

HP Asset Manager

Версия ПО: 5.20

Расширенное использование

Дата выпуска документа: 01 October 2009
Дата выпуска ПО: October 2009



Правовые уведомления

Уведомления об авторских правах

(c) Copyright 1994-2009 Hewlett-Packard Development Company, L.P.

Уведомление об ограниченных правах

Конфиденциальное компьютерное программное обеспечение.

Для обладания продуктом, его использования или копирования необходима действующая лицензия от компании HP.

В соответствии с Федеральными правилами закупок США (FAR) 12.211 и 12.212, коммерческое программное обеспечение, документация на программное обеспечение и технические данные на коммерческие продукты лицензируются правительству США на условиях стандартной коммерческой лицензии поставщика.

Гарантия

Единственной гарантией на продукты и услуги компании HP являются прямые заявления о гарантии, сопровождающие такие продукты и услуги.

Никакая информация в настоящем документе не может быть истолкована как предоставление дополнительной гарантии.

Компания HP не несет ответственности за технические или редакционные ошибки, а также пропуски в настоящем документе.

Содержащаяся здесь информация может быть изменена без уведомления.

Получить дополнительные сведения о лицензионных соглашениях третьей сторон и/или лицензиях на ПО с открытым кодом либо просмотреть открытый код можно одним из следующих способов.

- В каталоге ThirdParty установочного диска
- В каталогах, в которых расположены двоичные файлы сторонних инструментов и/или инструментов с открытым исходным кодом, после установки Asset Manager
- С помощью URL-адреса компонента, указанного в руководстве Лицензионные соглашения на использование стороннего программного обеспечения и ПО с открытым исходным кодом

Уведомления о торговых марках

- Adobe®, Adobe logo®, Acrobat® and Acrobat Logo® are trademarks of Adobe Systems Incorporated.
- Corel® and Corel logo® are trademarks or registered trademarks of Corel Corporation or Corel Corporation Limited.
- Java™ is a US trademark of Sun Microsystems, Inc.
- Microsoft®, Windows®, Windows NT®, Windows® XP, Windows Mobile® and Windows Vista® are U.S. registered trademarks of Microsoft Corporation.
- Oracle® is a registered trademark of Oracle Corporation and/or its affiliates.
- UNIX® is a registered trademark of The Open Group.

Содержание

Глава 1. Подробные списки	13
Настраиваемые подробные списки	14
Системные подробные списки	15
Глава 2. Сохранение истории	17
Создание записи	18
Изменение поля в таблице или ссылки «1» (например: «Пользователь актива»)	19
Добавление n-ссылки к другой таблице (например: «Активы, охватываемые контрактом»)	19
Удаление n-ссылки на другую таблицу	20
Изменение n-ссылки на другую таблицу	20
Сохранение истории признаков, связанных с записями	20
Глава 3. Запросы AQL	23
Инструменты AQL	23
Представление	23
Редактор запросов	26
Рекомендации по созданию запросов AQL	33
Сортировка и индексы:	43
Синтаксис AQL	45
Справочная информация по функциям языка AQL	58

Примеры запросов:	64
Глава 4. Crystal Reports	71
Ограничения	71
Общий обзор	71
Установка и настройка для активации доступа Windows-клиентов Asset Manager к отчетам, сохраненным в базе данных Asset Manager	73
Установка и настройка для активации доступа веб-клиентов Asset Manager к отчетам, сохраненным в базе данных Crystal Reports Server	74
Отображение отчета	88
Обеспечение доступности отчетов	92
Изменение отчета Seagate Crystal	97
Создание отчета со сведениями	97
Печать отчета	99
Идентификация отчетов Crystal Reports, свойственных данному модулю	101
Привязка отчета к кнопке на экране (Windows-клиент) или к действиям на экране (веб-клиент)	102
Глава 5. Инфопанели	103
Создание инфопанели	103
Пример инфопанели	105
Глава 6. Статистика	107
Создание статистики	107
Использование скрипта вместо запроса	109
Оптимизация работы	110
Масштабирование сведений о статистике	111
Определение максимального количества групп для индивидуального представления	111
Просмотр статистики	112
Отображение статистики	113
Примеры статистики	113
Определение настраиваемых цветов для статистики в виде диаграммы	126
Глава 7. Действия	129
Рекомендации по разработке	129
Определение действия.	130

Создание действия	132
Примеры действий	141
Использование переменных	145
Тестирование действия	145
Выполнение действия	146
Связывание действия с кнопкой на экране	148

Глава 8. Обмен сообщениями 149

Обзор обмена сообщениями	149
------------------------------------	-----

Глава 9. Рабочий процесс 153

Определения	154
Общий обзор	156
Реализация рабочего процесса	157
Использование графического редактора рабочего процесса	158
Пример рабочего процесса, используемого при утверждении заявки	160
Контекст рабочего процесса	178
Роли в рабочем процессе	180
Операции рабочих процессов	182
Задачи	187
События	191
Переходы рабочих процессов	202
Оповещения рабочих процессов и лимиты времени	202
Группы выполнения рабочих процессов	204
Отслеживание рабочего процесса	205
Удаление экземпляров завершенных рабочих процессов	207
Техническая информация: модель данных	212

Глава 10. Экспорт данных и создание представлений SQL 215

Определения	215
Экспорт данных из базы данных Asset Manager	216
Управление представлениями SQL в базе данных Asset Manager	217
Рекомендации	218
Определение скрипта экспорта	218
Выполнение скрипта экспорта	224

Глава 11. Скрипты 227

Определение скрипта	227
Применение скриптов	229

Введение в функции	230
Классификация функций Basic	234
Первые шаги в написании скриптов	234
Библиотеки скриптов	237
Советы и предупреждения	238
Первый пример	243
Второй пример	246

Глава 12. Календари 249

Обзор календарей	249
Влияние календарей на некоторые области функциональности	250
Методология, используемая для создания календаря	250
Описание создания календаря	250

Глава 13. Часовые пояса 257

Управление часовыми поясами	257
Внедрение часовых поясов	258
Создание часовых поясов	258
Управление часовыми поясами	259
Управление часовыми поясами в Asset Manager Automated Process Manager	265
Последствия для различных операций	266

Глава 14. Вычисляемые поля 271

Определение вычисляемого поля	271
Полезность вычисляемых полей	272
Создание вычисляемых полей	272
Использование вычисляемых полей	277

Глава 15. Мастера 281

Соглашения о нотации	281
Определения	282
Структурный шаблон	285
Шаблон страницы мастера	287
Общее число пунктов	287
Общая структура и синтаксис	287
Свойства узла	288
Соединение мастеров	293
Функции Basic	294
Определение корневого узла	295
Синтаксис корневого узла	295

Свойства корневого узла	295
Подузлы корневого узла	300
Определение узла «Страница»	301
Синтаксис узла «Страница»	301
Свойства узла «Страница»	301
Подузлы узла «Страница»	303
Определение узла перехода	303
Синтаксис узла перехода	303
Свойства узла перехода	304
Особенности узла перехода	304
Определение конечного узла	305
Определение начального узла	306
Определение узла таймера	307
Определение узлов «Длинный» и «Строка»	308
Определение узла элемента управления	308
Общий синтаксис узла элемента управления	309
Типы элементов управления и связанные свойства	309
Использование графического редактора	336
Пример создания мастера	340
Часто задаваемые вопросы	344

Глава 16. Новости 349

Определение позиции новостей	349
Обзор новостей	350
Важность позиций новостей	350
Сообщение для рассылки	350
Список рассылки новостей	350
Отображение новостей	351

Указатель 353

Список иллюстраций

3.1. Редактор запросов - режимы составления запросов	27
7.1. Действие исполняемого типа - окно сведений	142
7.2. Действие типа «Обмен сообщениями» с ссылочным объектом - окно сведений	143
8.1. Обмен сообщениями - обзор	150
9.1. Рабочий процесс - упрощенная схема	154
9.2. Рабочий процесс в Asset Manager - обзор	157
9.3. Рабочий процесс - проверка заявки	161
9.4. Рабочий процесс - схема проверки заявки	175
9.5. Вкладка параметров действия типа «База данных»	193
9.6. Пример схемы синхронного рабочего процесса	200
9.7. Пример схемы асинхронного рабочего процесса	201
9.8. Схема рабочего процесса с заключительным событием	202
9.9. Рабочий процесс - основные таблицы, позволяющие определить схему рабочего процесса	212
9.10. Рабочий процесс - основные таблицы, необходимые, когда выполняется экземпляр рабочего процесса	213
11.1. Составитель скриптов	232
12.1. Календарь - вкладка «Расписания»	251
12.2. Календарь - вкладка «Предварительный просмотр»	255
15.1. Мастера - структурные шаблоны	286
15.2. Кнопка выполнения и отладки	337
15.3. Мастер - пример организации	341

