarzania zdarzenia opisano w sekcji pt. "Przetwarzanie zdarzenia", w rozdziale "Workflow" 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:

Tabela 8.1. Przypadki ograniczenia instancji workflow

Opcja Pojedyncza aktywna instancja workflow dla obiektu w zakładce Ogólne schematu workflow.	
Zatwierdzona	Nie zatwierdzona

Opcja Ponownie zainicjuj instancję workflow jeżeli jest w toku w zakładce Ogólne 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).

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:

Tabela 8.2. Sposoby przetwarzania zdarzenia

	Rejestracja zdarzenia i przetwarzanie przez serwer	Rejestracja zdarzenia i przetwarzanie natychmiastowe	Przetwarzanie natychmiastowe bez rejestracji
Zdarzenie typu Okresowo	AssetCenter Server inicjuje zdarzenie, jeśli są spełnione warunki aktywacji. Częstotliwość inicjowania jest zdefiniowana w zakładce Parametry szczegółów zdarzenia.	AssetCenter Server inicjuje zdarzenie, jeśli są spełnione warunki aktywacji. Częstotliwość inicjowania jest zdefiniowana w zakładce Parametry szczegółów zdarzenia.	AssetCenter Server inicjuje zdarzenie, jeśli są spełnione warunki aktywacji. Częstotliwość inicjowania jest zdefiniowana w zakładce Parametry szczegółów zdarzenia.
	Natychmiast po jego wystąpieniu, AssetCenter Server rejestruje zdarzenie w tabeli o nazwie SQL amWfOccurEvent .	Natychmiast po jego wystąpieniu, AssetCenter Server rejestruje zdarzenie w tabeli o nazwie SQL amWfOccurEvent .	Natychmiast po jego wystąpieniu, AssetCenter Server nie rejestruje zdarzenia w tabeli o nazwie SQL amWfOccurEvent .
	Przejście jest aktywowane później przez AssetCenter Server (częstotliwość, z którą AssetCenter Server monitoruje przejścia do uaktywnienia jest zdefiniowana w opcjach programu AssetCenter Server).	Przejście jest aktywowane natychmiast przez AssetCenter Server.	Jednak przejście jest aktywowane natychmiast przez AssetCenter Server.

	Rejestracja zdarzenia i przetwarzanie przez serwer	Rejestracja zdarzenia i przetwarzanie natychmiastowe	Przetwarzanie natychmiastowe bez rejestracji
Zdarzenie typu Baza danych lub Zdarzenia systemowe inicjowane przez AssetCenter (wynik czynności typu Pytanie lub Działanie użytkownika, wynik czynności typu Działanie automatyczne lub Test / skrypt wykonywanej przez AssetCenter)	Natychmiast po wystąpieniu zdarzenia, jest ono rejestrowane w tabeli o nazwie SQL amWfOccurEvent przez stację roboczą AssetCenter. Przejście jest aktywowane później przez AssetCenter Server (częstotliwość, z którą AssetCenter Server monitoruje przejścia do uaktywnienia jest zdefiniowana w opcjach programu AssetCenter Server).	Natychmiast po wystąpieniu zdarzenia, jest ono rejestrowane w tabeli o nazwie SQL amWfOccurEvent przez stację roboczą AssetCenter. Przejście jest aktywowane natychmiast przez stację roboczą AssetCenter.	Po wystąpieniu zdarzenia, nie jest ono rejestrowane w tabeli o nazwie SQL amWfOccurEvent , jednak przejście jest aktywowane natychmiast przez stację roboczą AssetCenter.

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

Tabela 8.3. Sposoby przetwarzania zdarzenia

	Rejestracja zdarzenia i przetwarzanie natychmiastowe	Przetwarzanie natychmiastowe bez rejestracji
Zdarzenie typu Okresowo	AssetCenter Server inicjuje zdarzenie, jeśli są spełnione warunki aktywacji. Częstotliwość inicjowania jest zdefiniowana w zakładce Parametry szczegółów zdarzenia. Natychmiast po jego wystąpieniu, AssetCenter Server rejestruje zdarzenie w tabeli o nazwie SQL amWfOccurEvent . Przejście jest aktywowane natychmiast przez AssetCenter Server.	AssetCenter Server inicjuje zdarzenie, jeśli są spełnione warunki aktywacji. Częstotliwość inicjowania jest zdefiniowana w zakładce Parametry szczegółów zdarzenia. Gdy występuje zdarzenie, nie jest ono rejestrowane w tabeli o nazwie SQL amWfOccurEvent , jednak przejście jest aktywowane natychmiast przez AssetCenter Server.
Zdarzenie typu Baza danych lub Zdarzenia systemowe inicjowane przez AssetCenter (wynik czynności typu Pytanie lub Działanie użytkownika, wynik czynności typu Działanie automatyczne lub Test / skrypt wykonywanej przez AssetCenter)	Natychmiast po wystąpieniu zdarzenia, jest ono rejestrowane w tabeli o nazwie SQL amWfOccurEvent przez stację roboczą AssetCenter. Przejście jest aktywowane natychmiast przez stację roboczą AssetCenter.	Po wystąpieniu zdarzenia, nie jest ono rejestrowane w tabeli o nazwie SQL amWfOccurEvent , jednak przejście jest aktywowane natychmiast przez stację roboczą AssetCenter.
Zdarzenie systemowe inicjowane przez AssetCenter Server (wynik czynności typu Działanie automatyczne lub Test / skrypt wykonywanej przez AssetCenter Server) lub zdarzenia wyzwalanego przez alarm czynności.	Natychmiast po wystąpieniu zdarzenia, jest ono rejestrowane w tabeli o nazwie SQL amWfOccurEvent przez AssetCenter Server. Przejście jest aktywowane natychmiast przez AssetCenter Server.	Gdy występuje zdarzenie, nie jest ono rejestrowane w tabeli o nazwie SQL amWfOccurEvent , jednak przejście jest aktywowane natychmiast przez AssetCenter Server.

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

Zgodnie z wyborami dokonanyymi 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

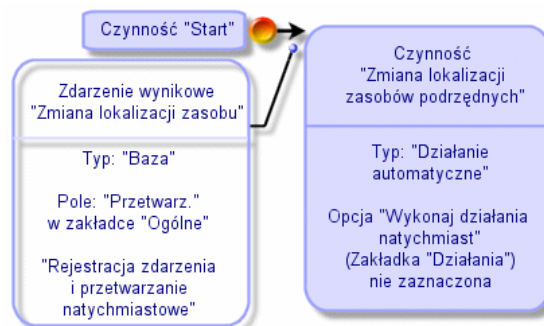
Aby zrealizować synchroniczny schemat workflow, trzeba zdefiniować:

- Zdarzenia typu **Baza danych**, a mianowicie **Rejestracja zdarzenia i przetwarzanie natychmiastowe** (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:

Rysunek 8.8. Przykład synchronicznego schematu workflow



W tym przypadku, gdy następuje zmiana lokalizacji zasobu i klikniesz przycisk **Modyfikuj**:

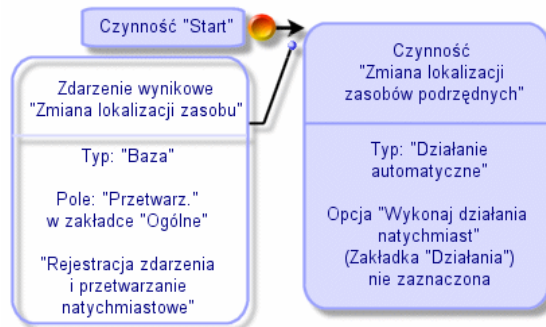
- 1 Rozpoczyna się transakcja bazy danych.v
- 2 Lokalizacja zasobu zostaje zmodyfikowana.
- 3 Uruchamiana jest instancja workflow.
- 4 Aktywowane jest przejście workflow.
- 5 Zmieniana jest lokalizacja zasobów podrzędnych.
- 6 Następnie cała transakcja jest sprawdzana.

Jeśli podczas jednego z tych etapów wystąpi błąd, zarówno lokalizację 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.

Rysunek 8.9. Przykład asynchronicznego schematu workflow



Zdarzenie końcowe

Definicja

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

Przykład:

Rysunek 8.10. 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:

- 1 Kliknij zdarzenie prawym przyciskiem myszy.
- 2 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 AssetCenter.

Ostrzeżenie:

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

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 AssetCenter Server. Częstotliwość monitorowania jest zdefiniowana w opcjach programu AssetCenter Server.

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 schematów 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.

AssetCenter Server monitoruje tworzenie nowych grup wykonawczych workflow.

Gdy tylko AssetCenter Server wykryje nową grupę wykonawczą G, tworzy nowy moduł monitorowania **Wykonanie zasad workflow dla grupy wykonawczej "G"**.

Mechanizm ten ma następujące zalety:

- Pozwala on zdefiniować harmonogramy weryfikacji dla każdej grupy wykonawczej workflow.
- Różne grupy wykonawcze workflow mogą być monitorowane przez różne instancje AssetCenter Server.

Po wykryciu grupy wykonawczej workflow, AssetCenter Server 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.

Usuwanie zakończonych instancji workflow

Konieczność usuwania zakończonych instancji workflow

Podczas wykonywania schematów workflow tworzone są instancje workflow (tabela **Instancje workflow** (amWfInstance)).

Instancje workflow nie są usuwane automatycznie, nawet po zakończeniu ich wykonywania (pole **Stan** (seStatus)).

Jeżeli schematy workflow są często używane, liczba utworzonych instancji workflow może być bardzo znaczna.

Powoduje to niepotrzebny rozrost rozmiarów bazy danych i może zmniejszyć wydajność AssetCenter.

Radzimy regularnie usuwać instancje workflow, których wykonanie zostało zakończone.

Automatyczne usuwanie zakończonych instancji workflow

Aby zautomatyzować usuwanie zakończonych instancji workflow w produkcyjnej bazie danych:

- 1 Dodaj pole do tabeli **Schematy workflow** (amWfScheme), aby określić okres czasu, po upływie którego przestarzałe instancje schematu workflow zostaną usunięte.
- 2 Utwórz działanie usuwające przestarzałe instancje workflow.
- 3 Utwórz schemat workflow automatyzujący wykonanie tego działania.
- 4 Skonfiguruj AssetCenter Server w celu zautomatyzowanie wykonania tego schematu workflow.

Dodaj pole do tabeli Schematy workflow

Dodaj następujące pole do tabeli **Schematy workflow** (amWfScheme):

Parametr	Wartość
Nazwa SQL	AutoCleaningDelay

Parametr	Wartość
Nazwa	Termin usuwania instancji
Opis	Okres czasu, po którym są usuwane instancje workflow
Typ	Czas trwania (okres czasu)
Utwórz indeks dla pola	Tę opcję należy pozostawić niezaznaczoną.
Opis (zakładka Pomoc)	Określa, po upływie jakiego czasu można usunąć zakończone instancje workflow.
Przykład	-1: instancje workflow nie są nigdy usuwane.Wartość dodatnia lub pusta: instancje workflow są usuwane po upływie określonego terminu.

Informacje na temat dodawania pola do istniejącej tabeli można znaleźć w podręczniku **Administracja**, rozdział **Dostosowywanie bazy danych**, podrozdział **Tworzenie nowych obiektów/ Tworzenie pól, łączy lub indeksów**.

Wypełnij pole Termin usuwania instancji.

Wypełnij pole **Termin usuwania instancji** (AutoCleaningDelay) w każdym z używanych schematów workflow.

Połączenie z produkcyjną bazą danych

- 1 Uruchom AssetCenter.
- 2 Połącz się z produkcyjną bazą danych.

Tworzenie działania usuwającego przestarzałe instancje

- 1 Wyświetl listę działań (menu **Narzędzia/ Działania/ Edytuj**).
- 2 Kliknij przycisk **Nowy**.
- 3 Wypełnij następujące pola:

Nazwa	Nazwa SQL	Wartość
Nazwa	Name	Usuwanie zakończonych instancji workflow
Kontekst	ContextTable	Schematy workflow (amWfScheme)
Type	seActionType	Skrypt
Nazwa SQL	SQLName	DeleteFinishedWfInstances
Skrypt działania	Skrypt	Patrz (*) poniżej.

(*) Skrypt działania:

```

Const NumberOfInstanceToDelete = 50

Dim IRc As Long
Dim i As Long

i = 0

If [AutoCleaningDelay] >= 0 Then
  Dim hqWfInstance As Long
  hqWfInstance = AmQueryCreate()
  IRc = AmQueryExec(hqWfInstance, "SELECT IWfInstanceId FROM
amWfInstance WHERE IWfSchId = "& [IWfSchId] & " And seStatus =
1 AND ADDSECONDS(dtCompleted, " & [AutoCleaningDelay] & ")
< GetDate()" )

  Do While (IRc = 0 And i < NumberOfInstanceToDelete)
    Dim hrWfInstance As Long
    hrWfInstance = AmGetRecordHandle(hqWfInstance)
    IRc = AmDeleteRecord(hrWfInstance)
    IRc = AmReleaseHandle(hrWfInstance)
    IRc = AmQueryNext(hqWfInstance)
    i = i + 1
  Loop
End If

```

- 4 Kliknij przycisk **Utwórz**.

Tworzenie schematu workflow automatyzującego wykonanie działania

- 1 Wyświetl listę schematów workflow (menu **Narzędzia/ Workflow/ Schematy workflow**).
- 2 Kliknij przycisk **Nowy**.
- 3 Wypełnij następujące pola:

Nazwa	Nazwa SQL	Wartość
Nazwa	Name	Usuwanie zakończonych instancji workflow
Poz. katalogowa	Ref	ADM_CLEAN_WF_INSTC
Kontekst obiektu początkowego	StartContextTable	Schematy workflow (amWfScheme)

Nazwa	Nazwa SQL	Wartość
Grupa wykonawcza	GroupName	Podaj nazwę w celu zautomatyzowania wykonywania schematy workflow w AssetCenter Server (na przykład ADMIN).
Jedno aktywne workflow dla obiektu	bUniqueActive	Zaznacz tę opcję.
Nie zapisuj instancji w bazie	bTransient	Tę opcję należy pozostawić niezaznaczoną.

- 4 Kliknij przycisk **Utwórz**.
- 5 Wybierz zakładkę **Czynności**.
- 6 Kliknij czynność **Początek** i wybierz menu **Dodaj zdarzenie**.
- 7 Wypełnij następujące pola:

Nazwa	Nazwa SQL	Wartość
Nazwa	Name	Timer
Typ	seMonitoringType	Okresowe
Uruchom workflow ponownie, jeżeli jest już inne aktywne workflow	Okresowe	Tę opcję należy pozostawić niezaznaczoną.
Warunek AQL	AQLCond	AutoCleaningDelay >= 0

- 8 Wybierz i wypełnij zakładkę **Parametry** w zależności od potrzeb.
- 9 Kliknij przycisk **Dodaj**.
- 10 Kliknij prawym przyciskiem myszy i wybierz menu **Dodaj czynność**.
- 11 Wypełnij następujące pola:

Nazwa	Nazwa SQL	Wartość
Nazwa	Name	Clean W/F instances
Typ	seType	Działanie automatyczne
Rejestruj zadanie	bLogWorkItem	Zaznacz tę opcję.
Kontekst	ContextTable	Schematy workflow (amWfScheme)
Warunek rozpoczęcia	seInCond	OR

- 12 Kliknij przycisk **Dodaj**.
- 13 Wybierz zakładkę **Parametry**.
- 14 Wypełnij następujące pola:

Nazwa	Nazwa SQL	Wartość
Wykonaj działania natychmiast	bExecImmediately	Zaznacz tę opcję.
Działania	Działania	Usuwanie zakończonych instancji workflow

- 15 Kliknij przycisk **Zamknij**.
- 16 Kliknij dwukrotnie czynność **Clean W/F instances**.
- 17 Kliknij prawym przyciskiem zdarzenie **Wykonane** i wybierz menu **Szczegóły zdarzenia**.
- 18 Wypełnij następujące pola:

Nazwa	Nazwa SQL	Wartość
Przetworzono	seProcessingMode	Rejestracja zdarzenia i przetwarzanie natychmiastowe

- 19 Kliknij przycisk **Zamknij**.
- 20 Za pomocą myszki utwórz łącze między czynnościami **Początek** i **Clean W/F instances**.
- 21 Kliknij prawym przyciskiem zdarzenie **Wykonane** i wybierz menu **Zdarzenie końcowe**.
- 22 Kliknij przycisk **Modyfikuj**.

Konfigurowanie AssetCenter Server w celu zautomatyzowania wykonania schematu workflow

- 1 Uruchom AssetCenter Server.
- 2 Skonfiguruj moduł, który uruchomi wykonanie workflow **Usuwanie zakończonych instancji workflow** (menu **Narzędzia/ Konfiguruj moduły**).
Jest to moduł o nazwie **Wykonywanie reguł workflow dla grupy wykonawczej 'X'**, gdzie **X** oznacza wartość pola **Grupa wykonawcza** (GroupName) określoną w schemacie workflow.
- 3 Aby workflow było wykonywane automatycznie AssetCenter Server musi być aktywny.

9 Eksport danych i tworzenie widoków SQL

ROZDZIAŁ

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

Definicje skryptu i zapytania eksportu

Skrypty eksportu

Skrypty eksportu umożliwiają wyeksportowanie danych lub (ponowne) utworzenie/ usunięcie widoków SQL za pomocą AssetCenter Export lub **amexp.exe**. Skrypty eksportu można zapisać w celu ich ponownego użycia.

Skrypt eksportu zawiera:

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

Skrypt eksportu działa:

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

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 kontekstowe **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

- 1 Uruchom moduł AssetCenter Export za pomocą menu **Start** lub z grupy programu AssetCenter.
- 2 Zdefiniuj skrypt eksportu z trybem ustawionym na "Tryb eksportu":

- 1 W zakładce **Zapytania** wpisz zapytania definiujące eksportowane dane.
 - 2 W zakładce **Formatowanie** określ format plików tekstowych, do których zostaną wyeksportowane dane.
 - 3 Użyj polecenia menu **Plik/ Zapisz skrypt** lub **Plik/ Zapisz skrypt jako**, aby zapisać skrypt.
- 3 Wykonaj skrypt eksportu.
- bądź bezpośrednio w AssetCenter Export za pomocą ikony <Wykonaj skrypt> 
 - bądź uruchamiając **amexpl.exe** w oknie DOS.

 **Uwaga:**

Zgodnie z zasadami ograniczenia dostępu określonymi w AssetCenter, można uruchomić AssetCenter Export lub wykonać **amexpl.exe** jedynie posiadając uprawnienia administratora (login "Admin" lub użytkownik z uprawnieniami administratora).

Eksport danych za pomocą polecenia menu kontekstowego **Eksportuj listę**

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

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

 **Uwaga:**

Dalsze informacje o poleceniu menu **Plik/ Eksportuj** znajdują się w podręczniku **Ergonomia**, rozdział "Pierwsze kroki z AssetCenter", akapity "Listy rekordów", punkt "Eksportowanie listy".

Zarządzanie widokami SQL w bazie danych AssetCenter

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

Uwaga:

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:

- 1 Uruchom program AssetCenter Export.
- 2 Zdefiniuj skrypt eksportu z trybem ustawionym na "Tryb widoku":
 - 1 W zakładce **Zapytania** wpisz zapytania definiujące pobierane dane.
 - 2 W zakładce **Widoki** określ działania do wykonania: utworzenie, modyfikacja lub usunięcie widoków, wykonanie skryptu SQL bezpośrednio lub zapisanie go do pliku.
 - 3 Zapisz skrypt eksportu.
- 3 Wykonaj skrypt eksportu.
 - bezpośrednio w program AssetCenter Export
 - bądź uruchamiając **amexpl.exe**.

Zalecenia

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 kod wewnętrzny zasobów...

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 programu AssetCenter Export.

Skrypt eksportu zawiera:

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

Działania do wykonania, dotyczące tworzenia/usuwania widoków SQL.

Skrypt eksportu działa:

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

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:

- 1 Uruchom program AssetCenter Export.
- 2 Otwórz właściwą bazę danych. Ostrzeżenie: Połączenie można nawiązać tylko za pomocą nazwy logowania "Admin" lub nazwy logowania użytkownika o uprawnieniach administratora.
- 3 Utwórz nowy skrypt za pomocą polecenia menu **Plik/ Nowy skrypt** lub otwórz skrypt do edycji za pomocą polecenia menu **Plik/ Otwórz skrypt**.
- 4 U góry ekranu AssetCenter Export zdefiniuj, czy chcesz eksportować dane (Tryb eksportu), czy zarządzać widokami SQL (Tryb widoku) bazy danych.
- 5 Wpisz zapytania skryptu eksportowego w zakładce **Zapytania**.

- 6 Jeśli eksportujesz dane, podaj wyjściowy format eksportowanych danych w zakładce **Formatowanie**.
- 7 Jeśli chcesz zarządzać widokami SQL, określ co chcesz zrobić w zakładce **Widoki**.
- 8 Zapisz skrypt używając polecenia menu **Plik/ Zapisz skrypt** lub **Plik/ Zapisz skrypt jako**.

Definiowanie zapytań eksportowych

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

- Kliknij przycisk **Nowy**, aby dodać zapytanie eksportowe.
- Kliknij przycisk **Usuń**, aby usunąć wybrane zapytanie eksportowe.

Utwórz zapytanie w skrypcie eksportowym


- 1 Kliknij przycisk **Nowy** w zakładce **Zapytania**.
- 2 Podaj nazwę zapytania. Nazwa ta jest używana w raporcie wykonania w zakładce **Wiadomości** szczegółów skryptu eksportowego.
- 3 Możesz wpisać komentarze (nie są one eksportowane).
- 4 Zdefiniuj dane do pobrania w polu **Zapytanie**.
- 5 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.

Można albo wpisać zapytanie, albo kliknąć przycisk  w celu uzyskania dostępu do okna pomagającego przy tworzeniu 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(IModelId) FROM amModel 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:

- 1 Aby zdefiniować indeks sortowania, wybierz indeks w polu **Sortuj według indeksu**.
- 2 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 amModel
```

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

```
SELECT DISTINCT Brand FROM amModel
```

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.

Kliknięcie ikony  umożliwia podgląd wyniku zapytania w postaci listy rekordów. AssetCenter wyświetla liczbę rekordów spełniających zapytanie w prawej dolnej części okna.

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), UFT-8, UNICODE i Latin 1.

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), miesiący (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:
 - 1 Podaj nazwę i ścieżkę pliku, w którym będzie przechowywany wygenerowany SQL klikając przycisk .
 - 2 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 .

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

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

Wykonywanie skryptu eksportu z programu AssetCenter Export

Wykonywanie skryptu eksportu z programu AssetCenter Export:

- 1 Uruchom program AssetCenter Export.
- 2 Zdefiniuj i zapisz skryptu eksportu
- 3 Następnie wykonaj skrypt:
 - za pomocą polecenia menu **Działania/ Wykonaj skrypt**,
 - lub naciskając F8, albo
 - bądź klikając ikonę .

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

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.
- Eksport powiódł się.
- ▲ Ostrzeżenie.

Wykonywanie skryptów eksportowych w systemie DOS

Omówienie

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

Można wówczas wykonać, ręcznie lub automatycznie (na przykład za pomocą pliku batch), polecenie eksportu za pomocą programu **amexpl.exe**, który znajduje się w folderze podrzędnym **bin** folderu instalacyjnego AssetCenter.

Składnia

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

-verbose: wyświetla komunikaty podczas eksportu.

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

-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.

Ciągi znaków ujęte w <> nie mogą zawierać spacji.

Przykład:

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


10 Skrypty

ROZDZIAŁ

W niniejszym rozdziale opisano używanie skryptów.

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.

 **Uwaga:**

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

- Skrypty deklaracyjne. Są to skrypty eksportu i importu obiektów bazy danych AssetCenter, 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.
 - Kontrolki języka Visual Basic for Applications nie są dostępne.
-

 **Uwaga:**

Aby odnaleźć odnośnik w "Programmer's Reference" do funkcji lub kluczowego pojęcia języka Basic, umieść kursor na danym słowie i naciśnij klawisz F1: powoduje to wyświetlenie pomocy kontekstowej.

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 z tabeli modeli:

```
[Category.FullName]
```

 **Uwaga:**

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.

Przykład:

```
[RetVal=[Contact.Location] lub RetVal=[Contact.lLocalId]
```

Obydwa przykłady zwracają ten sam wynik, identyfikator łącza.

Zastosowanie skryptów

AssetCenter umożliwia wykorzystanie skryptu w języku Basic:

- aby skonfigurować wartość domyślną pola (menu kontekstowe **Konfiguruj obiekt**).
- Wartość domyślna cechy związanej z tabelą:
- W polach wyliczanych typu "Basic".
- aby skonfigurować pola (menu kontekstowe **Konfiguruj obiekt** 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 w tabeli amWfOrgRole.

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 <Nazwa funkcji> (<Parametr> As <Typ parametru>[, ..., <Parametr> As <Typ parametru>]) As <Typ funkcji>
```

```
<Program (skrypt) wykonywany przez funkcję. Program określa wartość zwrótną.>
```

```
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** (zwaney również "wartością zwracaną") przez funkcję programowalną, w następującym formacie:

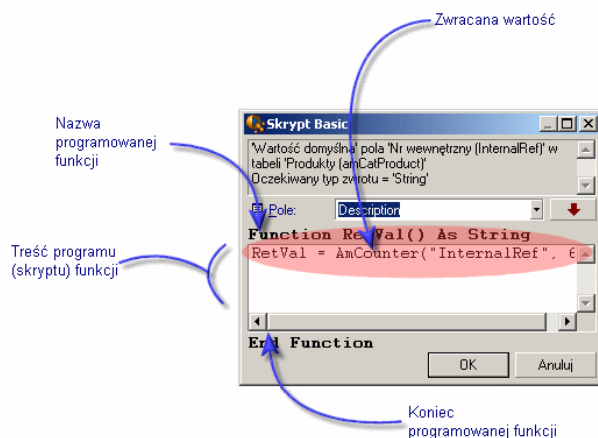
```
RetVal=<Expression>
```

Uwaga:

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:

Rysunek 10.1. Skrypt - 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 typu "Data", a AssetCenter wyświetli komunikat o błędzie, gdy zostanie utworzony kolejny rekord w tej tabeli.

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:

Tabela 10.1. Funkcje/parametry - typy

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 <Nazwa funkcji>() As <Typ funkcji>
```

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.

Uwaga:

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=<Wyrażenie odpowiadające typowo funkcji>
```


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 TM".
- 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 podrozdziale 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 z języku Basic

Przykładowy scenariusz

Cel

Zagwarantować, aby cecha "Podręcznik" była dostępna jedynie dla modelu "Komputer/ Płyta główna/" i jego podrzędnych.

Metoda


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

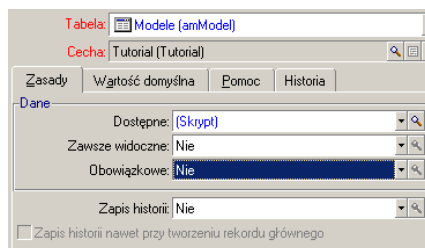
Krok 1: Utwórz cechę "Podręcznik"

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


Nazwa pola	Wartość pola
Nazwa (Nazwa SQL: TextLabel)	"Podręcznik"
Nazwa SQL (Nazwa SQL: SQLName)	"Podręcznik"
Typ wpisu (Nazwa SQL: seDataType)	Numeryczne

Kliknij przycisk **Utwórz**, aby utworzyć tę cechę.

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



Krok 2: Otwórz okno edycji

W zakładce **Ograniczenia** szczegółów parametrów cechy "Podręcznik", wybierz wartość "(Script)" dla pola **Dostępne** (nazwa SQL: seAvailable) jako . Kliknij przycisk . AssetCenter otworzy okno edycji skryptu:

Krok 3: Analiza i definicja algorytmu

Krok 3: Analiza i definicja algorytmu


- Ustawiać wartość pola **Dostępne** (nazwa SQL: seAvailable) na **Tak**, jeśli model jest "/Komputer/ Płyta główna/" lub jego podrzędne.
- Ustawiać wartość pola **Dostępne** (nazwa SQL: seAvailable) na **Nie** we wszystkich pozostałych przypadkach.

Nasz algorytm ma wobec tego następującą postać:

```
Jeżeli nazwa modelu zaczyna się od "/Komputer/Płyta główna" wówczas
Cecha jest dostępna
W przeciwnym wypadku
Cecha nie jest dostępna
```

Wartość pola **Pełna nazwa** (nazwa SQL: FullName) w tabeli modeli warunkuje zatem wartość pola **Dostępne** (nazwa SQL: seAvailable) cechy. Tylko to pole występuje w naszym algorytmie.

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

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 3.

```
If Left([FullName], Len("/Komputer/ Płyta główna"))="/Komputer/ Pł
yta główna" Then
  RetVal=1
Else
  RetVal=0
End If
```

 **Uwaga:**



Skrypty nie rozróżniają dużych i małych liter.

Kliknij przycisk **OK**, aby zatwierdzić skrypt.

Krok 5: Test skryptu Basic

W tym kroku sprawdzamy, czy skrypt działa prawidłowo.

- 1 Otwórz tabelę modeli wybierając polecenie menu **Repozytorium/Modele**. Kliknij przycisk **Nowy**, aby utworzyć nowy model.
- 2 Wypełnij tylko pola obowiązkowe.
 - 1 **Nazwa**

- 2 **Model nadrzędny** (Nazwa SQL: Parent): "Komputer/Płyta główna"
 - 3 **Natura** (Nazwa SQL: Nature)
 - 4 **Kod kreskowy** (nazwa SQL: BarCode)
 - 3 Kliknij przycisk **Utwórz**, aby utworzyć nowy model.
 - 4 Wybierz zakładkę **Cechy** i kliknij przycisk , aby dodać cechę. Na ekranie wyboru wyświetlona jest nazwa cechy, dla której właśnie utworzono skrypt.
 - 5 Zmień wartość pola **Model nadrzędny** na "/Komputer/" i zatwierdź zmianę klikając przycisk **Modyfikuj**.
 - 6 Wybierz zakładkę **Cechy** w szczegółach modelu i kliknij przycisk , aby dodać cechę. Ekran wyboru nie wyświetla już nazwy cechy, dla której właśnie utworzono skrypt.
- Skrypt prawidłowo spełnia swe zadanie.

Biblioteka skryptów

AssetCenter umożliwia zapisywanie bibliotek skryptów, aby scentralizować dostęp do tych skryptów.

Dostęp do biblioteki skryptów można uzyskać za pomocą menu **Administracja/Biblioteki skryptów**.

Zapisane biblioteki są wywoływane za pomocą polecenia API **amEvalScript**.

Więcej informacji na temat API **amEvalScript** znajduje się w podręczniku "Programmer's Reference", w podrozdziale "Alphabetic reference".

Pojęcia

W AssetCenter skrypt określa funkcję.

Utworzenie biblioteki skryptów oznacza zdefiniowanie zbioru funkcji.

Tworzenie biblioteki skryptów

Aby utworzyć bibliotekę skryptów:

- 1 Tworzenie listy bibliotek skryptów.

- 2 W polu **Nazwa** wpisz nazwę biblioteki.
- 3 Wpisz skrypt w polu **Skrypty**.
- 4 Zatwierdź klikając przycisk **Utwórz**.

Utwórz, na przykład, bibliotekę nazwaną "biblio" i wpisz następujący skrypt:

```
function FullName(strName As String, strFirstName As String) As String
    FullName = strFirstName & ", " & strName
end function
```

Funkcja ta zwraca ciąg złożony z nazwiska i imienia pracownika.

 **Ostrzeżenie:**

Każda utworzona funkcja musi mieć nazwę różniącą się od pozostałych funkcji w utworzonych bibliotekach skryptów.

Wywoływanie skryptu zapisanego w bibliotece skryptów

Aby wywołać skrypt biblioteki, należy określić kilka parametrów: nazwę biblioteki, funkcję zdefiniowaną w skrypcie i parametry powiązane z tą funkcją.

Utwórz na przykład działanie "callEvalScript" typu skrypt, które wykorzysta uprzednio utworzoną bibliotekę:

- 1 W polu **Kontekst** (nazwa SQL: ContextTable) wpisz nazwę tabeli Pracownicy (nazwa SQL: amEmplDept).
- 2 W zakładce **Skrypt** wpisz następujący skrypt:

```
Dim strFullName As String
strFullName = amEvalScript("biblio", "FullName", "", [Name], [FirstName])
amMsgBox (strFullName)
```

Ten skrypt wywołuje funkcję "FullName" z biblioteki "biblio" i wyświetla nazwisko i imię pracownika w oknie dialogowym.

- 3 Zatwierdź klikając przycisk **Utwórz**.

 Uwaga:

Parametr kontekstu, zwykle używany z API **amEvalSript**, nie jest używany w przypadku wywołania biblioteki skryptów.

Wskazówki i ostrzeżenia

W niniejszej sekcji podano kilka wskazówek pomocnych przy pisaniu skryptów.

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="2001/07/12 13:05:00"
```

 Uwaga:

Jako separator daty można również stosować łącznik ("-").