Список таблиц

1.1. Значения в системных подробных списках	15
3.1. AQL - соглашения о синтаксисе	45
3.2. AQL - логические операторы	50
3.3. AQL - операторы сравнения	51
3.4. AQL - функции группового типа	59
3.5. AQL - функции строкового типа	59
3.6. AQL - функции типа даты	60
3.7. AQL - примеры использования функции типа даты	62
3.8. AQL - функции числового типа	63
3.9. AQL - функции проверочного типа	64
9.1. Ограничение числа выполняющихся для данного объекта рабочих процессов - различные возможные случаи	196
9.2. Различные способы, которыми может быть обработано событие	197
9.3. Различные способы, которыми может быть обработано событие	199
11.1. Функции/параметры - типы	233
14.1. Типы вычисляемых полей	273
15.1. Используемые соглашения	282
15.2. Логические свойства корневого узла	296
15.3. Физические свойства корневого узла	298
15.4. Подузлы корневого узла	300
15.5. Логические свойства узла «Страница»	302
15.6. Физические свойства узла «Страница»	302
15.7. Подузлы узла «Страница»	303

15.8. Логические свойства узла перехода	304
15.9. Логические свойства конечного узла	305
15.10. Физические свойства конечного узла	306
15.11. Логические свойства начального узла	306
15.12. Логическое свойство узла таймера	307
15.13. Логические свойства узлов «Длинный» и «Строка»	308
15.14. Логические свойства, общие для всех элементов управления	310
15.15. Физические свойства, общие для всех элементов управления	311
15.16. Свойство элемента управления CHECKBOX	313
15.17. Физические свойства элемента управления COMBOBOX	314
15.18. Физические свойства элемента управления OPTIONBUTTONS	315
15.19. Физические свойства элемента управления LISTBOX	316
15.20. Методы элемента управления LISTBOX	318
15.21. Обязательное логическое свойство элемента управления LISTBOX	319
15.22. Физические свойства элемента управления LABEL	320
15.23. Физические свойства элемента управления PROGRESSBAR	320
15.24. Физические свойства элемента управления COMMANDBUTTON	321
15.25. Физические свойства элемента управления DBLISTBOX	322
15.26. Физические свойства элемента управления DBQUERYBOX . . .	325
15.27. Физические свойства элемента управления DBEDIT в обычном режиме	328
15.28. Обязательное логическое свойство элемента управления DBPATH	329
15.29. Логическое свойство элемента управления LINKEDIT	329
15.30. Физическое свойство элемента управления TEXTBOX	331
15.31. Логические свойства элемента управления CHART	331
15.32. Физические свойства элемента управления CHART	332
15.33. Свойства элемента управления FILEEDIT	334
15.34. Свойства элемента управления TICKEDIT	334
15.35. Свойства элемента управления NUMBOX	335
15.36. Свойства элемента управления COMBOEDIT	335
15.37. Свойства данного элемента управления	336
16.1. Кнопки бегущей строки новостей	352

1 Подробные списки

Подробный список представляет собой список значений, которые Asset Manager предлагает для заполнения некоторых полей (стандартных полей на экране сведений или значение признака), например, заголовок, положение, страна, бренд. Они хранятся в таблице **Подробные списки** (selfemizedList) в базе данных.

Это позволяет стандартизировать значения в данных полях и упрощает ввод данных.

Если экран содержит поле подробного списка, его значения отображаются пользователям в виде «раскрывающегося списка». Чтобы назначить значение полю, пользователю достаточно выбрать запись из списка.

Значения списка хранятся в таблице **Значения подробного списка** (amItemVal) базы данных и связаны с соответствующими списками в таблице **Подробные списки** (selfemizedList)

Стандартная база данных Asset Manager содержит поля, которые связаны с двумя типами подробных списков:

- Настраиваемые подробные списки
- Системные подробные списки

Настраиваемые подробные поля могут быть закрытыми (не изменяются пользователями) или открытыми (пользователи могут добавлять новые записи списка). Независимо от того, является ли подробный список открытым или закрытым, изменения в него могут вноситься любым пользователем с правами на чтение и запись к таблице **Подробные списки** (selfemizedList).



ЗАМЕЧАНИЕ:

Администраторы могут создавать новые поля с помощью Asset Manager Application Designer, которые они могут связывать с системным или настраиваемым списком.

Настраиваемые подробные списки

Доступ к настраиваемым подробным спискам можно получить с помощью ссылки **Администрирование/Система/ Подробные списки** в навигационной панели.

Значения настраиваемых подробных списков

Список «значений», появляющийся в сведениях о подробном списке, содержит значения, которые будут предлагаться при заполнении пользователем поля, связанного с данным подробным списком.

Значения можно удалять, изменять или добавлять в сведениях подробного списка.

Открытые настраиваемые подробные списки

Для поля **Тип** (имя SQL: seType) в сведениях о таких подробных списках устанавливается значение Открытый.

Пользователи Asset Manager могут вводить другие значения, отличные от предлагаемых в списке.

Если пользователь вводит новое значение, оно добавляется к списку значений подробного списка (данный список значений используется всеми пользователями). Пользователь получает сообщение с запросом на подтверждение создания.

Закрытые настраиваемые подробные списки

Для поля **Тип** (имя SQL: seType) в сведениях о таких подробных списках устанавливается значение Закрытый.

Пользователи Asset Manager вводят только значения, включенные в данный список.

Системные подробные списки

Список значений в системном подробном списке определяется с помощью Asset Manager Application Designer. Список может быть настроен администраторами.

Данные подробные списки нельзя изменить с помощью ссылки **Администрирование/Система/ Подробные списки** в навигационной панели.

Значения в системных подробных списках

Отображаемые значения отличаются от значений, хранящихся в базе данных. В базе данных эти значения хранятся в виде чисел.

Пример поля Назначение (seAssignment) в сведениях о позиции портфеля:

Таблица 1.1. Значения в системных подробных списках

Значение, хранящееся в базе данных	Отображаемое значение
0	Используется
1	В запасе
2	Выбывший (или израсходованный)
3	Ожидание получения
4	Возврат для обслуживания
5	Возврат поставщику
6	Отсутствует

Существует несколько способов доступа к значениям в системном подробном списке:

- использование контекстной справки (только в клиенте Windows) для поля, заполняемого с помощью системного подробного списка;
- из Asset Manager Application Designer;
- использование файла database.txt для описания структуры базы данных. Этот файл находится в каталоге <папка установки Asset Manager>\ doc\ infos.

2 Сохранение истории

Любые изменения, внесенные в значения полей и ссылки таблиц в базе данных, могут отслеживаться и записываться. При каждом создании, изменении или удалении значения в поле, для которого сохраняется история, Asset Manager создает строку истории на вкладке **История** данного экрана. Чтобы этого добиться, необходимо указать сохранение истории для данного поля или данной ссылки. Для этого:

- 1 В контекстном меню выберите пункт Настройка объекта.
- 2 Перейдите на вкладку Общее экрана конфигурации.
- 3 Установите для поля История значение Да.
- 4 Для подтверждения нажмите **ОК**.



ЗАМЕЧАНИЕ:

Все изменения, касающиеся истории, сохраняются в базе данных сразу после нажатия кнопки **ОК в окне настройки базы данных. Можно также определить, сохраняется ли история для поля или ссылки, с помощью Asset Manager Application Designer.**

Если для каких-либо поля или ссылки сохраняется история, она доступна для всех пользователей Asset Manager.

Если хотя бы для одного поля или ссылки сохраняется история, в окне сведений о записи для данной таблицы появляется вкладка История. Здесь хранятся «строки истории», которыми подробно описываются все изменения, сделанные в поле или в ссылке.

Строки истории содержат несколько элементов информации:

- Модифицировано (имя SQL: dtLastModif): дата выполнения модификации.
- Автор (имя SQL: Author): лицо, выполнившее модификацию (имя входа, фамилия и имя).
- Поле или ссылка (имя SQL: Field): имя измененного поля (короткое описание).
- Предыдущее значение (имя SQL: PreviousVal): предыдущее значение измененного поля (кроме полей типа «комментарий»).
- Новое значение (имя SQL: NewVal): новое значение измененного поля (кроме полей типа «комментарий»). По умолчанию данное поле не отображается в списке. Чтобы его отобразить, щелкните правой кнопкой мыши список, затем выберите **Сервис/Настройка списка**.