Daty w języku "Basic"

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).

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.

 Uwaga:

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

Dostęp do spisu systemowego w trybie odczytu i zapisu

AssetCenter zarządza spisami systemowymi przypisując liczbę całkowitą do każdej możliwej wartości na spisie.

Weźmy przykład spisu używanego do wypełniania pola **Przypisanie** (nazwa SQL: seAssignment) w ramce **Przypisanie** zakładki **Ogólne** szczegółów zasobu.

W poniższej tabeli podano wartości używane na spisie:

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

Zatem, aby zdefiniować domyślną wartość spisu, należy:

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

```
RetVal=<Liczba całkowita związana z właściwą wartością>
```

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

```
RetVal=3
```

Uwaga:

Nie należy mylić spisu systemowego z zamkniętym spisem zdefiniowaną przez użytkownika.

Uwaga:

Pełny wykaz wartości spisu systemowego znajduje się w pliku Database.txt, zapisanym w folderze podrzędnym doc\infos foldera instalacyjnego AssetCenter. Dwie kolumny, "Data display and entry type" i "Additional information on data display and entry type", opisują typ listy pozycji i wartości przyjmowane przez spis.

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.

Uwaga:

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.IPersId]`

Komentarze do skryptu 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]=
```

- 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 (<Numer błędu>, <Komunikat o błędzie>)
```

Uwaga:

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.

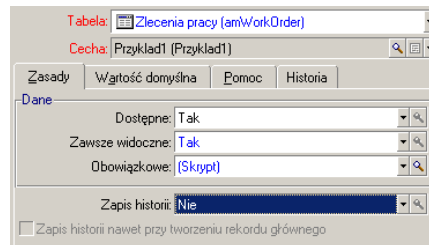
Pierwszy przykład

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

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:



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** (Nazwa SQL: seStatus) w tabeli zleceń pracy.

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

To pole jest wypełniane na podstawie spisu systemowego. Jest ona następująca:

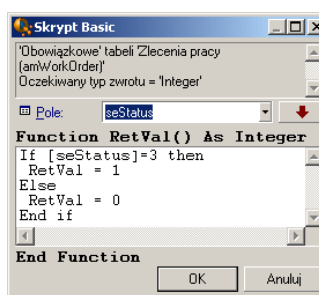
Wartość spisu	Skojarzona liczba
Zgłoszone	0
Zaplanowane	1
W toku	2
Zamknięte	3

Wartość ze spisu, która nas interesuje, to:

- **Zamknięte**, co odpowiada wartości "3"

Krok 2: Utworzenie skryptu Basic

Teraz algorytm z kroku 1 trzeba wyrazić w języku Basic.



Kliknij przycisk **OK**, aby zatwierdzić skrypt.

Krok 3: Test skryptu Basic

Etap ten pozwala zagwarantować, że skrypt będzie działał prawidłowo.

- 1 Otwórz tabelę zleceń pracy za pomocą polecenia menu **Repozytorium/Zlecenia pracy** i wybierz zlecenie pracy, którego stan ma wartość **Zamknięte** (jeśli brak takiej karty, utwórz zlecenie pracy o takim stanie).
- 2 Wybierz zakładkę **Cechy**. AssetCenter dodał cechę do zlecenia pracy, a cecha ta jest obowiązkowa.
- 3 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

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"
```

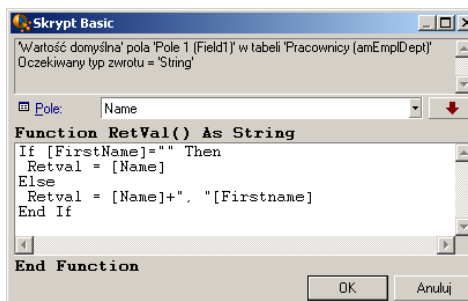
Wartości pól **Nazwisko** (nazwa SQL: Name) i **Imię** (nazwa SQL: FirstName) w tabeli pracowników warunkują wartość domyślną pola **Pole1** (nazwa SQL: Field1) w szczegółach pracownika. Tylko te dwa pola pojawiają się w naszym algorytmie.

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.



Kliknij przycisk **OK**, aby zatwierdzić skrypt.

Krok 3: Test skryptu Basic

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

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

Skrypt działa prawidłowo.

11

Kalendarze

ROZDZIAŁ

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

Omówienie kalendarzy

Omówienie kalendarzy

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** (Nazwa SQL: tsDelay) definiowany w tych alarmach jest zatem podawany w godzinach pracy.

Ostrzeżenie:

Po zmodyfikowaniu kalendarza w bazie danych, aby zmiany w polach AssetCenter powiązanych z kalendarzem zostały wzięte pod uwagę musisz zamknąć i ponownie uruchomić AssetCenter.

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

Aby utworzyć kalendarz wykonaj następujące kroki:

- 1 Rozpocznij od identyfikacji kalendarza nadając mu **Nazwa** (nazwa SQL: Name).
- 2 Jeśli jest taka potrzeba, skojarz kalendarz ze strefą czasową wypełniając pole **Strefa czasowa** (nazwa SQL: TimeZone).
- 3 Zdefiniuj codzienne godziny pracy w zakładce **Harmonogramy** szczegółów kalendarza.
- 4 Zdefiniuj codzienne godziny pracy w zakładce **Wyjątki** szczegółów kalendarza.

- 5 Na koniec możesz zweryfikować działanie kalendarza w zakładce **Podgląd**.

Opis tworzenia kalendarza

Kalendarz tworzony jest w następujących krokach:

- 1 Wprowadzenie informacji ogólnych
- 2 Wypełnianie zakładki Harmonogram
- 3 Wypełnianie zakładki Wyjątki
- 4 Sprawdzanie kalendarza

Wprowadzenie informacji ogólnych

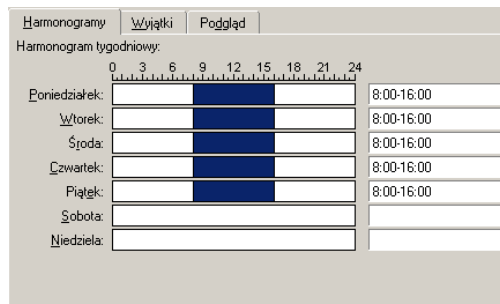
Przed wprowadzeniem aktualnych godzin pracy i wyjątków, należy oznaczyć kalendarz wypełniając pole **Nazwa** (nazwa SQL: Name) na ekranie szczegółów.

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

Wypełnianie zakładki Harmonogram

Zakładka **Harmonogramy** 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**.

Rysunek 11.1. Kalendarze - zakładka Harmonogramy



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.
 - 1 Kliknij kontrolkę na początku okresu harmonogramu.
 - 2 Rozszerz obszar zaznaczenia przeciągając myszą do końca okresu harmonogramu. AssetCenter automatycznie wypełnia pole tekstowe po prawej stronie kontrolki z podziałką.
 - 3 Powtarzaj czynności w miarę potrzeb.
- "Ręcznie", za pomocą pola tekstowego. W polu tym obowiązuje następująca składnia:

```
<Początek okresu roboczego>-<Czas końca okresu roboczego>;< Czas p
oczątku okresu roboczego >-< Czas końca okresu roboczego >;...
```

Dla czasu stosowany jest następujący format:

```
<gg: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.

Uwaga:

AssetCenter automatycznie wypełnia suwak z podziałką znajdujący się po lewej stronie pola tekstowego.

Wypełnianie zakładki Wyjątki

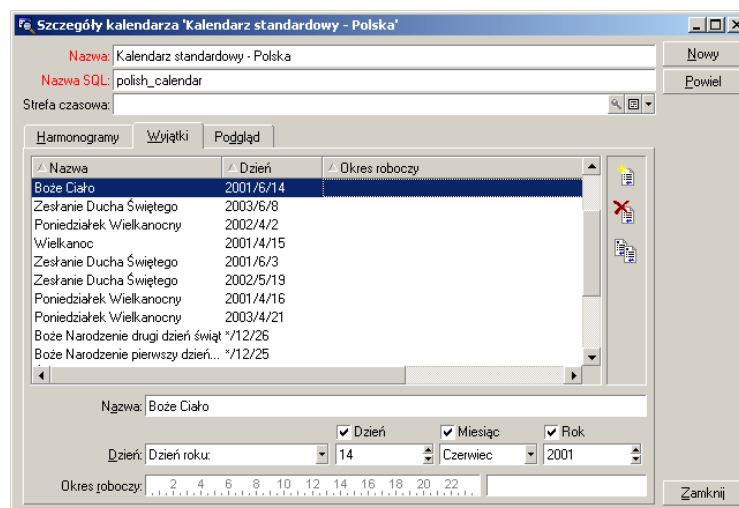
Zakładka **Wyjątki** w szczegółach kalendarza definiuje wyjątki w tygodniowych okresach pracy określonych w zakładce **Harmonogramy**.

Metodologia


Wyjątek tworzy się w następujący sposób:



- 1 Określ Nazwę wyjątku.
- 2 Zdefiniuj obszar zastosowania wyjątku wypełniając pole **Dzień**. Wyjątki można definiować dla danego dnia, miesiąca lub roku.
- 3 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

- 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.

 **Uwaga:**

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 Dzień	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.

Wartość pola Dzień	Kontekst zastosowania wyjątku
"Ostatni"	Reguła znajduje zastosowanie do dnia tygodnia zdefiniowanego w polu Dzień miesiąca(-y) i roku(lat) zdefiniowanych w polach Miesiąc i Rok . Przykład "Ostatni" wtorek każdego miesiąca przez cały rok 2000.

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.

- 1 Kliknij przycisk **Nowy**, aby rozpocząć tworzenie wyjątku.
- 2 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.
- 3 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.

- 1 Kliknij przycisk **Nowy**, aby rozpocząć tworzenie wyjątku.
- 2 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".
- 3 Ponieważ wyjątek odnosi się tylko do wybranych dni (wszystkich dni w sierpniu), pole **Dni** trzeba ustawić na "Dzień roku:..
- 4 W tym okresie pracownicy pracują od godziny 8.30 do 12.30. Na zakończenie wprowadzania wyjątku wprowadź okres od 8.30 to 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** w ramce **Test**.

Rysunek 11.2. Kalendarz - zakładka Podgląd

Test		
Data początkowa	Data końcowa	Uptyw czasu
2002-03-19 15:09:24	2002-03-19 15:09:24	0s
Kalendarz związany z datą początkową		
	0	24
18 marzec 2002		8:00-16:00
19 marzec 2002		8:00-16:00
20 marzec 2002		8:00-16:00
21 marzec 2002		8:00-16:00
22 marzec 2002		8:00-16:00
23 marzec 2002		
24 marzec 2002		

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

 **Uwaga:**

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

12 | Strefy czasowe

ROZDZIAŁ

Niniejszy rozdział wyjaśnia, w jaki sposób używać stref czasowych.

 Uwaga:

Funkcjonalność ta jest dostępna jedynie w przypadku niektórych umów licencyjnych AssetCenter.

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 maja1998 o 17.30.

Wprowadzanie stref czasowych

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



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

Tworzenie stref czasowych

W przeciwieństwie do systemu Windows, AssetCenter zarządza strefami czasowymi biorąc pod uwagę modyfikacje w zasadach zmiany czasu letniego i zimowego. Pozwala to 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.

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

- 1 Wybierz polecenie menu **Plik/ Import**. AssetCenter otwiera okno wyboru importu.
- 2 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.
- 3 Kliknij przycisk **Importuj**. AssetCenter wykonuje import zgodnie ze skryptem.

Zarządzanie strefą czasową

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

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> et <DaylightInfo> osobno zwane są "argumentami"

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

	Argument <DaylightInfo> jest pusty	Argument <DaylightInfo> ma wartość
Argument <Year> 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	Opis	Możliwa wartość
<u>podrzędny</u> <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>"Monday" "Tuesday" "Wednesday" "Thursday" "Friday" "Saturday" "Sunday"</p> <p>Puste (w tym przypadku, należy ustawić wartość argumentu <DDayPos> na wartość pomiędzy 1 i 31, aby określić dzień zmiany z czasu standardowego na czas letni)</p>

Argument	Opis	Możliwa wartość
<SMonth>	Miesiąc zmiany z czasu letniego na czas standardowy.	"January" "February" "March" "April" itp. "November" "December"
<SDayPos>	Pozycja w miesiącu dnia zmiany z czasu letniego na czas standardowy.	"First" "Second" "Third" "Fourth" "Last" "Penultimate" (przedostatni) Wartość między 1 a 31, gdy argument <DDay> 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.	"Monday" "Tuesday" "Wednesday" "Thursday" "Friday" "Saturday" "Sunday" Puste (w tym przypadku, należy ustawić wartość argumentu <SDayPos> na wartość pomiędzy 1 i 31, aby określić dzień zmiany z czasu letniego na czas standardowy)

Argument	Opis	Możliwa wartość
podrzędny <DMonth>	Miesiąc zmiany z czasu standardowego na czas letni.	"January" "February" "March" "April" itp. "November" "December"
<DDayPos>	Pozycja w miesiącu dnia zmiany z czasu standardowego na czas letni.	"First" "Second" "Third" "Fourth" "Last" "Penultimate" (przedostatni) Wartość między 1 a 31, gdy argument <SDay> 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 zarządzanie czasem letnim

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

```
2000=0,60,Sunday,October,last,03:00:00,Sunday,March,last,02:00:00 |
0,60,Sunday,September,last,03:00:00,Sunday,March,last,02:00:00
```

Zidentyfikujmy zastosowane parametry:

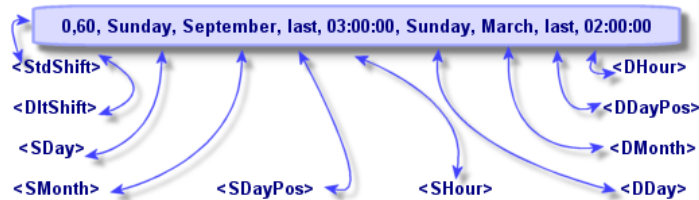
```
2000=0,60,Sunday,October,last,03:00:00,Sunday,March,last,02:00:00
0,60,Sunday,September,last,03:00:00,Sunday,March,last,02:00:00
```

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.

Wynika z tego, że:

 **Uwaga:**

Od roku 2000, 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 2000, 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 programie AssetCenter Server

AssetCenter Server pozwala konfigurować testy dotyczące stref czasowych. Wybierz polecenie menu **Opcje/ Konfiguruj moduły**.

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 AssetCenter Server. 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, AssetCenter Server 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 AssetCenter Server).

Specyficzne aspekty opcji Weryfikuj strefę czasową serwera bazy danych

 Uwaga:

Aby to zrobić, komputer, na którym uruchomiony jest AssetCenter Server musi mieć prawidłowy czas oraz prawidłowe informacje dotyczące zmian czasu letniego.

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).

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:

- 1 Po raz pierwszy, podczas gdy AssetCenter Server nawiązuje połączenie z bazą danych.
- 2 Następnie regularnie, zgodnie z harmonogramem zdefiniowanym na ekranie konfiguracji programu AssetCenter Server (**Narzędzia/Konfiguruj moduły**).

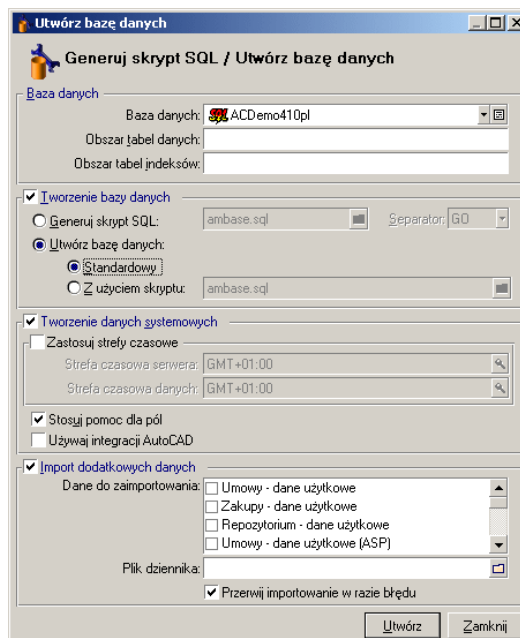
Skutki dla różnych operacji

Strefy czasowe mają wpływ na szereg operacji:

- Tworzenie bazy danych
- Połączenie z bazą danych
- Import i eksport

Tworzenie bazy danych

Przy tworzeniu bazy danych AssetCenter umożliwia zdefiniowanie opcji zarządzania strefami czasowymi. Wybierz polecenie menu **Działanie/ Utwórz bazę danych** w aplikacji AssetCenter Database Administrator. Ramka **Utwórz dane systemowe** zawiera opcje dotyczące stref czasowych.



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.

 Uwaga:

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 **am.ini**, znajdującym się w folderze głównym Windows, pozycji "LocalTimeZone", która określa strefę czasową stacji klienta.

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 am.ini lub strefą czasową zdefiniowaną w systemie Windows.

W następniej tabeli zestawiono różne możliwe sytuacje:

Czy w pliku am.ini jest "LocalTimeZone"?	Czy znaleziono zgodną strefę w tabeli stref czasowych?	Informacja zapisana w pliku am.ini (pozycja "LocalTimeZone")
Tak	Tak	Strefa czasowa bazy danych
	Nie	Bez zmian
Nie	Tak	Strefa czasowa bazy danych
	Nie	Strefa czasowa bazy danych

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:

$$\text{Clock difference} = \text{Modulus}((\text{Różnica czasu pomiędzy danymi minutach})/30)$$

 **Uwaga:**

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)

$$\text{Clock differential} = \text{Modulus}(43/30) = 13 \text{ minutes}$$

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_ITimeZoneCheckInMns** w pliku **am.ini** (znajdującym się w głównym folderze systemu Windows), w sekcji [option].

```
[option]
g_ITimeZoneCheckInMns = 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.

 **Uwaga:**

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.

13 Pola wyliczane

ROZDZIAŁ

Użyj polecenia menu **Administracja/Pola wyliczane**, aby uzyskać dostęp do ekranu tworzenia 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".

 Uwaga:

Pola wyliczane to pola wirtualne, z parametrem tylko do odczytu (w bazie danych AssetCenter 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 Basic**.

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:

- Przedmowa
- Metodologia tworzenia

Przedmowa

Każdy typ pola wyliczanego ma inne właściwości, które określają jego zastosowanie.

W następnym tabeli podsumowano główne różnice występujące pomiędzy trzema typami:

Tabela 13.1. Typy pól wyliczanych

Typ pola	Właściwości pól tego typu			Pole wyliczone przez	Cechy języka, na których opiera się wzór wyliczenia	
	Mogą być wyświetlane	Mogą być sortowane	Mogą być używane w filtrach		Korzyści	Wady
AQL	Tak	Tak	Tak	Serwer bazy danych	Duże możliwości Zintegrowany edytor	Ograniczony język. 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.

 Uwaga:

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
Mogą być wyświetlane	SELECT Warunek
Mogą być sortowane	SELECT ORDER BY GROUP BY Warunki
Mogą być używane w filtrach	SELECT ORDER BY GROUP BY WHERE HAVING Warunki

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

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):
 - Pole wyliczane
 - AQL.
 - Basic.



Uwaga:

Wszędzie tam, gdzie to możliwe, zalecamy stosowanie najmniej "kosztownego" rozwiązania.

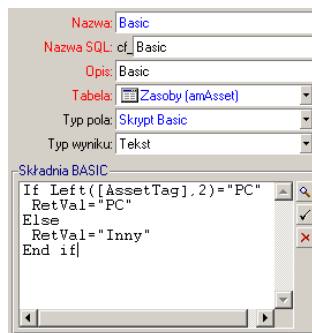
Na przykład:

- Jeśli pole służy tylko do celów informacyjnych, wystarczy pole wyliczane 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 **Administracja/ Pola wyliczane**. AssetCenter wyświetli okno tworzenia 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.

Uwaga:

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 datą, jest zatem wyświetlane w taki sam sposób, jak wszystkie inne pola typu "Data" w bazie danych.

Wpisz wzór wyliczania pola

Pozostaje teraz wyedytować wzór obliczeniowy pola. Wzór można wpisać bezpośrednio w polu tekstowym znajdującym się w dolnej części ekranu (nazwa tego pola zmienia się w zależności od typu pola wyliczanego), albo uzyskać dostęp do pomocy przy wpisywaniu danych dostępnych dla każdego typu pola dzięki kliknięciu przycisku  lub naciśnięciu klawisz "F4".

Uwaga:

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 **Administracja/ Uprawnienia użytkownika**. AssetCenter wyświetli okno tworzenia uprawnień użytkownika.

 **Uwaga:**

Pola wyliczane są dostępne tylko do odczytu.

- 1 Wprowadź krótki opis uprawnienia użytkownika w polu **Opis** (nazwa SQL: Description) (nazwa SQL: Description) i, opcjonalnie, komentarz w polu **Komentarz** (nazwa SQL: Comment).
- 2 Rozwiń strukturę drzewa tabeli związanej z polem wyliczonym. Gałąź oznaczona ikoną  (Pola wyliczane) zawiera pełną listę pól wyliczanych dla danej tabeli.
- 3 Następnie wybierz pole, dla którego chcesz edytować uprawnienia użytkownika. Pole wyboru **Read** w ramce **Pola, łącza i cechy** 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 **Edycja/Opcje**, w zakładce **Wyświetlanie**:

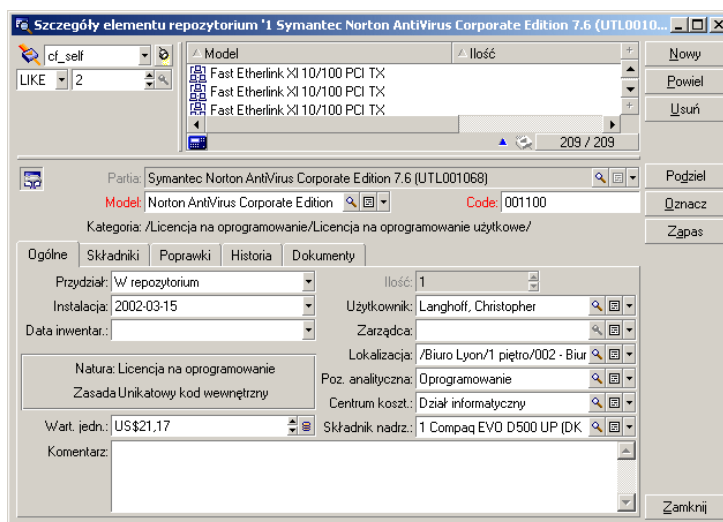
Można dodać je do listy tak samo, jak każde inne pole.

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
- API AssetCenter
- Raporty
- Formularze

Więcej informacji o modułach i zakresach funkcjonalności można znaleźć we właściwej dokumentacji.

14 | Kreatory

ROZDZIAŁ

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.

Zastosowane konwencje

Do opisu struktury kreatorów zastosowano następującą notację

Tabela 14.1. Zastosowane konwencje

[]	Nawiasy kwadratowe oznaczają odwołanie do wartości pola w bazie danych (w przypadku kreatorów kontekstowych) lub jednego z "pól specjalnych": "CurrentSelection" i "CurrentTable". Służą one również do podania parametrów opcjonalnych.
-----	--

< >	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 Basic, apostrof oznacza wiersz komentarza, który nie jest interpretowany przez AssetCenter.
; lub //	W kreatorach średnik lub dwa ukośniki oznaczają linię komentarza, która nie jest interpretowana 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.

Kontrolka

Kontrolka oznacza element graficzny umożliwiający edycję elementu danych. Typowe kontrolki 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".

 Uwaga:

W nazwach węzłów można stosować jedynie znaki alfanumeryczne. Nazwy węzłów mogą mieć maksymalnie długość 22 znaków.

Obiekt

Obiekt, to termin ogólny, który może oznaczać:

- Całego kreatora.
- Stronę kreatora.
- Kontrolka (pole wyboru, pole tekstowe, przycisk, listę rozwijaną, itp.) na stronie.
- Zmienną.
- Itp.

Obiekt nadrzędny i obiekt podrzędny

Jeśli obiekt "A" zawiera obiekt "B":

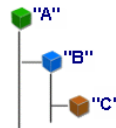
- Obiekt "A" jest obiektem nadrzędnym obiektu "B".
 - Obiekt "B" jest "obiektem podrzędnym" obiektu "A".
-

 Ostrzeżenie:

Związek ten dotyczy tworzenia, a nie dziedziczenia.

Pełna nazwa obiektu

Pełna nazwa obiektu składa się z nazw wszystkich jego obiektów nadrzędnych oraz nazwy samego obiektu. Każdy obiekt oddzielony jest kropką ("."). Jako przykład weźmy następującą strukturę:



Zatem pełna nazwa obiektu "C" ma następującą postać:

<Nazwa obiektu "A">.<Nazwa obiektu "B">.<Nazwa obiektu "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 klika przycisk **Dalej** kreatora, jest wykonywane pierwsze ważne przejście (tzn. takie, które spełnia wszystkie warunki). Jeżeli nie ma ważnego przejścia, przycisk **Dalej** jest zdezaktywowany.
- Jeżeli kreator ma właściwości obowiązkowe (MANDATORY), a nie zostały one wypełnione, nie można zastosować przycisku **Dalej**.
- Jeżeli użytkownik kliknie przycisk **Zakończ** przed zakończeniem wszystkich etapów kreatorów, w przypadku niedokończonych zostaną użyte wartości domyślne.

Model struktury

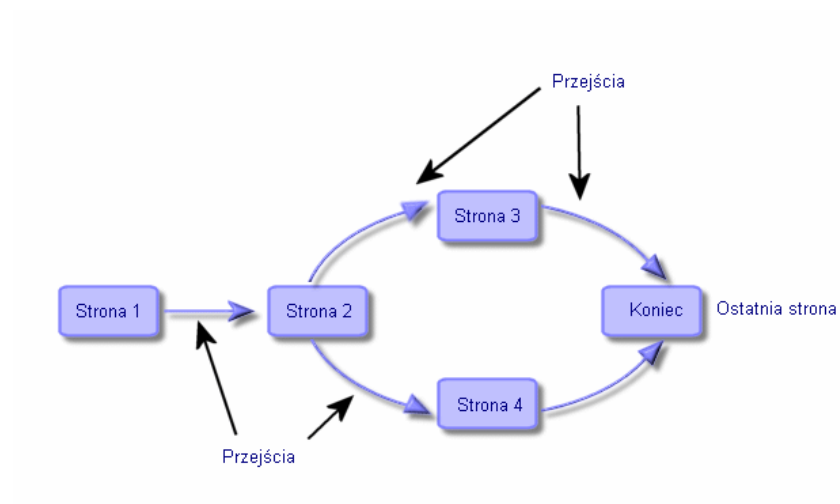
Działanie kreatora AssetCenter następuje po jego uruchomieniu. Z zasady, struktura kreatora opiera się na skrypcie definiującym strukturę, tzn.:

- Skrypt kreatora (a zatem kreator) składa się z węzłów.
- Skrypt kreatora (a zatem kreator) składa się z węzłów.
 - "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.

- Wartość właściwości jest określana za pomocą stałej lub za pomocą skryptu Basic (w takim przypadku wartość jest wynikiem ewaluacji skryptu).

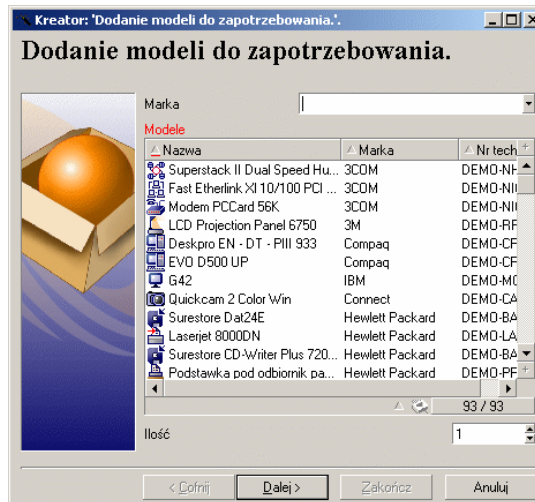
Kreatory składają się ze stron połączonych za pomocą przejść. Przejście z jednej strony na inną jest uwarunkowane wprowadzonymi informacjami i wyborami dokonanymi przez użytkownika. Następujący diagram ilustruje strukturę kreatora:

Rysunek 14.1. Kreatory - model struktury



Model strony kreatora

Strona kreatora jest zorganizowana w następujący sposób:



Uwagi ogólne

Kod kreatora (pole **Skrypt kreatora** (WizardScript) w tabeli **amAction** w zakładce **Kreator** szczegółów działania korzystającego z kreatora) ma postać uporządkowanego tekstu, składającego się z bloków ujętych w klamry ({}). Tekst ten definiuje strukturę kreatora, która ma postać drzewa. 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:

```

; Komentarz nie należący do skryptu
{ <Typ węzła> <Nazwa węzła>
  <Nazwa właściwości> = <Wartość właściwości>
}

```

```

' Komentarz zawarty w skrypcie
...
{ <Nazwa właściwości> =
...
}
{ <Typ węzła> <Nazwa węzła>
<Nazwa właściwości> = <Wartość właściwości>
...
{ <Nazwa właściwości> =
...
}
}
}
}

```

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.

Uwaga:

Uwaga! 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.

 Uwaga:

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 ($A=\{B\}$, $B=\{A\}$):

```
<Nazwa właściwości>=<Skrypt>
```

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:

- Zmiana właściwości następuje:
- Po działaniu użytkownika zmieniającym właściwość lub właściwość zależną.

Definicja stałej jako wartości właściwości

Następujące składnie definiują wartość stałą właściwości:

- Właściwość typu tekstowego:
 - $\langle \text{Nazwa właściwości} \rangle = \langle \text{Tekst} \rangle$
- Właściwość typu logicznego:
 - $\langle \text{Nazwa właściwości} \rangle = \text{TRUE}$
 - $\langle \text{Nazwa właściwości} \rangle = \text{FALSE}$
 - $\langle \text{Nazwa właściwości} \rangle$ (équivalent a $\langle \text{Nazwa właściwości} \rangle = \text{TRUE}$)
- Właściwość typu numerycznego:
 - $\langle \text{Nazwa właściwości} \rangle = 42$
- $\langle \text{Nazwa właściwości} \rangle = \{ \langle \text{Pełna nazwa zmiennej Basic lub właściwości} \rangle \}$

 Uwaga:

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:

```
{<Pełna nazwa właściwości>}
```

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 niejawną. 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

```
<Nazwa właściwości>=<Skrypt>
```

Na przykład:

```
Variable="Nazwa: "& {Name}
```

Poprzedni skrypt jednowierszowy jest równoważny z poniższym skryptem wielowierszowym:

```
{ Variable =
RetVal="Nazwę jest : " & {Name}
}
```

Składnia skryptu wielowierszowego

```
{ <Nazwa właściwości >=
<Skrypt>
}
```

Na przykład:

```
{ LABEL =
IF {Page1.Title}="Wybierz pracownika" THEN
RetVal="Pracownik"
ELSE
RetVal="Usługa"
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 metody jest następująca:

```
{węzeł.węzeł.węzeł[.właściwość][.metoda([arg1[, arg2[?]]])}]
```

gdzie:

- węzeł: nazwa węzła
- właściwość: nazwa właściwości
- metoda: nazwa metody
- arg1, arg2,...: zmienna lub wyrażenie języka Basic (nie powinny zawierać nawiasów klamrowych)

 Uwaga:

W tym przykładzie, znaki "[" i "]" obejmują elementy opcjonalne.