ЗАМЕЧАНИЕ:

При импорте базы данных из более старой версии Asset Manager, поле Новое значение для строк истории будет пустым.

- Предыдущий комментарий (имя SQL: memPreviousCmt): предыдущее значение поля типа «комментарий». Поля типа «комментарий» не обрабатываются в том же порядке, что и остальные поля, так как они иначе хранятся в базе данных (максимальный размер: 32767 символов).

Поведение Asset Manager неодинаково и зависит от типа объекта, для которого сохраняется история.

Создание записи

Создания записей фиксируются, если Asset Manager указано отслеживать историю всех изменений, сделанных в идентификационном поле, соответствующем первичному ключу таблицы.

Записи Asset Manager:

- Модифицировано: дата создания
- Автор: автор создания
- Поле или ссылка: «создание»
- Предыдущее значение: «создание»

Изменение поля в таблице или ссылки «1» (например: «Пользователь актива»)

Записи Asset Manager:

- Модифицировано: дата выполнения модификации.
- Автор: автор модификации.
- Поле или ссылка: имя модифицированного поля.
- Предыдущее значение: предыдущее значение модифицированного поля.
- Новое значение: новое значение модифицированного поля.



ЗАМЕЧАНИЕ:

Если модифицированное поле заполнено с помощью системного подробного списка, в полях **Предыдущее значение** и **Новое значение** сохраняется значение, отображаемое в поле ввода системного подробного списка, а не значение, сохраненное в базе данных.

Например, в таблице **Наряды на работу** (amWorkOrder) поле **Статус** (seStatus) заполняется из системного подробного списка. Одна из записей этого списка отображается со статусом **Уведомлено** и сохраняется как 0.

В полях **Предыдущее значение** и **Новое значение** хранится **Уведомлено**, а не 0.

Такой принцип работы реализован в программе, начиная с Asset Manager версии 5.00.

В предыдущих версиях использовалось значение, хранившееся в базе данных.

Добавление n-ссылки к другой таблице (например: «Активы, охватываемые контрактом»)

Записи Asset Manager:

- Модифицировано: дата выполнения добавления.
- Автор: автор добавления.
- Поле или ссылка: имя ссылки.
- Предыдущее значение: ссылки связанной записи, которая была добавлена.
- Новое значение: новое значение модифицированной ссылки.

Удаление n-ссылки на другую таблицу

Записи Asset Manager:

- Модифицировано: дата выполнения удаления.
- Автор: автор удаления.
- Поле или ссылка: ссылки связанной записи, которая была удалена.
- Предыдущее значение: ссылки связанной записи, которая была удалена.
- Новое значение: новое значение модифицированной ссылки (пустое).

Изменение n-ссылки на другую таблицу

Asset Manager не записывает изменения, внесенные в ссылку. Для отслеживания необходимо удалить устаревшую ссылку, затем добавить новую.

Сохранение истории признаков, связанных с записями

В Asset Manager можно сохранять историю признаков, как и любого другого поля базы данных. История признаков охватывает:

- добавление признаков;
- удаление признаков;
- изменение значения признака.

В истории сохраняются несколько типов действий.

Добавление признака

Добавление новых признаков записывается, если для поля Сохранять историю (имя SQL: seKeepHistory) для признака установлено значение Да и если выбран параметр Сохранять историю даже при создании основной записи (имя SQL: bCreationHistory).

Записи Asset Manager:

- Модифицировано (имя SQL: dtLastModif): дата добавления признака.
- Автор (имя SQL: Author): лицо, добавившее признак.
- Предыдущее значение: «создание».
- Поле или ссылка: имя SQL признака.

Удаление признака

Удаление признаков записывается, если для поля Сохранять историю для признака установлено значение Да.

Записи Asset Manager:

- Модифицировано: дата удаления признака.
- Автор: лицо, удалившее признак.
- Поле или ссылка: имя SQL признака.
- Предыдущее значение: «Удалить признак («Значение признака»)».
- Новое значение: новое значение признака (пустое).

Изменение значения признака

Изменения признаков записываются, если для поля Сохранять историю (имя SQL: seKeepHistory) для признака установлено значение Да.

Записи Asset Manager:

- Модифицировано: дата модификации признака.
- Автор: лицо, модифицировавшее признак.
- Поле или ссылка: имя SQL признака.
- Предыдущее значение: предыдущее значение признака.
- Новое значение: новое значение модифицированного признака.



ВНИМАНИЕ:

При удалении записи также удаляются все строки истории, или напрямую, или с помощью Сервера Asset Manager.

Создание, удаление или изменение строк истории

Нельзя сохранять историю создания строк истории.

Создание строк истории

Чтобы инициировать создание строк истории в будущем, необходимо установить для поля Сохранять историю значение Да. Для этого выберите вкладку Параметры в сведениях о признаке и щелкните кнопку  напротив строки параметров.

Asset Manager отображает экран со сведениями о параметрах признака. Поле Сохранять историю находится во вкладке Ограничения на этом экране.

Когда для данного поля установлено значение Да, Asset Manager автоматически создает строки истории для данного признака. Строки истории

можно просматривать на вкладке История связанной с данным признаком таблицы.



ВНИМАНИЕ:

При удалении записи также удаляются все строки истории, или во время удаления, или с помощью Сервера Asset Manager. Нельзя сохранять историю создания истории.

3 Запросы AQL

Инструменты AQL

Можно выполнять запросы AQL в графическом интерфейсе Asset Manager, в данном случае они контекстные, или выполнять их в Asset Manager Export, в таком случае они будут неконтекстными.

Представление

В данном разделе описывается язык AQL и перечисляется, где может понадобиться использовать запросы.

AQL

AQL (Advanced Query Language) - язык запросов, который использует Asset Manager для доступа к базе данных Asset Manager. Его можно сравнить с языком SQL. Запросы, написанные на языке AQL, автоматически преобразовываются в язык SQL, соответствующий используемой СУБД.



ЗАМЕЧАНИЕ:

Перед тем, как непосредственно использовать язык AQL, надо быть хорошо знакомым с языком SQL и иметь фундаментальные познания в базах данных.

Полезность языка AQL

Язык AQL лучше подходит для запросов к базе данных Asset Manager, чем язык SQL, по следующим причинам.

Независимость от СУБД

Всеми различными СУБД, которые поддерживает Asset Manager, используются различные версии языка SQL, которые несовместимы друг с другом. Язык AQL не зависит от используемой СУБД.

Поэтому при миграции к другой СУБД запросы, написанные на языке AQL, будут выполняться также, как и до этого.

Например, функция Substring в языке AQL эквивалентна функции Substr для версии SQL Oracle for WorkGroups и функции Substring для версии Microsoft SQL Server SQL.

Создание оптимизированного кода SQL

С помощью языка AQL создается код SQL, оптимизированный в соответствии с используемой СУБД.

Это приводит к большим различиям при использовании индексов. Например, при поиске полных имен моделей с принудительным использованием индексов

Код модели (Model_IDModelId) и **Полное имя (FullName)** на языке AQL было бы написано следующее:

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

Созданный код SQL будет различаться в соответствии с используемой СУБД и будет оптимизирован для нее. Таким образом, код SQL Oracle будет следующий:

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

Для Microsoft SQL Server будет использоваться код:

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

Код IBM DB2:

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


Упрощенный доступ к базе данных Asset Manager

Благодаря языку AQL упрощается обработка ссылок и объединений. Тем самым значительно упрощается доступ к базе данных при написании запросов по сравнению с непосредственным использованием языка SQL.

Кроме того, язык AQL упрощает доступ к признакам, позволяя использовать их как прямые поля в связанных таблицах.

С помощью языка AQL также упрощается использование вычисляемых полей.

Особенности языка AQL по сравнению с языком SQL

Язык AQL не поддерживает заказы на языке DDL (язык определения данных).

Язык AQL содержит дополнения, которыми упрощается обработка объединений, признаков и вычисляемых полей.



ВНИМАНИЕ:

Никогда не следует выполнять записи в базе данных Asset Manager непосредственно с помощью инструкций SQL.

Запросы в Asset Manager

Запросы позволяют объединять критерии, касающиеся информации в таблице или в связанных таблицах.