Na przykład, aby pobrać liczbę linii z kontrolki "LISTBOX" na stronie "PAGE1", używamy metody "COUNT" związanej z tym typem kontrolki. Polecenie ma następującą postać:

```
{PAGE1.LISTBOX.VALUES.COUNT()}
```

Właściwość typu tabela

Właściwości typu tabela to właściwości, których wartość jest definiowana zgodnie z następującym formatem:

```
<Kolumna | Kolumna | Kolumna | ...>=<Identyfikator wiersza>, <Kolumna | Kolumna | Kolumna | ...>=<Identyfikator wiersza>, ...
```

Wartości tych właściwości można przeglądać w postaci tabeli:

		Kolumna 1	Kolumna 2	Kolumna 3
Numer wiersza: 1	Identyfikator wiersza (np.: 18)	Komórka (1,1)	Komórka (2,1)	Komórka (3,1)
Numer wiersza: 2	Identyfikator wiersza (np.: 29)	Komórka (1,2)	Komórka (2,2)	Komórka (3,2)
Numer wiersza: 3	Identyfikator wiersza (np.: 78)	Komórka (1,3)	Komórka (2,3)	Komórka (3,3)
Itp.	Itp.	Itp.	Itp.	Itp.

 Uwaga:

Identyfikator jest typu 'Tekst'

Przykład

Rozważmy właściwość "VALUES" węzła "LISTBOX" 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	Colombo	Gerard
2	64	Lübeck	Alexander
3	24	Daquin	William

Używanie zmiennych globalnych CurrentTable i CurrentSelection

Zawartość tych zmiennych można pobrać za pomocą następującej składni:

```
[CurrentTable]
[CurrentSelection]
```

Następująca tabela przedstawia cechy tych dwóch zmiennych:

Nazwa zmiennej	Opis zmiennej	Uwagi
CurrentTable	Zawiera nazwę SQL tabeli aktywnej w momencie uruchomienia kreatora. Jeżeli nie ma aktywnej tabeli, zawiera pusty ciąg. Zmienna typu "String".	Zmienna ta jest automatycznie wypełniana przez AssetCenter. Użytkownik nie może spowodować zmiany wartości.
CurrentSelection	Zawiera listę wewnętrznych identyfikatorów rekordów, rozdzielonych przecinkiem, wybranych w momencie uruchomienia kreatora. Zmienna typu "String".	Zmienna ta jest automatycznie wypełniana przez AssetCenter. Zawiera pusty ciąg, jeżeli nie zostały wybrane żadne elementy, lub nie została podana żadna tabela. Użytkownik nie może spowodować zmiany wartości.

Sekwencjonowanie kreatorów

Po wykonaniu kreatora można go użyć do zainicjowania wykonania innego kreatora i przekazania parametrów (zmiennych) do tego kreatora. Metoda ta jest nazywana sekwencjonowaniem kreatorów.

Wykonanie

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()
- AmDecrementLogLevel()
- AmExecTransition()
- AmLog()
- AmMsgBox()
- AmPagePath()
- AmProgress()
- AmRefreshProperties
- AmSetProperty
- AmUpdateDetail

- AmValueOf
- |AmWizChain

! Ostrzeżenie:

Przy wywoływaniu funkcji języka Basic ze skryptu kreatora, zawsze należy przypisywać do zmiennej wartość zwracaną przez funkcję. W przeciwnym razie kompilator języka Basic zwróci błąd. Zatem następujący przykład nie zostanie skompilowany:

```
AmGetFieldLongValue(hRecord, "IUserId", {IEmplDeptId})
```

Prawidłowy skrypt ma następującą postać:

```
Dim IValue as Long
IValue=AmGetFieldLongValue(hRecord, "IUserId", {IEmplDeptId})
```

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:

```
' Blok głównych właściwości węzła 'root'
NAME=...
IMAGE=...
...
' Definiowanie węzłów podrzędnych węzła "root"
{ FINISH
...
}
{ PAGE
...
}
```

```

}
{ TRANSITION
...
}

```

Właściwości węzła 'Root'

W następujących tabelach podano wszystkie właściwości logiczne i fizyczne, które mogą być zdefiniowane w węzle głównym:

Tabela 14.2. 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 Przewodnik"	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" (ciąg znaków).	TITLE="Kreator Przewodnik"	Usilnie zalecamy zdefiniowanie wartości dla tej właściwości.

Nazwa	Opis właściwości	Przykład	Uwagi
właściwości=Wartość			
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. Skrypt ma charakter deklaracji i nie jest wykonywany. Aby wykonać skrypt przy uruchomieniu, użyj węzła "START". Właściwość typu "Script".	{GLOBAL=Dim Filter As String}	
COMMON	Zawiera funkcje Basic automatycznie dodawane do wszystkich kreatorów. Właściwość odpowiada plikowi gbbase.wiz (tylko do odczytu), dodanemu do bazy danych w momencie jej tworzenia.		

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 NAME określa, gdzie w sekcji pliku .ini będą przechowywane wartości. Właściwość typu logicznego.	SERIALIZE=TRUE	Domyślnie właściwość ta ma wartość "FALSE".
MODAL=<TRUE FALSE>	Określa modalność (=TRUE) lub nie (=FALSE) kreatora		

Tabela 14.3. Właściwości fizyczne węzła 'Root'

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" (ciąg znaków).	IMAGE="Kolor.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 głębokość koloru ekranu wynosi 16.

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
WIDTH=<Width>	Definiuje domyślną szerokość ("<Width>") okna kreatora. Jest ona wyrażona w jednostce twip. Właściwość typu "Long".	WIDTH=6000	
HEIGHT=<Height>	Definiuje domyślną wysokość ("<Height>") okna kreatora. Jest ona wyrażona w jednostce twip. Właściwość typu "Long".	HEIGHT=5000	
MINWIDTH=<MinWidth>	Określa minimalną szerokość okna kreatora. Wartość wyrażona w twips .		
MINHEIGHT=<MinHeight>	Określa minimalną wysokość okna kreatora. Wartość wyrażona w twips .		

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
CTRLHEIGHT=<CtrlHeight>	Określa wysokość kontrolki o stałym rozmiarze pionowym (na przykład kontrolka TEXTBOX). Wartość odstępu wyrażona w twips .		
LABELSPACING=<labelSpacing>	Określa odstęp między tytułem kontrolki a samą kontrolką, w przypadku gdy tytuł znajduje się nad kontrolką. Wartość odstępu wyrażona w twips .		
CTRLSPACING=<CtrlSpacing>	Określa pionowy odstęp między dwoma kontrolkami. Wartość odstępu wyrażona w twips .		

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
IMGBORDER=<Width>	Określa odstęp poziomy między obrazem kreatora a jego kontrolkami. Wartość wyrażona w twips .		
NAVIGATION=<TRUE FALSE>	Wyświetla (=TRUE) lub nie (=FALSE) pasek narzędzi w oknie kreatora zawierający przyciski Dalej , Anuluj .		
CONFIRMCANCEL=<TRUE FALSE>	Określa (=TRUE) lub nie (=FALSE) wiadomość potwierdzającą anulację.		
DEFAULTONNEXT=<TRUE FALSE>	Domyślnie wybiera (=TRUE) przycisk Dalej . Jeżeli DEFAULTONNEXT=FALSE wówczas wybieranym przyciskiem jest Zakończ .		

Węzły podrzędne węzła 'Root'

Typy węzłów podrzędnych, które można zdefiniować dla węzła głównego, podane są w następującej tabeli. Każdy typ węzła stanowi "Obiekt".

Tabela 14.4. Węzły podrzędne węzła "Root" (głównego)

Typ węzła	Opis
PAGE	Opisuje stronę kreatora.
FINISH	Opisuje końcowe przejście z ostatniej strony kreatora (do zakończenia). Ten węzeł typu "Transition" (przejście) nie ma właściwości "FROM" (z) i "TO" (do).
START	Zawiera na przykład skrypt wykonywany przy uruchomieniu kreatora (za pomocą właściwości "DO" -wykonaj) oraz nazwę strony początkowej kreatora (właściwość "TO" -do).
PARAMS	Umożliwia przeniesienie parametrów z jednego kreatora do drugiego, jeżeli właściwość CHAIN węzła podrzędnego FINISH została wypełniona.
TIMER	Umożliwia powiązanie datownika ze stroną kreatora.

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:

```
' Określenie strony
{ Page <Nazwa strony>
' Blok właściwości węzła "Page"
IMAGE=...
TITLE=...
' Definiowanie węzłów podrzędnych węzła "Page"
{ TRANSITION
```

```

...
}
{ <Typ kontroli> <Nazwa kontroli>
...
}
...
}

```

Właściwości węzła 'Page'

W poniższych tabelach podano wszystkie właściwości, które można zdefiniować w węzle "Strona":

Tabela 14.5. Właściwości logiczne węzła "Page" (strona)

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
TITLE="<Tytuł strony>"	Określa tytuł strony. Jest on wyświetlany czcionką wytluszczoną u góry strony. Właściwość typu "String" (ciąg znaków).	TITLE="Przenieś"	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 otwarceniu strony za pomocą przycisku Dalej lub Cofnij . Właściwość typu "Script".	{ONENTER = AmMsgBox ("Hello") }	

Tabela 14.6. 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 zdefiniujesz wartość właściwości "IMAGE16", jest ona używana zamiast właściwości "IMAGE" jeśli głębokość koloru ekranu wynosi 16.

Węzły podrzędne węzła 'Page'

Dla węzła "Page" można zdefiniować dwa typy węzłów podrzędnych:

Tabela 14.7. Węzły podrzędne węzła "Page" (strona)

Typ węzła / "Obiekt"	Opis
<Typ kontrolki> <Nazwa kontrolki>	Określa kontrolkę wyświetlaną na bieżącej stronie.
TRANSITION <Nazwa przejścia>	Określa przejście pomiędzy bieżącą stroną, a inną stroną kreatora.

Typ węzła / "Obiekt"	Opis
TIMER	Umożliwia powiązanie datownika ze stroną kreatora.

Definicja węzła 'Transition'

Węzeł "Transition" (przejście) opisuje przejście pomiędzy dwiema stronami kreatora. Składa się on wyłącznie z bloku właściwości.

 Uwaga:

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:

```
' Określenie przejścia
{ TRANSITION0 <Nazwa przejścia>
' Blok właściwości węzła "transition"
FROM=...
TO=...
CONDITION=...
}
```

Właściwości węzła 'Transition'

W poniższej tabeli podano wszystkie właściwości, które można zdefiniować w węźle "Transition":

Tabela 14.8. Logiczne właściwości węzła "Transition"

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
FROM=<Nazwa strony wyjściowej>"	Określa nazwę wyjściowej strony 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 nazwę docelowej strony przejścia. Właściwość typu "String" (ciąg znaków).	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 "Skrypt" zwracająca wartość logiczną.	CONDITION={Comment}'uzytkownik"	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" nie mającego właściwości "FROM" i "TO". Poza tym składnia i właściwości węzła "FINISH" są takie same, jak węzła "Transition".

Właściwość CHAIN (dołącz), specyficzna dla węzła końcowego, pozwala zainicjować wykonanie następnego kreatora.

Tabela 14.9. Właściwości logiczne węzła "Finish"

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
CHAIN=<nazwa SQL kreatora do uruchomienia>	Określa nazwę SQL kreatora do uruchomienia przy zakończeniu bieżącego kreatora. Jeżeli ta właściwość nie została wypełniona, żaden kreator nie zostanie wykonany. Właściwość typu "String"	CHAIN= "Przenieś"	

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
CONDITION=<Skrypt>	Określa warunek zatwierdzający przycisk Zakończ . Właściwość typu "Script".		
DO=<Skrypt>	Określa skrypt do wykonania przy kończeniu kreatora. Właściwość typu "Skrypt" zwracająca wartość logiczną.		

**Uwaga:**

Węzeł PARAMS pozwala przekazywać parametry do następnego kreatora.

Tabela 14.10. Właściwość fizyczna węzła 'Finish'

Nazwa właściwości=Wartość	Opis właściwości	Przykład
SUMMARY=<TRUE FALSE>	Wyświetla (=TRUE) lub nie (=FALSE) stronę podsumowania przy wykonywaniu kreatora. Funkcje amLog i amProgress umożliwiają wypełnienie tej strony.	
SHOWPROGRESSBAR=<TRUE FALSE>	Wyświetla (=TRUE) lub nie (=FALSE) pasek zaawansowania na stronie podsumowania.	
SHOWLOGLIST=<TRUE FALSE>	Wyświetla (=TRUE) lub nie (=FALSE) dziennik zaawansowania na stronie podsumowania.	
LABEL="Nazwa"	Wyświetla nazwę strony podsumowania.	

ISHTML=<TRUE FALSE>	Określa charakter tekstu nazwy. Właściwość typu logicznego.
TITLE="Tytuł"	Określa tytuł strony podsumowania. Domyślnym tytułem strony podsumowania jest tytuł węzła głównego.

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".

Tabela 14.11. Właściwość logiczna węzła "Start"

Nazwa właściwości=Wartość	Opis właściwości	Przykład
DO=<Skrypt>	Określa skrypt do wykonania przy uruchomieniu. Właściwość typu "Skrypt" zwracająca wartość logiczną.	
TO="<Nazwa strony początkowej>"	Określa nazwę pierwszej wyświetlanej strony. Właściwość typu "String" (ciąg znaków).	

 **Uwaga:**

Jeżeli węzeł nie istnieje, kreator otwiera się na pierwszej stronie.

Definicja węzła 'Timer'

Węzeł "Timer" umożliwia wykonanie zadania w regularnych odstępach czasu.

Tabela 14.12. Właściwość logiczna węzła "Timer"

Nazwa właściwości=Wartość	Opis właściwości	Przykład
AUTO= <TRUE FALSE>	Określa, czy datownik jest automatycznie uruchamiany przy wyświetlaniu strony. Właściwość tę można użyć do zatrzymania lub uruchomienia datownika. Właściwość typu logicznego.	
ENABLED= <TRUE FALSE>	Określa, czy datownik jest aktywny (=TRUE) lub nie (=FALSE). Właściwość typu logicznego.	
INTERVAL=termin	Określa odstęp czasu (termin) pomiędzy dwoma wykonaniami datownika. Czas wyrażony w milisekundach.	
TIMER="Skrypt"	Określa, co będzie wykonywane za każdym razem, gdy upłynie termin ("INTERVAL") datownika. Właściwość typu skrypt.	

VALUE=tickcount	<p>Liczba upłynięć terminów. Każda właściwość, która zależy od terminu zostanie automatycznie poddana ponownej ewaluacji w regularnych odstępach czasu.</p> <p>Wartość ta (tickcount) jest powiększana automatycznie.</p>
-----------------	---

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.

Tabela 14.13. Właściwość logiczna węzła typu Long lub String

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
VALUE=<Wartość>	<p>Określa wartość zmiennej, której nazwa jest identyczna z nazwą węzła.</p> <p>Właściwość typu "Long" dla węzła typu Long lub "String" dla węzła typu String.</p>	VALUE=12	

 Uwaga:

Węzły typu Long i String można definiować w dowolnym węźle kreatora. Nie mają one węzłów podrzędnych.

Definicja węzła 'Control'

Kontrolka na stronie służy do interakcji z użytkownikiem. Na danej stronie można zdefiniować dowolną ilość kontroltek. AssetCenter w pełni kontroluje organizację kontroltek na stronie. Nie trzeba określać pozycji każdego definiowanej kontrolki.

Węzły typu "Control" składają się wyłącznie z bloku właściwości dotyczących danej kontrolki.

Ogólna składnia węzła typu "Control"

Ogólna składnia węzła typu "Control" jest następująca:

```
' Określenie kontrolki
{ <Typ kontrolki> <Nazwa kontrolki>
' Właściwości kontrolki
...
}
```

Typy kontroltek i ich właściwości

Wszystkie kontrolki mają wspólne właściwości. Istnieją jednakże właściwości specyficzne dla poszczególnych kontroltek.

Wspólne właściwości kontroltek

W poniższej tabeli podano opcjonalne właściwości odnoszące się do wszystkich kontroltek:

Tabela 14.14. Właściwości logiczne wspólne dla wszystkich kontroltek

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
MANDATORY= <TRUE FALSE>	Zmusza użytkownika do wypełnienia kontrolki dla zachowania poprawności przejścia. Właściwość ta jest ignorowana, jeżeli kontrolka: <ul style="list-style-type: none"> nie jest widoczna jest tylko do odczytu została zdezaktywowana 	MANDATORY=TRUE	Właściwość ta nie jest dostępna dla kontroltek typu "CHECKBOX" (pole wyboru) i "LABEL" (etykieta)
VALUE=<Wartość>	Określa domyślną wartość tworzonej kontrolki. <Wartość> zależy od danej kontrolki. Typ tej właściwości zależy od typu kontrolki (logiczny, tekstowy, itd.).	Na przykład, jeśli jest to kontrolka "CHECKBOX", <Wartość> może być następująca: "TRUE" (prawda) lub "FALSE" (fałsz).	

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
PERMANENT=<TRUE FALSE>	<p>Podczas przejścia z jednej strony kreatora na drugą, kontrolki są usuwane.</p> <p>Określa, czy kontrolka jest zachowywana podczas przejścia z jednej strony na drugą (=TRUE), czy usuwana (=FALSE).</p> <p>Właściwość typu logicznego.</p>		
SERIALIZE=<TRUE FALSE>	<p>Ta właściwość pozwala na serializację kreatora na poziomie kontrolki. Jeżeli serializacja węzła głównego jest aktywna (=TRUE), można ją zdezaktywować na poziomie kontrolki.</p>		<p>Domyślnie właściwość ta ma wartość właściwości SERIALIZE węzła głównego.</p>
HELP="pomoc"	<p>Ta właściwość pozwala na dodanie tekstu pomocy o formacie HTML w kontrolce kreatora. Pomoc jest dostępna za pomocą klawiszy Shift+F1.</p>		

Tabela 14.15. Właściwości fizyczne wspólne dla wszystkich kontroltek

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
<code>VISIBLE=<TRUE FALSE></code>	Określa, czy kontrolka jest widoczna (=TRUE) lub nie (=FALSE). Właściwość typu logicznego.	<code>Label1.Visible=TRUE</code>	
<code>ENABLED = <TRUE FALSE></code>	Określa, czy kontrolka jest aktywna (=TRUE) lub nie (=FALSE). Właściwość typu logicznego.	<code>Choice1.Enabled=FALSE</code>	
<code>READONLY = <TRUE FALSE></code>	Określa, czy kontrolka jest tylko do odczytu (=TRUE) i nie może być edytowany przez użytkownika; lub edytowalny (=FALSE)	<code>READONLY=TRUE</code>	
<code>LABEL = "<Texte du label>"</code>	Definiuje opcjonalny tekst, wyświetlany nad kontrolką lub po jej lewej stronie. Właściwość typu "String" (ciąg znaków).	<code>Choice1.Label="Wybierz osobę"</code>	Ta kontrolka obsługuje znaczniki HTML.

Nazwa właściwość=Wartość	Opis właściwości	Przykład	Uwagi
LABELLEFT	<p>Ta właściwość pozwala na przesunięcie nazwy kontrolki w lewo.</p> <p>Stosowanie tej właściwości wymaga wypełnienia właściwości XOFFSET.</p> <p>Właściwość typu logicznego.</p>		
XOFFSET	<p>Określa odstęp zarezerwowany dla nazwy kontrolki, jeżeli została ona umieszczona po lewej stronie kontrolki za pomocą właściwości LABELLEFT.</p> <p>Właściwość typu twip.</p>		
ISHTML	<p>Określa charakter tekstu nazwy.</p> <p>Domyślnie jest to HTML.</p> <p>Właściwość typu logicznego.</p>		Ta nazwa obsługuje znaczniki HTML.
INDENT	<p>Ta właściwość pozwala na przesunięcie kontrolki i jej nazwy w prawo.</p> <p>Właściwość typu twip.</p>		

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
YOFFSET	Określa 'miejsce przed' przed kontrolką i jego nazwą. Właściwość typu twip .		
YOFFSET2	Określa 'miejsce po' po kontrolce i jej nazwie. Właściwość typu twip .		

Kontrolka CheckBox

Kontrolka "CHECKBOX" definiuje pole wyboru.

Właściwości

Oprócz opcjonalnych właściwości, wspólnych dla wszystkich kontroltek, kontrolka "CHECKBOX" przyjmuje następującą właściwość:

Tabela 14.16. Właściwość kontrolki "CHECKBOX" (pole wyboru)

Nazwa właściwości=Wartość	Opis właściwości	Przykład
CAPTION="<Tekst>"	Definiuje tekst pola wyboru. Tekst nie może mieć formatu HTML i musi zawierać jeden wiersz. Właściwość typu "String" (ciąg znaków).	TEXT="Określ według nazwy"

Kontrolka ComboBox

Kontrolka "COMBOBOX" (lista kombinowana) definiuje pojedynczy wybór z wyliczenia (listy pozycji) predefiniowanych wartości.

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich kontroltek, kontrolka "COMBOBOX" przyjmuje następującą właściwość:

Tabela 14.17. Właściwości fizyczne kontrolki "COMBOBOX" (lista kombinowana)

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
VALUES="<Nazwa=Wartość, Nazwa=Wartość, Nazwa=Wartość,...>"	Definiuje pary wartości ("Nazwa"="Wartość") dla kontrolki listy kombinowanej. "Nazwa" określa tekst, który jest wyświetlany w kontrolce, "Wartość" - przypisywaną wartość, jeśli dana nazwa zostanie wybrana przez użytkownika. Właściwość typu "String" (ciąg znaków).	VALUES="Tabela złowazbikozjovk"	Jeśli parametr "Value" zostanie pominięty, AssetCenter automatycznie przypisze wartość. Na przykład, wyrażenie: Przykład VALUES = "A,B,C" jest równoważne VALUES = "A=1,B=2,C=3"

Kontrolka OptionButtons

Kontrolka "OPTIONBUTTONS" (przyciski opcji) definiuje grupę przycisków opcji (przycisków "radiowych").

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich kontroltek, kontrolka "OPTIONBUTTONS" przyjmuje następujące właściwości:

Tabela 14.18. Właściwości fizycznie kontroltek "OPTIONBUTTONS" (przyciski opcji)

Nazwa właściwości=Wartość	Opis właściwości	Przykład
VALUES="<Tytuł=Wartość, Tytuł=Wartość, Tytuł=Wartość,...>"	Określa pary wartości ("Tytuł"="Wartość") dla kontrolki "OPTIONBUTTONS". Parametr "Nazwa" określa tekst przycisku opcji, parametr "Wartość" określa wartość przypisaną do kontrolki jeśli użytkownik wybierze dany przycisk opcji. Właściwość typu "String" (ciąg znaków).	VALUES="Tabela zadów=zadUzytkownik=uzytkownik"
BORDER=<TRUE FALSE>	Określa, czy wokół przycisków opcji rysowana jest ramka (=TRUE), czy nie (=FALSE) Jeżeli grupa przycisków znajduje się w ramce, tekst jest dodawany na brzegu ramki. Tekst nie może mieć formatu HTML ani kilku wierszy. Właściwość typu logicznego.	BORDER= TRUE

Kontrolka ListBox

Kontrolka "LISTBOX" (pole listy) definiuje listę obiektów do wyboru. Kontrolki "LISTBOX" mogą być kontrolki wielokolumnowymi.

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich kontrolki, kontrolka "LISTBOX" przyjmuje następujące właściwości:

Tabela 14.19. Właściwości fizyczne kontrolek "LISTBOX" (pole listy)

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
LISTHEIGHT = <Procent>	Określa rozmiar kontrolki "LISTBOX" w odniesieniu do innych kontrolki "LISTBOX" w całym kreatorze. Właściwość typu "Long".	LISTHEIGHT=50	Jeśli istnieją dwie kontrolki "LISTBOX" o wartościach tej właściwości odpowiednio "10" i "20", druga kontrolka będzie dwukrotnie większa od pierwszej.
MULTISEL = <TRUE FALSE>	Określa, czy kontrolka obsługuje zaznaczanie wielu pozycji (=TRUE), czy nie (=FALSE). Właściwość typu logicznego.	MULTISEL=TRUE	
COLTITLE = '<Column Column Column>'	Określa nazwę kolumn listy. Zastępuje parametr "Column" tytułem kolumny. Właściwość typu "String" (ciąg znaków).	COLTITLE = "Nazwisko Imię"	
COLWIDTH = '<Width Width Width>'	Określa szerokość kolumny proporcjonalnie do całkowitej szerokości kontrolki. Właściwość typu "String" (ciąg znaków).	COLWIDTH = "50 50"	

Nazwa	Opis	Przykład	Uwagi
właściwość=Wartość VALUES = "<Tekst Tekst ...= Wartość, Tekst Tekst ...= Wartość,...>"	Definiuje pary wartości ("Tekst Tekst ..."="Wartość") dla kontrolki "LISTBOX". Parametr "Tekst Tekst .." określa tekst wyświetlany w każdej kolumnie danego wiersza kontrolki "LISTBOX", Parametr "Wartość" określa wartość, jaką przybiera kontrolka jeśli użytkownik wybierze dany wiersz. Właściwość typu "String" (ciąg znaków).	VALUES="Tabela zasobów=zasób, Użytkownik=użytkownik"	Jeśli parametr "Value" zostanie pominięty, AssetCenter automatycznie przypisze wartość. Na przykład, wyrażenie: VALUES="A,B,C" jest opiewane jako VALUES="A,B,C" Przykład Właściwość tę można wypełniać 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".
EDITABLE="<0 1>"	Określa, czy można edytować tekst kolumn. Właściwość typu "String".	EDITABLE="0 1"	
TABLE="<nom de la table>"	Określa kontekst stosowania tytułu kolumn, jeżeli został on określony przez właściwość "COLNAME".	TABLE="amEmplDept"	

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
COLNAME="nazwa nazwa"	Określa tytuł i właściwości kolumn listy za pomocą nazw SQL pól. Należy wypełnić właściwość "TABLE". W tytule kolumny zastąp "nazwę" nazwą SQL pola. Jeżeli tytuł kolumny został wypełniony za pomocą właściwości "COLTITLE", ma ona pierwszeństwo przez właściwością "COLNAME", ale zachowuje typ pola SQL (tekst, data, itd.).	COLNAME="nazwa nazwa"	
MULTISEL = <TRUE FALSE>	W przypadku listy określa stosowanie wielokrotnego zaznaczenia. Właściwość typu logicznego.	MULTISEL=1	

Tabela 14.20. Metody kontrolki "LISTBOX"

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Komentarz
VALUES (i)	Zwraca zawartość linii (i)	a b c	
VALUES.COUNT()	Wylicza liczbę linii we właściwości "VALUES".	retval = {listbox1.values.count()}	

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Komentarz
VALUES.CELL(h,v)	Zwraca zawartość komórki określonej przez współrzędne (poziome, pionowe).	VALUES.CELL(2,4)	
VALUES.COLUMN(i)	Zwraca zawartość (wartość) kolumny (i). Jeżeli i=0 lub brak wartości, wówczas polecenia zwraca identyfikatory (ID) kolumny.	VALUES.COLUMN(1)	
VALUES.SORT(iCol, bAsc)	Sortuje kolumnę (i) rosnąco lub malejąco (bAsc=1 lub bAsc=0)	<pre>{ LISTBOX lb VALUES = "pierwszy,drugi,trzeci" } { COMMANDBUTTON btn { CLICK = RetVal = {lb.Values.Sort(1)} } }</pre>	
VALUES (i,0) ID(i)	Zwraca zawartość identyfikatora wiersza (i).		

Tabela 14.21. Obowiązkowa logiczna właściwość kontrolki "LISTBOX" (lista)

Nazwa właściwości=Wartość	Opis właściwości	Przykład
TABLE=<Nazwa tabeli>	Nazwa tabeli używanej do pobierania nazw kolumn. Właściwość typu "String" (ciąg znaków).	TABLE= amAsset

Nazwa właściwości=Wartość	Opis właściwości	Przykład
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. Kontrolka 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" (ciąg znaków).	COLNAME="Nazwisko Imię"

Kontrolka Label

Kontrolka "LABEL" (etykieta) określa etykietę. Kontrolka ten ma następującą właściwość:

Tabela 14.22. Właściwości fizyczne kontrolki "LABEL"

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
CAPTION=<Tekst>	Zawiera tekst wyświetlany w etykiecie.	CAPTION="Wybierz lokalizację"	

Kontrolka ProgressBar

Kontrolka "PROGRESSBAR" definiuje wskaźnik zaawansowania.

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich kontrolki, kontrolka "PROGRESSBAR" przyjmuje następujące właściwości:

Tabela 14.23. Właściwości fizyczne kontrolka "PROGRESSBAR"

Nazwa właściwości=Wartość	Opis właściwości	Przykład
MAXVALUE=<Wartość maksymalna>	Określa wartość maksymalną odpowiadającą 100% wskaźnika zaawansowania. Właściwość "VALUE" wskazuje bieżącą wartość kontrolka. Właściwość typu "Long".	MAXVALUE=200

Kontrolka CommandButton

Kontrolka "COMMANDBUTTON" definiuje przycisk polecenia inicjujący działanie.