Можно использовать запросы:

- для создания фильтров для списков записей. В таком случае запросы, в общем случае, имеют простую форму и основаны на операторе WHERE;
- для определения представлений;
- для определения условий экспорта в модели экспорта;
- для создания отчетов Crystal Reports;
- для создания мастеров;
- если используется интерфейс API Asset Manager;
- если Asset Manager используется как сервер DDE.

AQL (Advanced Query Language) - внутренний язык запросов Asset Manager, предназначенный для упрощения доступа к базе данных Asset Manager.

Asset Manager имеет в своем составе редактор, который позволяет писать запросы на языке AQL:

- или косвенно с помощью графического интерфейса;
- или с помощью написания запроса непосредственно на языке AQL.



ВНИМАНИЕ:

Чтобы показать возможности языка AQL, в приведенных ниже примерах используется весь синтаксис AQL. В частности, явно показаны операторы **SELECT**, **WHERE** и **FROM**. Некоторые функции, например, фильтры запросов (в которых пользователь определяет в запросе AQL только оператор **WHERE**) или построитель выражений, заметно упрощают создание запросов для пользователя (не показываются некоторые операторы). Данные примеры нельзя использовать для таких функций.

Редактор запросов

Asset Manager имеет в своем составе редактор запросов. Данный инструмент позволяет составлять запросы и предварительно просматривать их результаты. Он предназначен, в основном, для администраторов баз данных и опытных пользователей.



ЗАМЕЧАНИЕ:

Функция предварительного просмотра отсутствует в веб-клиенте **Asset Manager**, а также некоторых упрощенных версиях Windows-клиента **Asset Manager**.

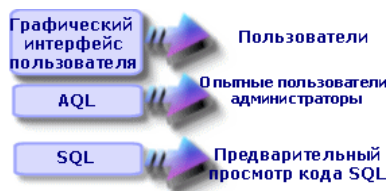
Обзор

Редактор запросов позволяет составлять запросы:

- или с помощью графического интерфейса (помогающего составлять запросы);
- или с помощью написания запроса непосредственно на языке AQL.

В случае использования Windows-клиента Asset Manager преобразование кода в SQL можно в реальном времени посмотреть с помощью функции предварительного просмотра как при использовании графического метода, так и при написании запросов непосредственно на языке AQL (оба подхода часто объединяются). Однако нельзя писать запросы непосредственно на языке SQL.

Рисунок 3.1. Редактор запросов - режимы составления запросов



С помощью редактора запросов опытный пользователь или администратор могут создавать, изменять или удалять запросы AQL. Такие запросы могут использоваться в соответствующем контексте их авторами или другими пользователями.

Доступ к редактору запросов

Клиент Windows

Редактор запросов доступен в Windows-клиенте Asset Manager:

- С помощью пункта меню Инструменты/Запросы. С помощью данного меню можно создавать запросы для своего собственного использования, которые также могут свободно использовать другие пользователи. Данные запросы могут выполняться:



ЗАМЕЧАНИЕ:

Вкладка «Запрос» имеет две вкладки, Фильтр (оператор WHERE) и Предварительный просмотр:

- **Вкладка Фильтр (оператор WHERE) представляет собой графический интерфейс, которым устанавливаются условия запроса. Им определяются элементы оператора WHERE.**
 - **С помощью вкладки Предварительный просмотр отображается преобразование запроса в код SQL и выполняется его проверка.**
- с помощью фильтра запроса, когда отображается основная таблица запроса;
 - С помощью многочисленных функций Asset Manager, которым вызываются запросы: ограничения запросов, фильтры запросов, конфигурация списка и так далее;
 - с помощью внешних программ: Asset Manager Export Tool и так далее.

Показываемая версия редактора запросов до некоторой степени упрощена в соответствии с контекстом.

Пример: предположим, что у вас есть следующий запрос:

```
SELECT [FIRST_ROWS] <field>[, <field>...] FROM <таблица> [WHERE <оператор>] [ORDER BY <оператор>]
```

В упрощенных версиях редактора запросов (простые фильтры, фильтры запросов и так далее) необходимо определять в запросе только оператор WHERE. Остальные компоненты запроса (начальная таблица, поля и так далее) являются неявными. Например, в случае фильтра запросов, фильтр применяется для таблицы, поля и условия сортировки представляют собой столбцы и условия сортировки, определенные с помощью пункта контекстного меню **Сервис/Настройка списка**. То же самое верно для редактора запросов, доступного с помощью пункта меню Инструменты/Запросы.

Таким образом, полный запрос:

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

при использовании в фильтре запроса (когда в явной форме дается только оператор WHERE) для таблицы моделей запишется следующим образом:

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

С другой стороны, команда Настройка списка обеспечивает доступ к более функциональной версии редактора запросов.

- Вкладкой Столбцы и сортировка определяются поля, которые должны отображаться в столбцах, и условия сортировки (такие условия сортировки соответствуют оператору ORDER BY).
- При установленном флажке Выполнить принудительное индексирование в коде SQL замещается оператор FIRST_ROWS.
- Вкладкой Фильтр (оператор WHERE) определяется оператор WHERE.
- Таблица является неявной.

Веб-клиент

Доступ к редактору запросов можно получить только с помощью таблицы «Запросы».

- Выберите пункт меню **Администрирование/Запросы**.
- Щелкните кнопку **Создать**.
- Выберите вкладку **Запрос**.
- Щелкните значок над текстовым полем, редактор запросов отобразится во всплывающем окне.

Создание запроса с помощью редактора запросов

Этап 1: заполнение полей в верхней части сведений о запросе.

Необходимо указать начальную таблицу для своего запроса.

Если необходимо, чтобы создаваемый запрос был доступен другим пользователям, снимите флажок Скрыто (имя SQL: bPrivate) на вкладке

Общее.

ЗАМЕЧАНИЕ:

Администратор имеет доступ ко всем запросам, хранящимся в базе данных, даже к запросам с установленным параметром Скрыто.

После заполнения основной информации в запросе щелкните **Запрос**, чтобы начать работу над запросом.

Этап 2: определение условий фильтра

Редактор запросов Asset Manager позволяет определять условия, которыми объединяются поля, выражения вычисления, константы и операторы.

Можно определить одно или несколько условий фильтров.

Клиент Windows

Определение условия фильтра в клиенте Windows.

- 1 В начальной таблице выберите поле, константу или выражение (Поле 1), которые необходимо сравнить с полем, константами или выражением (Поле 2).
- 2 Подтвердите условие фильтра, переместив его в нижнюю часть экрана с помощью кнопки .
- 3 Чтобы определить несколько условий фильтра, связанных такими логическими операторами как AND и OR, создайте другие условия фильтра и подтвердите их с помощью кнопки **AND** или **OR**.
- 4 Подтвердите запрос, нажав кнопку **Изменить** в сведениях о запросе.

ЗАМЕЧАНИЕ:

Чтобы изменить условия фильтра, нажмите кнопку  для очистки содержимого окна или измените непосредственно код AQL.



ЗАМЕЧАНИЕ:

Вместо использования графического инструмента можно ввести свой запрос непосредственно на языке AQL в текстовом поле в нижней части вкладки Фильтр (оператор WHERE).

Веб-клиент

Чтобы определить условие фильтра в веб-клиенте, выберите поля, функции и операторы на соответствующих вкладках.

Веб-клиент содержит следующие типы операторов, которые позволяют использовать единый интерфейс для построения выражений и определения условий фильтра

- арифметические
- реляционные
- логические



ЗАМЕЧАНИЕ:

В веб-клиенте Asset Manager имеются различные элементы запросы (например, поля и операторы), определяемые в одном текстовом поле, а не в разных полях. Важно подбирать и размещать элементы в запросе в правильной последовательности.

Этап 3: предварительный просмотр выполнения запроса (только клиент Windows)

Чтобы проверить запрос и просмотреть его преобразование на языке SQL:

- 1 Перейдите на вкладку Предварительный просмотр в сведениях о запросе.
- 2 Щелкните по значку  : Asset Manager предоставит предварительный просмотр результатов запроса в виде списка записей. Справа внизу окна показывается число записей, удовлетворяющих запросу.



ЗАМЕЧАНИЕ:

Код SQL, содержащийся на вкладке Предварительный просмотр, не может изменяться непосредственно.

Поля, используемые в запросе

При определении условий фильтра в запросе можно вызывать:

- поле в таблице, имеющее отношение к запросу;
- связанное поле;

- признаки, связанные с таблицей.

Запись выражений

Выражения  позволяют выполнять вычислительные операции в запросе. Например, можно использовать функцию **Вычислить** для подсчета числа записей - результата запроса.