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich kontrolki, kontrolka "COMMANDBUTTON" przyjmuje następujące właściwości:

Tabela 14.24. Właściwości fizyczne kontrolki "COMMANDBUTTON"

Nazwa właściwości=Wartość	Opis właściwości	Przykład
WIDTH=<Width>	Określa szerokość przycisku w jednostkach twip. Właściwość typu "Long".	WIDTH=250

Nazwa właściwości=Wartość	Opis właściwości	Przykład
HEIGHT=<Height>	Określa wysokość przycisku w jednostkach twip. Właściwość typu "Long".	HEIGHT=125
CAPTION=<Tekst>	Określa tekst (nie w formacie HTML) wyświetlany na przycisku. Właściwość typu "String" (ciąg znaków).	CAPTION="Start"
CLICK=<skrypt Basic>	Określa skrypt języka Basic wykonywany po kliknięciu przycisku przez użytkownika.	

Kontrolka DBListBox

Kontrolka "DBLISTBOX" określa listę rekordów do wyboru z bazy danych. Może mieć on postać kontrolki wielokolumnowego. Lista wyświetlana w kontrolce jest wynikiem częściowego zapytania AQL (używany jest tylko warunek WHERE) bazy danych AssetCenter.

Uwaga:

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 kontrolkę typu "LISTBOX" (pole listy).

Właściwości

Oprócz opcjonalnych kontrolki, wspólnych dla wszystkich właściwości, kontrolka "DBLISTBOX" przyjmuje następujące właściwości:

Tabela 14.25. Właściwości fizyczne kontrolka "DBLISTBOX"

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
TABLE="<Nazwa SQL tabeli>"	Określa tabelę, do której kierowane jest zapytanie. Właściwość typu "String" (ciąg znaków).	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" (ciąg znaków).	COLNAME = "Name FirstName"	
COLWIDTH = "<Width Width Width >"	Określa szerokość kolumn bazy danych, proporcjonalnie do ogólnego rozmiaru kontrolki "DBLISTBOX". Właściwość typu "String" (ciąg znaków).	COLWIDTH="40 60"	
LISTHEIGHT = <Procent>	Określa rozmiar kontrolki "DBLISTBOX" względem innych kontrolki "DBLISTBOX" w całości kreatora. Właściwość typu "Long".	LISTHEIGHT=50	Jeśli istnieją dwa kontrolki "DBLISTBOX" o wartościach tej właściwości odpowiednio "10" i "20", druga kontrolka będzie dwukrotnie wyższy względem pierwszego.

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
TREE=<TRUE FALSE>	Wyświetla dane w postaci drzewa (=TRUE) lub nie (=FALSE). Właściwość typu logicznego.	TREE=TRUE	Właściwość ta ma domyślnie wartość "FALSE"
MULTISEL = <TRUE FALSE>	Określa, czy kontrolka obsługuje zaznaczanie wielu pozycji (=TRUE), czy nie (=FALSE). Właściwość typu logicznego.	MULTISEL=TRUE	
DBLCLICK = <TRUE FALSE>	Jeśli właściwość ta jest ustawiona jako TRUE, w przypadku podwójnego kliknięcia, AssetCenter będzie symulować kliknięcie przycisku Dalej bieżącej strony.	DBLCLICK=FALSE	
FILTER = "<Condition>"	Określa warunek "WHERE" języka AQL filtrujący rekordy przetwarzane przez zapytanie. Właściwość typu "String" (ciąg znaków).	FILTER = "User.ImplDeptId = 'Colombo, Gérard'"	

Nazwa właściwość=Wartość	Opis właściwości	Przykład	Uwagi
MAXSEL = <TRUE FALSE>	Określa możliwość pobrania ponad 99 elementów (=TRUE) lub (=FALSE). Domyślnie zaznaczenie jest ograniczone (=TRUE) 99 elementów.		
VALUES.SORT(iCol, bAsc)	Sortuje kolumnę (iCol) rosnąco lub malejąco (bAsc=1 lub bAsc=0)	VALUES.SORT(2, 0)	
[Value.]ISELECTION()	Jeżeli zaznaczenie zawiera wartości różne od 0, co oznacza, że użytkownik nie zaznaczył żadnej pozycji listy, metoda ta zwraca wartość różną od zera.	MANDATORY = not {dblistbox1.IsSelection()} (przejsicie nie jest możliwe jeżeli użytkownik nic nie zaznaczył na liście)	
TABLELABEL(iNameType)	Ta metoda zwraca tytuł tabeli, dla której została zdefiniowana kontrolka. Typy tytułów (iNameType) są następujące: <ul style="list-style-type: none"> • 1 - Nazwa systemowa • 2 - Nazwa SQL • 3 - Nazwa domyślna • 4 - Opis (tekst pomocy) Należy wypełnić właściwość "TABLE".	RetVal = {dblistbox1.table.label(2)}	

Kontrolka DBQueryBuilder

Kontrolka "DBQUERYBOX" definiuje listę rekordów do wyboru w bazie danych AssetCenter. Kontrolka ten może mieć postać wielokolumnową. Lista wyświetlana w kontrolce jest wynikiem zapytania AQL kierowanego do bazy danych AssetCenter.

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich kontroltek, kontrolka "DBQUERYBOX" przyjmuje następujące właściwości:

Tabela 14.26. Właściwości fizyczne kontrolki "DBQUERYBOX"

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
QUERY="<Pełne zapytanie AQL>"	Definiuje zapytanie AQL zwracające informacje wyświetlane w kontrolce "DBLIST". Właściwość typu "String" (ciąg znaków).	QUERY="SELECT Name, FirstName FROM amEmpIDept WHERE Location=Biurowiec Polaris"	
COLTITLE="<Kolumna Kolumna ..>"	Definiuje tytuły kolumn listy. Właściwość typu "String" (ciąg znaków).	COLTITLE = "Name FirstName"	
COLWIDTH="<Szerokość Szerokość ..>"	Definiuje rozmiar kolumn listy jako procent ogólnego rozmiaru kontrolki. Właściwość typu "String" (ciąg znaków).	COLWIDTH = "50 50"	


Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
LISTHEIGHT=<Procent>	Definiuje rozmiar kontrolki "DBQUERYBOX" proporcjonalnie do innych kontrolki "DBQUERYBOX" w całym kreatorze. Właściwość typu "Long".	LISTHEIGHT=50	Jeśli istnieją dwie kontrolki "DBQUERYBOX" o wartościach tej właściwości odpowiednio "10" i "20", druga kontrolka będzie dwukrotnie wyższy od pierwszego.
TREE=<TRUE FALSE>	Wyświetla dane w postaci drzewa (=TRUE) lub nie (=FALSE). Właściwość typu logicznego.	TREE=TRUE	Właściwość ta ma domyślnie wartość "FALSE"
MAXSEL = <TRUE FALSE>	Określa możliwość pobrania ponad 99 elementów (=TRUE) lub (=FALSE). Domyślnie zaznaczenie jest ograniczone (=TRUE) 99 elementów.		

Nazwa właściwości=Wartość	Opis właściwości	Przykład	Uwagi
MULTISEL=<TRUE FALSE>	Określa, czy kontrolka obsługuje zaznaczanie wielu pozycji (=TRUE), czy nie (=FALSE). Właściwość typu logicznego.	MULTISEL=TRUE	
DBLCLICK=<TRUE FALSE>	Jeśli właściwość ta jest ustawiona jako TRUE, w przypadku podwójnego kliknięcia, AssetCenter będzie symulować kliknięcie przycisku Dalej bieżącej strony. Właściwość typu logicznego.	DBLCLICK=FALSE	
[Value.]ISSELECTION()	Jeżeli zaznaczenie zawiera wartości różne od 0, co oznacza, że użytkownik nie zaznaczył żadnej pozycji listy, metoda ta zwraca wartość różną od zera.		

Kontrolka DBEdit

Kontrolka "DBEDIT" tworzy kontrolkę identyczną z kontrolką do wypełniania pola w bazie danych AssetCenter. Postać kontrolki różni się w zależności od typu pola (data, walutowe, itp.).

 Uwaga:

Przycisk z lupą  w tej kontrolce pozwala wybierać wartości znajdujące się w bazie danych, można jednak wprowadzać inne wartości.

W przypadku tej kontrolki, właściwość "VALUE" jest typu "Variant" (typ zależny od kontrolki).

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich kontroltek, kontrolka "DBEDIT" przyjmuje następujące właściwości:

Tabela 14.27. Właściwości fizyczne kontrolki "DBEDIT" w trybie "Normal"

Nazwa właściwości=Wartość	Opis właściwości	Przykład
TABLE="<Nazwa SQL tabeli>"	Nazwa SQL tabeli wyjściowej. Właściwość typu "String" (ciąg znaków).	TABLE="amAsset"
FIELD="<Nazwa SQL pola>"	Nazwa AQL pola używanego dla kontrolki. Właściwość typu "String" (ciąg znaków).	FIELD="seAcquMethod"

Nazwa właściwości=Wartość	Opis właściwości	Przykład
TABLE.LABEL([iNameType])	<p>Ta metoda zwraca tytuł danej tabeli.</p> <p>Typy tytułów (iNameType) są następujące:</p> <ul style="list-style-type: none"> • 1 - Nazwa systemowa • 2 - Nazwa SQL • 3 - Nazwa domyślna • 4 - Opis (tekst pomocy) <p>Należy wypełnić właściwości "TABLE" i "NAME".</p>	<p>Patrz niniejszy rozdział, podrozdział Właściwości fizyczne kontrolki "DBQUERYBOX"</p>
FIELD.LABEL([iNameType])	<p>Ta metoda zwraca tytuł danego pola.</p> <p>Typy tytułów (iNameType) są następujące:</p> <ul style="list-style-type: none"> • 1 - Nazwa systemowa • 2 - Nazwa SQL • 3 - Nazwa domyślna • 4 - Opis (tekst pomocy) <p>Należy wypełnić właściwość "TABLE".</p>	

Kontrolka DBTable

Kontrolka "DBTABLE" tworzy kontrolkę służącą do wprowadzania tabeli w bazie danych AssetCenter.

Właściwości

Kontrolka ten nie ma dodatkowych właściwości.

Kontrolka DBPath

Kontrolka "DBPATH" tworzy kontrolkę do wprowadzania ścieżki bazy danych AssetCenter.

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich kontroltek, kontrolka "DBPATH" musi posiadać następującą obowiązkową właściwość:

Tabela 14.28. Obowiązkowa właściwość logiczna kontrolki "DBPATH"

Nazwa właściwości=Wartość	Opis właściwości	Przykład
TABLE=<Nazwa SQL tabeli>"	Nazwa tabeli, z której chcemy wybrać ścieżkę. Właściwość typu "String" (ciąg znaków).	TABLE= amAsset

Kontrolka LinkEdit


Kontrolka "LINKEDIT" tworzy kontrolkę do wprowadzania łącza w bazie danych AssetCenter.

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich kontroltek, kontrolka "LINKEDIT" posiada następujące właściwości:

Tabela 14.29. Właściwość logiczna kontrolki "LINKEDIT"

Nazwa właściwości=Wartość	Opis właściwości	Przykład / Uwagi
TABLE=<nazwa SQL tabeli>"	Nazwa tabeli, w której chcemy wybrać łącze. Właściwość typu "String" (ciąg znaków).	TABLE="amAsset"
FILTER=<warunek WHERE zapytania AQL>	Definiuje filtr AQL. Właściwość typu "String" (ciąg znaków).	Ta właściwość jest opcjonalna.

Nazwa właściwości=Wartość	Opis właściwości	Przykład / Uwagi
LINK=<Nazwa SQL łącza>"	Nazwa SQL łącza tabeli określonej we właściwości "TABLE". Właściwość opcjonalna	LINK="POrLine"
ZOOM=<TRUE FALSE>	Wyświetla lupę (=TRUE) lub nie (=FALSE). Ta właściwość jest stosowana jedynie wtedy, gdy kreator nie jest modalny (właściwość MODAL=FALSE w węźle głównym).	
SRCCHOICE=<TRUE FALSE>	Wyświetla (=TRUE) lub nie (=FALSE) ikonę  . Ta właściwość jest stosowana jedynie wtedy, gdy kreator nie jest modalny (właściwość MODAL=FALSE w węźle głównym).	
TABLE.LABEL([iNameType])	Ta metoda zwraca tytuł źródłowej tabeli łącza. Typy tytułów (iNameType) są następujące: <ul style="list-style-type: none"> • 1 - Nazwa systemowa • 2 - Nazwa SQL • 3 - Nazwa domyślna • 4 - Opis (tekst pomocy) Należy wypełnić właściwość "TABLE".	Patrz niniejszy rozdział, podrozdział Właściwości fizyczne kontrolki "DBQUERYBOX"

Nazwa właściwości=Wartość	Opis właściwości	Przykład / Uwagi
LINK.LABEL([iNameType])	<p>Ta metoda zwraca tytuł łącza.</p> <p>Typy tytułów (iNameType) są następujące:</p> <ul style="list-style-type: none"> • 1 - Nazwa systemowa • 2 - Nazwa SQL • 3 - Nazwa domyślna • 4 - Opis (tekst pomocy) <p>Należy wypełnić właściwości "TABLE" i "LINK".</p>	

Kontrolka TextBox

Kontrolka "TEXTBOX" (pole tekstowe) tworzy kontrolkę do wprowadzania tekstu.

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich kontroltek, kontrolka "TEXTBOX" może mieć następujące właściwości:

Tabela 14.30. Właściwości fizyczne kontrolki "TEXTBOX"

Nazwa właściwości=Wartość	Opis właściwości	Przykład
MULTILINE=<Liczba>	Właściwość ta ma wartość "0", jeśli kontrolka "TEXTBOX" jest jednowierszowa, lub wartość numeryczną określającą procent wyświetlanej wysokości kontrolki, jeśli kontrolka jest wielowierszowa.	MULTILINE=50
PASSWORD=<TRUE FALSE>	Ta właściwość ukrywa (=TRUE) lub nie (=FALSE) wpisany tekst.	

Kontrolka CHART

Kontrolka "CHART" umożliwia wyświetlenie wykresu. Może się on składać z kilku serii.

Właściwości

Oprócz opcjonalnych właściwości wspólnych dla wszystkich kontroltek, kontrolka "DBEDIT" przyjmuje następujące właściwości:

Tabela 14.31. Właściwości logiczne kontrolki "CHART"

Nazwa właściwości	Opis właściwości	Przykład
SERIES	Określa listę nazw serii, oddzielonych pionową kreską. Lista ta nie może być pusta; w przeciwnym wypadku nic nie zostanie wyświetlone.	SERIES="cena zakupu cena sprzedaży"
VALUES	Określa wartości numeryczne serii wykresu. Tabela dwuwymiarowa.	VALUES="1 2,1 4"
FORMAT	Określa typ danych: <ul style="list-style-type: none"> • Długa całkowita (long) • Zmiennoprzecinkowa o podwójnej precyzji (double) • Liczba (number) • Wartość procentowa (percent) 	
SERIE	Numer serii, którą kliknąłeś Właściwość CHART musi być w trybie interaktywnym (=TRUE).	
INDEX	Numer kolumny, którą kliknąłeś. Właściwość CHART musi być w trybie interaktywnym (=TRUE).	
CLICK	Wywołuje skrypt właściwości w przypadku kliknięcia wykresu.	

Tabela 14.32. Właściwości fizyczne kontrolki "CHART"

Nazwa właściwości=Wartość	Opis właściwości	Przykład
MODE	Określa typ wykresu: <ul style="list-style-type: none"> • MODE=0: słupki pionowe • MODE=1: słupki poziome • MODE=3: kołowy 	
LABELS	Określa tytuły kolumn.	styczeń luty
3D	Określa, czy wykres jest 3-W (=TRUE) lub nie (=FALSE).	
COLORS	Określa listę kolorów dla każdej serii. Elementy listy są rozdzielone pionową kreską. Wartość RGB w systemie dziesiętnym.	255 16777215 16711680 Wyświetla kolory niebieski, biały i czerwony.
INTERACTIVE	Określa, czy wykres jest interaktywny (=TRUE) lub nie (=FALSE), to znaczy czy jest aktywowany przez kursor.	
POPUP	Wyświetla (=TRUE) lub nie (=FALSE) menu popup.	
BACKGROUND	Określa, czy wykres ma tło (=TRUE) lub nie (=FALSE)	
BACKIMAGE	Określa ścieżkę obrazu służącego za tło dla wykresu. Należy zatwierdzić właściwość "BACKGROUND" (=TRUE), aby wyświetlić tło.	
STACKED	Określa, czy słupki wykresu są skumulowane (=TRUE) lub nie (=FALSE)	
CHARTHEIGHT	Określa względny rozmiar kontrolki "CHART" względem innych kontrolki obecnych w kreatorze.	
CAPTION	Wyświetla tytuł.	
ELEVATION	Określa stopień nachylenia elementów wykresu kołowego rozsuniętego z efektem 3-W	

Nazwa właściwość=Wartość	Opis właściwości	Przykład
ROTATION	Określa kąt obrotu dla wykresu Wartość wyrażona w stopniach.	
DISPLAYLABELS	Wyświetla (=TRUE) lub nie (=FALSE) tytuły kolumn (LABELS).	
DISPLAYLEGEND	Wyświetla (=TRUE) lub nie (=FALSE) legendę serii.	

Kontrolka FILEEDIT

Kontrolka wyświetla okno dialogowe umożliwiające zapisanie lub załadowanie pliku lub folderu.

Tabela 14.33. Właściwości kontrolki "FILEEDIT"

Nazwa właściwość=Wartość	Opis właściwości	Przykład
OPENMODE	Określa typ okna dialogowego: <ul style="list-style-type: none"> • OPENMODE=1: otwórz okno • OPENMODE=2: zapisz plik • OPENMODE=4: otwórz folder • OPENMODE=8: zapisz folder 	
FILTERS	Określa kryteria wyświetlania plików, których lista znajduje się w oknie dialogowym.	(*.* *.txt *.scn *.scn
DEFEXT	Określa domyślne rozszerzenie pliku.	(*.*scn) *.scn

Kontrolka TICKEDIT

Kontrolka ta umożliwia dodanie harmonogramu.

Tabela 14.34. Właściwości kontrolki TICKEDIT

Nazwa właściwości=Wartość	Opis właściwości	Przykład
VALUE	Przedstawienie, w postaci ciągu znaków, parametrów określonych przez użytkownika w harmonogramie.	
LISTHEIGHT	Określa rozmiar kontrolki "DBLISTBOX" względem innych kontrolki "DBLISTBOX" obecnych w kreatorze. Właściwość typu "Long".	

Kontrolka CALENDAR

Kontrolka umożliwia dodanie kalendarza.

Kontrolka TIMESPANEDIT

Kontrolka umożliwia dodanie obszaru wprowadzania czasu trwania.

Kontrolka NUMBOX

Kontrolka umożliwia dodanie kontrolki typu liczbowego.

Tabela 14.35. Właściwości kontrolki NUMBOX

Nazwa właściwości=Wartość	Opis właściwości	Przykład
MINVALUE	Określa minimalną wartość liczby. Domyślnie wartość nieskończona.	
MAXVALUE	Określa maksymalną wartość liczby. Domyślnie wartość nieskończona.	

Nazwa	Opis właściwości	Przykład
właściwość=Wartość		
FORMAT	Określa format liczby: <ul style="list-style-type: none"> LONG, której format został zdefiniowany w panelu sterowania systemu operacyjnego. RAWLONG DOUBLE o formacie zdefiniowanym w panelu sterowania systemu operacyjnego. 	

Kontrolka COMBOEDIT

Kontrolka umożliwia dodanie kontrolki typu lista rozwijana.

Tabela 14.36. Właściwości kontrolki COMBOEDIT

Nazwa	Opis właściwości	Przykład
właściwość=Wartość		
VALUES	Definiuje pary wartości ("Nazwa"="Wartość") dla kontrolki. "Nazwa" określa tekst, który jest wyświetlany w kontrolce, "Wartość" - przypisywaną wartość, jeśli dana nazwa zostanie wybrana przez użytkownika. Właściwość typu "String".	

Kontrolka DATETIMEEDIT

Kontrolka umożliwia dodanie kontrolki typu data.

Tabela 14.37. Właściwości kontrolki

Nazwa właściwości=Wartość	Opis właściwości	Przykład
FORMAT	Określa format kontrolki: <ul style="list-style-type: none"> • Data (date) • Czas (time) • Data i czas Format kontrolki zależy od parametrów systemowych określonych przez użytkownika.	2002/02/07 13:37:19 2002/02/07 13:37:19

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.

- 1 Krok 1: Analiza potrzeb
- 2 Krok 2: Określenie organizacji kreatora
- 3 Krok 3: Zapis struktury kreatora za pomocą języka skryptowego

Krok 1: Analiza potrzeb

Zadaniem tego kreatora jest przeniesienie zasobów z jednej lokalizacji do innej. W tym celu musimy:

- 1 Określić przenoszone zasoby.
- 2 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.

 **Uwaga:**

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:

- 1 Liczbę stron.
- 2 Powiązania między poszczególnymi stronami.
- 3 Zawartość każdej z tych stron.

 **Uwaga:**

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:

Rysunek 14.2. Kreator - przykład organizacji



Następnie, na podstawie następnego diagramu, określamy sposób definiowania kreatora:

Strona	Może prowadzić do stron
TypWyboruZasobow	WybierzZasob, WybierzUzytkownika, WybierzLokalizacje
WybierzZasob	WybierzNowaLokalizacje
WybierzUzytkownika	WybierzZasob
WybierzLokalizacje	WybierzZasob
WybierzNowaLokalizacje	Brak

Następnie definiujemy zawartość stron. Oznacza to kontrolki umożliwiające użytkownikowi dokonanie wyboru:

Strona	Zadanie tej strony	Stosowana kontrolka
TypWyboruZasobow	Pozwala użytkownikowi wybrać jedną z trzech możliwości	Kontrolka "CHOICEBOX"
WybierzZasob	Pozwala użytkownikowi wybrać zasoby z listy rekordów w tabeli zasobów.	Kontrolka "DBLISTBOX"
WybierzUzytkownika	Pozwala użytkownikowi wybrać z tabeli wydziałów i pracowników użytkownika, którego zasoby będą przenoszone.	Kontrolka "DBLISTBOX"
WybierzLokalizacje	Pozwala użytkownikowi wybrać bieżącą lokalizację z tabeli lokalizacji.	Kontrolka "DBLISTBOX"
WybierzNowaLokalizacje	Pozwala użytkownikowi wybrać nową lokalizację z tabeli lokalizacji.	Kontrolka "DBLISTBOX"

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 = "Przeprowadzka pracowników"
VERSION = "699"
;=====
;
=====
;Ask which user to move. By default, use selection in amEmplDept if con
text is on this table

```

```

=====
=====
{ PAGE pgUser
  TITLE = "Wybierz pracowników do przeniesienia"
  { DBLISTBOX Users
    COLNAME = "Name | FirstName"
    COLWIDTH = "50 | 50"
    DBLCLICK = 1
    LABEL = "Pracownicy do przeniesienia"
    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 = "Wybierz nową lokalizację"
  { STRING UserName
    VALUE = AmDbGetString("SELECT FirstName + ' ' + Name FROM a
mEmplDept WHERE IEmplDeptId IN (" & {pgUser.Users} & ")" )
  }
  { LABEL LABEL1
    CAPTION = "Użytkownik (-cy) : " & {UserName}
  }
}

```

```

{ DBLISTBOX NewLocId
  COLNAME = "Name"
  COLWIDTH = "100"
  DBLCLICK = 1
  TABLE = "amLocation"
  VALUE = "-1"
}
{ TRANSITION trNewLocToAssets
  TO = "pgRecap"
}
}

;=====
;=====
;Recapitulation
;=====
;=====
{ PAGE pgRecap
  TITLE = "Podsumowanie"
  { LISTBOX Users
    COLTITLE = "Nom"
    COLWIDTH = "100"
    LABEL = "Pracownicy do przeniesienia"
    MANDATORY = 0
    MULTISEL = 1
    READONLY = 1
    VALUE = ""
    VALUES = AmDbGetList("SELECT FullName FROM amEmpIDept W
HERE FullName LIKE LikeParam(amEmpIDept_2:FullName)+'%' AND
amEmpIDept_2:IDeptID 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 lLocalId as long
  lLocalId = {pgNewLoc.NewLocId}

  lErr = amStartTransaction()

  dim i as Integer
  For i = 1 To iEmplCount
    lErr = AmProgress((100 * i) / iMax)
    lErr = AmLog("Przeniesienie pracownika" + {pgRecap.Users.VALUE
S(i,1)})
    hRecord = AmGetRecordFromMainId("amEmplDept", {pgRecap.Use
rs.VALUES(i,0)} )
    If hRecord <> 0 then
      lErr = AmSetFieldLongValue( hRecord, "lLocalId", lLocalId)
      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("Transakcja została anulowana")
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

Uwaga:

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:

-  przełącza edytor pomiędzy trybem tekstowym i trybem graficznym.
-  przenosi węzeł o jeden poziom wyżej wewnątrz węzła nadrzędnego.
-  przenosi węzeł o jeden poziom niżej wewnątrz węzła nadrzędnego.
-  usuwa wybrany węzeł.

Polecenia wykonywania i usuwania błędów

Polecenia te pozwalają kompilować, usuwać błędy i wykonywać skrypt:

Rysunek 14.3. Przycisk polecenia wykonywania i usuwania błędów



Narzędzie do wyszukiwania

Na pasku narzędzi umieszczone jest narzędzie wyszukiwania, za pomocą którego można wyszukać ciąg znaków w strukturze drzewa kreatora (skrót klawiszowy "Ctrl+F" uruchamia tę kontrolkę 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).

Uwaga:

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.
- Wartości zmodyfikowane są wyświetlane w kolorze niebieskim.

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.



Uwaga:

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 nadrzędny. 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
-

Uwaga:

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.

Uwaga:

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.

 **Uwaga:**

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" kontrolki "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:

```
{Właściwość} = 123
```

Odpowiedź

Aby przypisać wartość do właściwości, należy użyć funkcji "AmsetProperty()", jak poniżej:

```
Dim irc as Integer
irc= AmsetProperty("Właściwość", 123)
```

 Uwaga:

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:

- 1 Zbiera informacje za pomocą szeregu stron (dostęp z zapisem jest zabroniony nawet dla administratora AssetCenter)
- 2 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 kontrolki "DBLISTBOX" w stronie kreatora spada wydajność. Czy to normalne?

Odpowiedź

Problem ten występuje przy zastosowaniu kontrolki "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 kontrolce "LISTBOX".

Odpowiedź

Należy użyć właściwości "EDITABLE" tej kontrolki. 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("am", "AssetCenter", "OpenTable(amAsset)")
```

Pytanie

Jaka jest różnica między właściwościami "COLNAME" i "COLTITLE" kontrolki "LISTBOX"?

Odpowiedź

Tytuły kolumn w kontrolce "LISTBOX" mogą być zdefiniowane automatycznie lub ręcznie:

- Właściwość "COLNAME", związana z właściwością "TABLE" umożliwia automatyczne określenie tytułów kolumn w kontrolce "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 = "Name | | FirstName"
COLTITLE = " | A | B"
...
```

wyświetla następujące etykiety kolumn kontrolki "LISTBOX": Nazwisko, A, B.


Właściwość "COLNAME" określa również typ kontrolki używanego, gdy możliwa jest edycja wartości kolumn.

15 | Zawiadomienia

ROZDZIAŁ

Niniejszy rozdział wyjaśnia, w jaki sposób rozsyłać zawiadomienia i zarządzać nimi z pomocą AssetCenter.

Lista zawiadomień jest dostępna za pomocą menu **Narzędzia/ Zawiadomienia**.

Pasek wyświetlania zawiadomień można aktywować/ dezaktywować za pomocą menu **Okno/ Wyświetl roletę zawiadomień** lub przycisku  na pasku narzędzi.

Definicja zawiadomienia

Zawiadomienie jest to aktualna informacja do rozesłania określonej grupie pracowników w danym okresie czasu.

Pracownicy ci należą do grup pracowników.

A priori, zawiadomienia zawierają jedynie informacje na najbliższy okres.

Przykład zawiadomienia: "Serwer XXX będzie niedostępny między 11.00 a 12.00 10 lutego 2002".

Ogólne funkcjonowanie zawiadomień

Tworzenie zawiadomienia

Użytkownik, który posiada uprawnienia tworzenia w tabeli zawiadomień może tworzyć zawiadomienia za pomocą menu **Narzędzia/Zawiadomienia**.

W szczegółach zawiadomienia można znaleźć:

- treść wiadomości,
- listę dystrybucyjną dla zawiadomienia,
- okres ważności zawiadomienia.

Odczyt zawiadomienia

Roleta zawiadomień umożliwia użytkownikowi odczytanie wszystkich zawiadomień, pod warunkiem, że znajduje się on na ich listach dystrybucyjnych.

Priorytet zawiadomienia

W celu określenia priorytetu zawiadomienia, wystarczy wypełnić pole **Priorytet (Nazwa SQL : seSeverity)** w szczegółach zawiadomienia.

Z każdym priorytetem jest powiązany kolor, który można wybrać w polu **Kolor (Nazwa SQL : IColor)**; pojawi się on na rolicie zawiadomień.

Treść rozsyłanego zawiadomienia

Zakładka **Wiadomość** w szczegółach zawiadomienia zawiera tekst do rozesłania.

Wiadomość może zawierać 255 dowolnych znaków.




Lista odbiorców zawiadomienia

Zakładka **Rozsyłanie** w szczegółach zawiadomienia zawiera listę grup pracowników, którzy będą mogli odczytać zawiadomienie.

Pole wyboru **Do wszystkich grup pracowników (Nazwa SQL : bAllGroups)**

Jeżeli zaznaczysz to pole, członkowie wszystkich grup pracowników odczytają zawiadomienie.

W przeciwnym wypadku odczytają ją jedynie członkowie grup wybranych na liście w zakładce **Rozsyłanie**.

Użyj przycisków ,  i , aby dodać, usunąć, wyświetlić lub zmodyfikować grupy pracowników na liście odbiorców.

Pole wyboru **Z uwzględnieniem podgrup (Nazwa SQL : bChildGrps)**

Zawiadomienie można wysłać do wszystkich grup pracowników lub tylko do wybranych. Jeżeli zaznaczysz to pole (domyślnie jest ono zaznaczone), wszystkie podgrupy wybranej grupy odczytają zawiadomienie, ponieważ tabela grup pracowników ma charakter hierarchiczny.

Wyświetlanie zawiadomienia

Aby wyświetlić zawiadomienia:

- 1 Aktywuj roletę zawiadomień.
- 2 Użyj przycisków znajdujących się na roletce zawiadomień, aby je wyświetlić.

Możliwe jest określenie:

- kolorów wyświetlania zawiadomień w zależności od ich priorytetu.
- trybu automatycznego odświeżania rolety zawiadomień.

Aktywacja rolety zawiadomień.

Wszyscy użytkownicy AssetCenter mogą aktywować roletę zawiadomień. Pojawiają się na niej zawiadomienia przeznaczone dla nich. Roletę zawiadomień można aktywować/ dezaktywować na dwa różne sposoby:






- za pomocą menu **Okno/ Wyświetl roletę zawiadomień**
- za pomocą ikony  na pasku narzędzi

Tabela 15.1. Przyciski na roletce zawiadomień.

	Kliknij ten przycisk, aby odczytać bieżące wiadomości nie czekając na termin odświeżania określony w opcjach (menu Edycja/ Opcje).
	Kliknij ten przycisk, aby wyświetlić poprzednią wiadomość.
	Kliknij ten przycisk, aby wyświetlić następną wiadomość.
	Kliknij ten przycisk, aby przerwać lub ponownie uruchomić wyświetlanie zawiadomień na roletce.



October 10, 2002