Можно записать выражение двумя способами:

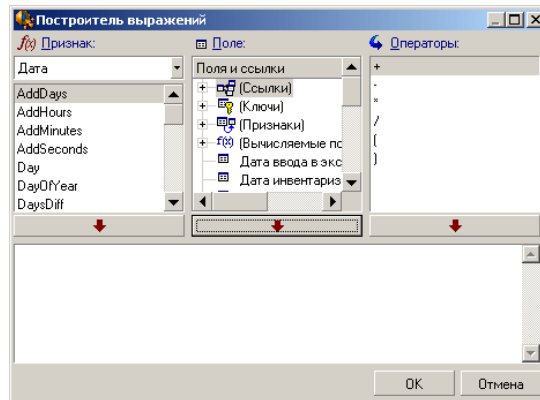
- ввести его непосредственно в соответствующее поле;
- или использовать построитель выражений Asset Manager.

Клиент Windows

В клиенте Windows построитель выражений интегрирован и доступен только в редакторе запросов.

Чтобы получить доступ к построителю запросов, в редакторе запросов на вкладке Фильтр (оператор WHERE) нажмите кнопку , расположенную рядом с ячейкой для значения поля.

Данная кнопка доступна, только если в поле (Field 1 или Field 2) внесен параметр **Выражение**.



В состав построителя выражений входят три столбца:

- Столбец Функция со списком существующих функций AQL. При щелчке по кнопке  применяется фильтр для списка функций AQL в соответствии с их типом: «Группа», «Строка», «Дата», «Число», «Тест»;
- столбец Поле со списком различных полей, которые можно использовать в запросе;

- столбец Операторы со списком операторов, которые можно использовать в выражении.

Чтобы вставить в выражение Функцию, Поле или Оператор:

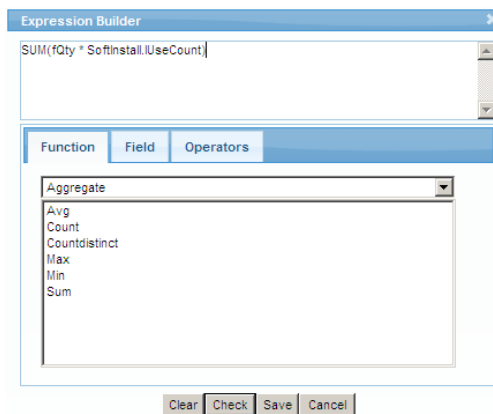
- 1 Выберите функцию, поле или оператор.
- 2 Щелкните .

После того, как выражение определено, щелкните **ОК**, чтобы передать его на вкладку Фильтр (оператор WHERE) в сведениях о запросе.

Веб-клиент

В веб-клиенте построитель выражений — это автономный инструмент, независимый от редактора запросов. Но оба инструмента имеют схожий интерфейс, функции и использование. Единственное отличие — в построителе выражений используется меньше операторов, чем в редакторе запросов.

Чтобы получить доступ к построителю запросов, щелкните значок над текстовым полем для редактирования выражений. Пстроитель выражений отобразится во всплывающем окне.



В состав построителя выражений входят три вкладки:

- Вкладка Функция содержит различные функции, группированные по категориям. Список доступных функций изменяется, если в раскрывающемся списке выбрана другая категория. Дважды щелкните функцию, чтобы вставить ее в поле ввода в верхней части построителя выражений.
- Вкладка Поле является контекстной и содержит поля и ссылки, доступные для текущей таблицы базы данных Asset Manager. Щелкните поле или ссылку, чтобы вставить ее в поле ввода.

- Вкладка **Операторы** содержит арифметические операторы, применимые к выражению. Дважды щелкните оператор, чтобы вставить его в поле ввода.

Кнопки в нижней части построителя выражений упрощают построение выражений.

- Кнопка **Очистить** очищает содержимое поля ввода без предупреждения.
- Кнопка **Проверить** проверяет синтаксис выражения. Над полем ввода отображается сообщение, которое указывает, допустимо выражение или нет.
- Кнопка **Сохранить** сначала проверяет синтаксис выражения. Если синтаксис верен, окно закрывается и выражение сохраняется в базе данных Asset Manager. В противном случае окно остается открытым и в нем отображается сообщение об ошибке.
- Кнопка **Отмена** закрывает окно без изменений.

Константы: только клиент Windows

Константы **K** представляют собой фиксированные значения, назначаемые критериям выбора. Например, если необходимо найти все модели с брендом 3Com, следует назначить постоянное значение 3Com связанному полю

Brand.Name в таблице моделей.

Чтобы выбрать константу:

- 1 Щелкните по значку .
- 2 Отобразится окно выбора со значениями, присутствующими в базе данных для поля, указанного в качестве условия поиска.



ЗАМЕЧАНИЕ:

Даже для полей типа «Подробный список» в окне, отображаемом при щелчке по значку , отображаются только значения, используемые в базе данных.

Рекомендации по созданию запросов AQL

Рекомендуется изучить данный раздел перед тем, как приступить к созданию запросов на языке AQL.

В этом разделе рассматривается:

- нотация, специфичная для языка AQL;

- специфика языка AQL и базы данных Asset Manager, оказывающая влияние на оптимальное составление запросов.

Данный раздел дополняется разделами, озаглавленными [Синтаксис AQL](#) [стр. 45] и [Справочная информация по функциям языка AQL](#) [стр. 58].



ВНИМАНИЕ:

Запросами, записанными на языке AQL, используются имена SQL (SQLName) полей, ссылок и таблицы в базе данных. Описание структуры базы данных и исчерпывающий список таких имен можно найти в файле Database.txt и Database.xml.

Этот файл находится в папке: [Asset Manager installation folder]/doc/infos. Чтобы иметь доступ к данному файлу, необходимо установить документацию.

Представление объединений AQL

Определение

Объединение суммирует несколько таблиц данных в единый запрос.

Объединения AQL

Описанием базы данных Asset Manager, кроме определения таблиц и полей, определяются связи между таблицами. Это позволяет автоматизировать создание объединений на уровне AQL.

Ссылки AQL в выражениях записываются как:

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

Упрощая объединения таким образом, с помощью языка AQL также упрощается создание большинства запросов, используемых для базы данных Asset Manager.

Пример

Приведенный ниже запрос, написанный на языке AQL, возвращает для каждой модели:

- ее **Код (IModelId)**;
- ее **Полное имя (FullName)**;
- **Имя** таблицы, связанной с брендами (**amBrand**).

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

А это тот же самый запрос, написанный для версий SQL Oracle или Microsoft SQL Server:

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

Два объединения между таблицей **Модели (amModel)** и таблицей **Бренды (amBrand)** автоматически обрабатываются в языке AQL. С помощью графического редактора запросов Asset Manager, для создания соответствующего кода AQL достаточно щелкнуть поля выделенной таблицы или связанной таблицы в списке с древовидной структурой, чтобы создать соответствующий код AQL.

Для всех систем, кроме Oracle и DB2, число внешних объединений ограничено единицей.

Для Microsoft SQL Server 7 и MSSQL 2000 можно изменить файл amdb.ini, чтобы обойти возможные проблемы при выполнении запроса. Для изменения данного файла в сведениях о соединении используется следующая инструкция:

```
UseSql92Join=1
```

Местоположение файла: ► Руководство по установке и обновлению Asset Manager, глава Файлы .ini и .cfg.

Причины применения и полезность записей первичного ключа со значением 0

Записи первичного ключа со значением «0»

Модуль данных Asset Manager имеет некоторые особенности:

- первичный и внешний ключи каждой таблицы являются полями типа «32-битное целое»;
- для внешнего ключа, который не указывает на запись, устанавливается значение «0» (а не NULL);
- каждая таблица имеет пустую запись, для первичного ключа которой установлено значение «0».

Полезность

С такими записями первичных ключей со значением «0» результаты запроса, использующего не внешнее объединение между двумя данными таблицами, A и B, могут включать записи из таблицы A, которые не связаны ни с какой реальной записью в таблице B (ссылка не заполнена). Это записи в таблице A, которые связаны с записью первичного ключа со значением «0» в таблице B.

Пример:

следующим запросом, написанным на языке AQL, для каждой позиции портфеля возвращаются метка актива, имя его пользователя и ответственного за него:

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

В результатах запроса показывается позиция портфеля, которая не назначена какому-либо пользователю и (или) ответственному. На уровне базы данных такой актив связан с записью первичного ключа со значением «0» в таблице «Сотрудники и подразделения».

Причина таких особенностей

В данном разделе объясняется, почему используются записи первичного ключа со значением «0», хотя запросом, использующим внешнее объединение SQL, могут выбираться записи в таблице A, не связанные с записью в таблице B.

Записи первичного ключа со значением «0» позволяют компенсировать то обстоятельство, что некоторыми РСУБД не обрабатываются несколько внешних объединений: благодаря записям первичного ключа со значением «0», запросами SQL, создаваемыми из запросов AQL, не используются внешние объединения.

Пример:

запросом AQL, приведенным ниже, выполняется поиск каждой позиции портфеля, его метки актива и имени местоположения его пользователя. Результаты будут включать активы, для которых нет пользователя, и активы, у пользователей которых нет местоположения.

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

Если бы созданным кодом SQL использовались внешние объединения СУБД, код SQL, созданный для определенных механизмов баз данных, был бы следующим:

```
SELECT a.AssetId, l.name FROM amPortfolio a, amEmpDept e, amLocation l  
WHERE a.UserId = e.EmpDeptId AND e.LocalId = l.LocalId
```

Этот код не поддерживается определенными механизмами баз данных, так как им используется несколько внешних объединений одно за другим.

Однако, так как в таблице подразделений и сотрудников и местоположений есть запись первичного ключа со значением «0», нет необходимости использовать внешние объединения. Asset Manager, таким образом, создает код SQL без внешних объединений:

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

Такой запрос дает ожидаемые результаты, так как ссылки **Пользователь** и **Местоположение** по-прежнему указывают на запись в таблице подразделений и сотрудников или местоположений (они указывают на запись первичного ключа со значением «0», если ссылка не заполнена).

Последствия

- Важно учитывать данные записи в составляемых запросах, особенно при использовании групповых функций.

Пример:

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

При выполнении указанного выше запроса, которым подсчитывается число активов в таблице активов, в результатах учитывается запись первичного ключа со значением «0». Поэтому, чтобы получить действительное число активов в базе данных, необходимо уменьшить число результатов на 1.

- Необходимость создавать внешние объединения на уровне СУБД возникает редко.



ЗАМЕЧАНИЕ:

Примечание: если действительно необходимо создавать внешние объединения на уровне СУБД, следует использовать операторы AQL «=*» и «*=».

Использование значения NULL

Asset Manager использует значение NULL для СУБД только в двух случаях:

- для пустого поля типа «Текст»;
- для незаполненных полей типа «Дата» или «Дата и время».

Как показано ниже, язык AQL позволяет использовать несколько эквивалентных синтаксисов. Они преобразуются им в эквивалентный действительный код SQL СУБД.

Для пустых полей типа «Текст» можно использовать любой из следующих синтаксисов, так как значение NULL будет храниться в базе данных:

WHERE <текстовое поле> = NULL

WHERE <текстовое поле> IS NULL

WHERE <текстовое поле> = "

Для незаполненных полей типа «Дата» или «Дата и время» можно использовать любой из следующих синтаксисов, так как значение NULL будет храниться в базе данных:

WHERE <поле даты или дата+час> = NULL

WHERE <поле даты или дата+час> IS NULL

WHERE <поле даты или дата+час> = []



ЗАМЕЧАНИЕ:

Примечание: если числовое поле не заполнено, его значение - «0». Аналогичным образом отсутствие ссылки описывается как «Ссылка = 0» или «внешний ключ = 0». Пример: «Location=0» или «!Locald=0».

Self

Self представляет собой выражение, эквивалентное строке описания для таблицы, для которой оно применяется.

Использование выражения Self позволяет упрощать запросы и учитывать любые настройки базы данных Asset Manager.

Пример:

если строка описания таблицы подразделений и сотрудников:

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

то запрос AQL

```
SELECT self FROM amEmplDept
```

эквивалентен запросу

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

CurrentUser

CurrentUser позволяет создавать запросы, которые зависят от того, кто подключился к базе данных.

CurrentUser может использоваться как выражение, например, в запросе, или как ссылка. Необходимо вводить данное выражение самостоятельно, так как оно не предлагается редактором запросов.

Использование в качестве выражения

Пример: поиск всех позиций портфеля, используемых сотрудником, подключенным к базе данных.

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

Использование в качестве ссылки

CurrentUser можно рассматривать как ссылку, которая находится в каждой таблице и которая указывает на запись, соответствующую текущему пользователю в таблице подразделений и сотрудников.

- В форме «CurrentUser» данная функция указывает на запись, соответствующую текущему пользователю.
- В форме «CurrentUser.Field» данная функция возвращает значение поля для текущего пользователя.

Пример: если какое-либо действие инициируется подключенным пользователем, есть возможность инициировать еще одно действие типа «сообщение», которым автоматически отправляется предупреждение подключенному пользователю. Необходимо только заполнить сведения о действии следующим образом:

Системные подробные списки

Если запросом AQL используется системный подробный список, необходимо использовать значения, хранящиеся в базе данных, а не отображаемые на экране.

Пример:

Следующим запросом выбираются контракты, для поля Тип (имя SQL: seType) которых установлено значение Основной договор лизинга.

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

Поле Тип (имя SQL: seType) является системным подробным списком. Значения, хранящиеся в базе данных:

- 0 для значения Другое;
- 1 для значения Основной договор лизинга;
- 2 для значения Договор аренды;
- 3 для значения Страхование;

- 4 для значения Обслуживание.

ЗАМЕЧАНИЕ:

Значения системного подробного списка можно найти с помощью Asset Manager Application Designer или файла database.txt, которым описывается структура базы данных.

Этот файл находится в папке: [Asset Manager installation folder]/doc/infos

Иерархические таблицы

Все иерархические таблицы содержат:

- поле «ПолноеИмя»;
- поле sLvl.

Поля «ПолноеИмя»

Для каждой записи в иерархической таблице в поле «ПолноеИмя» хранится значение поля записи, нижестоящее по отношению к древовидной структуре, состоящей из значений полей родительских записей до корневого уровня.

Значения отделяются символом «/» без пробелов. Данный символ показывается также в начале и в конце древовидной структуры.

Примеры:

- В таблице активов в поле «ПолноеИмя» хранится метка актива для данного актива, предваряемая меткой актива для своего родительского актива, в свою очередь, предваряемой меткой актива для соответствующего родительского актива, и так далее.

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

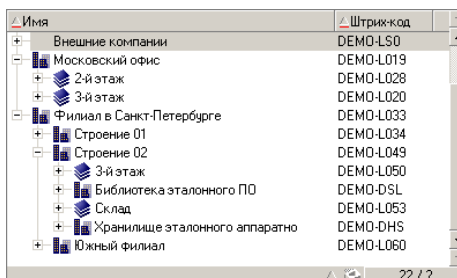
- В таблице местоположений в поле «ПолноеИмя» хранится имя местоположения, предваряемое именами родительских местоположений.

```
FullName = '/Milwaukee/Water St. Site/Building A/5th floor/'
```

поля sLvl

Для каждой записи в иерархической таблице полем sLvl указывается его уровень в древовидной структуре.

Корневой уровень - уровень 0.



Имя	Штрих-код
Внешние компании	DEMO-LS0
Московский офис	DEMO-L019
2-й этаж	DEMO-L028
3-й этаж	DEMO-L020
Филиал в Санкт-Петербурге	DEMO-L033
Строение 01	DEMO-L034
Строение 02	DEMO-L049
3-й этаж	DEMO-L050
Библиотека эталонного ПО	DEMO-D5L
Склад	DEMO-L053
Хранилище эталонного аппаратно	DEMO-DHS
Южный филиал	DEMO-L060

Следующим запросом выбираются запись Sales Head Office и ее подкомпоненты:

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

Следующим запросом выбирается запись Sales Head Office, но не ее подкомпоненты:

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

Следующим запросом выбираются подкомпоненты записи Sales Head Office, но не сама запись Sales Head Office:

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

Упрощенная нотация AQL

В данном разделе приведена нотация, используемая для упрощения записи запросов AQL.

Внешние ключи

В операторах, отличных от операторов SELECT и ORDER BY, имя SQL ссылки, после которого не следует точка, эквивалентно имени SQL связанного внешнего ключа.

Пример: оператор:

```
WHERE location = 0
```

эквивалентен

```
WHERE ILocald = 0
```

Здесь location - имя SQL ссылки «Местоположение» из таблицы подразделений и сотрудников на таблицу местоположений, а lLocald - имя SQL связанного внешнего ключа в таблице активов.

Строки описаний

В операторах SELECT и ORDER BY, имя SQL ссылки, после которого не следует точка, эквивалентно объединению <имя SQL ссылки>.self, что, в свою очередь, эквивалентно <имя SQL ссылки>.<Строка описания>.

Пример:

если строка описания таблицы подразделений и сотрудников:

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

то запрос AQL

```
SELECT user FROM amPortfolio
```

эквивалентен запросу

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

который сам по себе эквивалентен запросу

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

Признаки

Язык AQL дает прямой доступ к признакам таблицы, как если бы они были прямыми полями в таблице. Для поиска значений признака для данной таблицы необходимо только указать имя SQL данного признака (префикс fv_).

Пример: приведенным ниже запросом выполняется поиск значений признака с именем SQL **fv_WorkUnit** для таблицы **Подразделения и сотрудники (amEmplDept)**:

```
SELECT fv_WorkUnit FROM amEmplDept
```

Вычисляемые поля

Язык AQL упрощает использование связанных с таблицами вычисляемых полей.

Необходимо только указать имя SQL вычисляемого поля (префикс cf_).

Сортировка и индексы:

Язык AQL позволяет использовать две стратегии для запросов, использующих сортировку (оператор ORDER BY):

- режим, при котором Asset Manager принудительно использует индексы, указанные в запросе, если они применимы, и отображает результаты так, как и когда они извлечены;
- режим, при котором Asset Manager не использует принудительно индексы, указанные в запросе. В таком случае сортировку данных определяет СУБД.

Пример

В случае следующего запроса:

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

- Доступ без принудительного использования индексов: СУБД сканирует таблицу полностью, без использования индекса «Бренд», указанного в запросе. Находятся все элементы данных, удовлетворяющие запросу, сортируются в соответствии с индексом «Бренд» и отправляются пользователю. Результаты будут отображены только по прошествии определенного периода времени.
- В противоположном случае СУБД использует индекс «Бренд» и отображает результаты в том виде, в каком и когда они найдены. Первые элементы данных, таким образом, показываются быстро, но полное время обработки оказывается больше.

Принудительное использование индексов

Способ принудительного использования индексов зависит от способа создания запроса.

С помощью меню настройки списка

В Asset Manager можно настроить тип доступа к данным для каждого списка. Это возможно и для основных списков, и для списков на вкладках. Для этого:

- 1 Перейдите к списку, который необходимо настроить.
- 2 Щелкните правой кнопкой мыши.
- 3 Выберите пункт **Сервис/Настройка списка** из контекстного меню.
- 4 На вкладке Столбцы и сортировка установите флажок Выполнить принудительное индексирование, чтобы использовать индексы, указанные в запросе, и отображать результаты в том виде, в каком и когда они были созданы. Снимите данный флажок, чтобы выбрать другой тип доступа.

Использование AQL

Если запрос записывается непосредственно на языке AQL, можно указать принудительное использование индексов с помощью оператора FIRST_ROWS.
Пример:

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

 **ЗАМЕЧАНИЕ:**

Если сортировка используется для системных подробных списков (например, для таблицы признаков в поле **seDataType**), существует возможность, что при принудительном использовании индекса сортировка окажется недействительной.

Порядок сортировки

- На порядок сортировки влияют:
- СУБД;
 - использование индексов.

Для Oracle for WorkGroups

С принудительным использованием индексов

- Не показываются записи со значением NULL.
- Сортировка выполняется в соответствии со значением кодов ASCII, учитывая, таким образом, разницу между символами верхнего и нижнего регистров (двоичная сортировка).

Без принудительного использования индексов

- Отображаются записи со значением NULL.
- Oracle for WorkGroups нечувствителен к регистру.

Пример

Сортировка

Исходный список	A B C D a b NULL NULL
Список с принудительным использованием индексов	A B C D a b

В Microsoft SQL Server

Порядок сортировки зависит от параметра, устанавливаемого при создании базы данных. При настройке можно указать зависимость от регистра или учет символов с акцентами и так далее.

Меры предосторожности

В случае сложных запросов часто бывает сложно определить, повысится ли производительность при принудительном использовании индексов или нет. На практике рекомендуется выполнить проверку на производительность перед принятием окончательного решения.

Особенно рекомендуется выполнять проверку с принудительным использованием индексов и без в случае списка, который фильтруется, явно (с помощью простого фильтра, запроса) или неявно (с помощью ограничений доступа).

Синтаксис AQL

Для использования AQL необходимо знакомство с языком SQL. Однако подробное описание синтаксиса SQL выходит за рамки данного руководства. Более подробную информацию можно найти в соответствующей справочной документации.

Соглашения

Ниже приведены соглашения, используемые для описания синтаксиса AQL.

Таблица 3.1. AQL - соглашения о синтаксисе

[]	В такие скобки заключаются дополнительные элементы. Не следует печатать в этих скобках.
< >	В такие скобки заключаются логические элементы. Не следует печатать в этих скобках.
	Вертикальная черта указывает, что варианты выбора взаимоисключающие.
...	Многоточие указывает, что предыдущий текст может повторяться один или несколько раз.

Синтаксис запросов

Простые запросы

SELECT [DISTINCT] [FIRST_ROWS] <Список выбора>

[оператор FROM [стр. 52]]

[Оператор WHERE [стр. 53]]

[Оператор GROUP BY [стр. 54]]

[Оператор HAVING [стр. 55]]

[Оператор ORDER BY [стр. 56]]

Подзапросы

Языком AQL поддерживается использование подзапросов вместо полей.



ЗАМЕЧАНИЕ:

В подзапросах для инструкции **SELECT** разрешается только одно выражение.

(**SELECT** [DISTINCT] <выражение>

[оператор FROM [стр. 52]]

[Оператор WHERE [стр. 53]]

[Оператор GROUP BY [стр. 54]]

[Оператор HAVING [стр. 55]]

)



ВНИМАНИЕ:

Подзапросы должны заключаться в скобки.

Пример использования:

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

Запросы типа UNION

Параметр **UNION** позволяет группировать вместе результаты нескольких запросов:

SELECT <список выбора>

[оператор **FROM** [стр. 52]]

[Оператор **WHERE** [стр. 53]]

[Оператор **GROUP BY** [стр. 54]]

[Оператор **HAVING** [стр. 55]]

[**UNION** | **UNION ALL** | **INTERSECTS** | **MINUS**]

SELECT <список выбора>

[оператор **FROM** [стр. 52]]

[Оператор **WHERE** [стр. 53]]

[Оператор **WHERE** [стр. 53]]

[Оператор **GROUP BY** [стр. 54]]

[Оператор **HAVING** [стр. 55]]...

[Оператор **ORDER BY** [стр. 56]]

Элементы запроса

Поля и ссылки

Запросы выполняются относительно полей и ссылок в базе данных Asset Manager.

Можно указывать имя поля:

- в ссылке на начальную таблицу запроса. В таком случае нет необходимости указывать имя данной таблицы:

[Link. ...[Link.]]<field>

Примеры из таблицы **Позиции портфеля (AmPortfolio)**:

Model
User.Name
User.Location.Name

- В качестве абсолютной ссылки. В таком случае необходимо указать имя таблицы, которой принадлежит поле.
 - Или объявить таблицу в операторе **FROM** и использовать ее имя (или возможный псевдоним):
<таблица.[ссылка...]поле>
<псевдоним.[ссылка...]поле>
 - Или не объявлять таблицу в операторе **FROM** и вместо этого использовать двоеточие «:»:
<таблица.[ссылка...]поле>

<таблица[_псевдоним]:[ссылка[_псевдоним]...]поле>

Две последние нотации особенно полезны, если нельзя использовать оператор **FROM**.

Например, при записи запроса в Asset Manager есть доступ только к оператору **WHERE**. Начальная таблица запроса является неявной (таблица, для которой применяется фильтр, поле Таблица (**TableName**) в сведениях о запросе и так далее). Однако может понадобиться использовать в запросе другие таблицы. В таком случае, это позволяет сделать нотация «:».

Константы

Для констант, которые могут использоваться в запросе, действителен следующий синтаксис.

Числовые константы

В качестве десятичного разделителя используется точка.

Примеры:

12

52.23

Константы текстового типа

Такие константы заключаются в одинарные кавычки.

Примеры:

'Computer'

'Monitor'

Константы типа даты или времени

Константы типа даты или времени заключаются между символами #. Их формат должен удовлетворять следующим правилам:

- год записывается 4 цифрами;
- даты записываются как год-месяц-день;
- время записывается как часы-минуты-секунды;
- для времени используется формат 24 часов (а не 12 часов);
- используемый для дат разделитель - символ «/» или «-»;
- используемый для времени разделитель - символ «:»;
- месяцы, дни, часы, минуты и секунды записываются 2 цифрами;
- при совместной записи даты и времени дата всегда стоит впереди времени и отделяется пробелом.

Примеры:

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

#yyyy-mm-dd#
#hh:mm:ss#
#2004-01-01 01:00:03#

Переменные

Вместо указания константы (► [Константы](#) [стр. 48]) можно ввести переменную, значение которой заполняется пользователем при выборе фильтра, использующего этот запрос.

Для определения переменных может использоваться разный синтаксис:

- [\\$Variable](#) [стр. 49]
- [amDbVal\(\)](#) [стр. 49]

[\\$Variable](#)

Используйте следующий синтаксис:

`<имя SQL поля или ссылки> = $<Имя переменной>`

Пользователь фильтра запроса должен заполнить поле **<Имя переменной>** перед выполнением запроса.

<Имя переменной> не должно содержать пробелов.

[amDbVal\(\)](#)

Используйте следующий синтаксис:

`amDbVal(<Имя SQL поля или ссылки>,'<Метка>','<Значение по умолчанию>')`

Пользователь фильтра запроса должен заполнить поле **<Метка>** перед выполнением запроса.

В поле, которое необходимо заполнить, по умолчанию отображается **<Значение по умолчанию>**. **<Значение по умолчанию>** должно использовать синтаксис, описанный в разделе ► [Константы](#) [стр. 48].

Если **<Имя SQL поля или ссылки>** назначает ссылку, **<Значение по умолчанию>** должно содержать основной ключ связанной записи.

Если **<Имя SQL поля или ссылки>** назначает системный подробный список, **<Значение по умолчанию>** должно содержать число, сохраненное в базе данных.

Выражения

При составлении выражений используются:

- Константы
- Поля
- Функции

- Подзапросы

Можно комбинировать данные элементы с операторами и скобками, чтобы составлять сложные выражения.

Выражения сравнения имеют вид:

<выражение> <оператор сравнения> <выражение>

Логические выражения имеют вид:

<оператор сравнения> <AND | OR> <оператор сравнения>

Можно объединять несколько логических выражениях с помощью скобок.

Операторы

Логические операторы

Логические операторы применяются для связи двух выражений:

Таблица 3.2. AQL - логические операторы

Оператор	Значение
AND	Логическое AND
OR	Логическое OR

Чтобы оптимизировать запрос, иногда оказывается рациональным избежать использования логических операторов, если вместо них можно использовать оператор сравнения. Следующим примером проиллюстрировано, как оптимизировать фильтр запросов, используемый для выбора позиций портфеля, для поля Назначение (имя SQL: seAssignment) которых установлено значение Ожидает доставки или Возврат для обслуживания. Значения данных двух элементов системного подробного списка - «3» и «4» соответственно. Поэтому можно записать:

(seAssignment=3) OR (seAssignment =4)

Так как «4» - последнее значение в системном подробном списке, предпочтительнее записать данный запрос как

seAssignment >=3

Операторы сравнения

Операторы сравнения применяются для сравнения двух выражений.

Таблица 3.3. AQL - операторы сравнения

Оператор	Значение
=	Равно
<>	Отлично от
=!	
>	Больше, чем
<	Меньше, чем
>=	Больше или равно
=<	Меньше или равно
=*	Правое внешнее объединение. Из-за способа, которым языком AQL обрабатываются ссылки, использование данного оператора ограничено.
*=	Левое внешнее объединение. Из-за способа, которым языком AQL обрабатываются ссылки, использование данного оператора ограничено.
LIKE	Выполняется аналогично оператору «=» и позволяет также использовать подстановочные знаки.
NOT LIKE	Доступны следующие подстановочные знаки: «%» замещает любую строку символов; «_» замещает любой один символ; В зависимости от используемого механизма базы данных (SQL Server поддерживает его, Oracle for WorkGroups - нет): [abc?] позволяет определить список возможных символов (без пробелов между возможными значениями); [a-c] позволяет определить диапазон возможных значений. DB2 не поддерживает использование оператора LIKE X, если X включает имя столбца SQL. Для данного оператора поддерживаются только константы. Например, для DB2 следующий запрос неверен: SELECT COL1, COL2 FROM TABLE1 WHERE COL1 LIKE COL2
IS NULL	Проверяется, значение поля NULL или нет.
IS NOT NULL	Asset Manager разрешает только значение NULL для пустых текстовых полей и для незаполненных полей Дата или Дата+Время .

Операторы, специфические для подзапросов

Можно сравнить какое-либо значение с результатами подзапроса с помощью следующих операторов:

- = ANY (подзапрос)
- = ALL (подзапрос)
- = SOME (подзапрос)

Пример:

- Следующим запросом экспорта Asset Manager предоставляется список позиций портфеля, бренд которых используется в месте размещения Milwaukee. (Данный запрос не включает суб-местоположений):

```
SELECT DISTINCT IModelId, Model.Brand FROM amPortfolio WHERE Model.Brand = ANY (SELECT Model.IBrandID FROM amPortfolio WHERE Location.FullName LIKE '/Milwaukee site')
```

Для данного скрипта следующая связанная запись вызывает использование внешнего ключа данной записи.

Список выбора

Списками выбора определяются позиции, которые необходимо извлечь или отобразить. Они указываются инструкциями SELECT в запросах.

Список выбора состоит из одного или нескольких выражений, разделенных запятыми:

<выражение> [, <выражение> ...]

Каждое выражение может связываться с псевдонимом. Например:

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

Это особенно полезно на уровне запросов экспорта, чтобы назначить имя экспортируемому столбцу.



ЗАМЕЧАНИЕ:

Некоторыми СУБД ограничивается число выражений в данной инструкции SELECT.

оператор FROM

Оператором **FROM** указываются таблица или таблицы, с которыми связана инструкция **SELECT**.

Языком AQL разрешается использование псевдонимов для имен таблиц.

Синтаксис

FROM <имя таблицы> [псевдоним таблицы] [, <имя таблицы> [псевдоним таблицы] ...]

Начальная таблица запроса

Первая таблица, указываемая в операторе **FROM** запроса, является начальной таблицей.

Если запросом используется поле, для которого не указана таблица, в языке AQL принимается, что оно принадлежит начальной таблице запроса. Оператор

FROM языка AQL отличается от оператора с таким же именем в языке SQL.

Например, в следующем подзапросе на языке AQL выполняется поиск поля **AssetTag** в таблице amAsset:

```
SELECT AssetTag FROM amAsset
```

Число таблиц в запросе

Число таблиц, разрешенное для использования в запросе, зависит от используемой СУБД.

Пример:

- Oracle: можно использовать любое необходимое количество таблиц;
- Microsoft SQL Server: разрешенное число таблиц в запросе ограничено 16.



ВНИМАНИЕ:

При подсчете числа таблиц в запросе надо обязательно учитывать таблицы, не упомянутые явно, в частности, если в запросе используются ссылки. Также необходимо обращать внимание на нотацию fv_ (поиск значений признаков), которой создается дополнительное объединение на уровне СУБД. Аналогично могут создаваться дополнительные объединения нотацией cf_ (вычисляемые поля).

Примеры

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

Следующие запросы эквивалентны:

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

Оператор WHERE

Оператор **WHERE** языка AQL эквивалентен оператору с таким же именем в языке SQL.

Им указываются условия поиска, определяющие элементы, которые необходимо извлечь из базы данных. Данные условия могут также выражаться в операторах **HAVING**.

Синтаксис

WHERE <условия поиска>

Запись условий поиска

В большинстве случаев необходимо записывать данные условия с помощью следующей формы:

```
<WHERE | HAVING> [NOT] <выражение> <оператор сравнения> <выражение>  
<WHERE | HAVING> [NOT] <логическое выражение> <WHERE | HAVING>  
[NOT] <поле> [NOT] LIKE 'xxxxx' <WHERE | HAVING> [NOT] <логическое выражение>  
<AND | OR> <логическое выражение> <WHERE | HAVING> [NOT] <поле> IS [NOT] NULL
```

В некоторых случаях может понадобиться записать более сложные запросы, например:

```
<WHERE | HAVING> [NOT] EXISTS (<подзапрос>) <WHERE | HAVING> [NOT]  
] <выражение> [NOT] IN (<список значений> | <подзапрос>) <WHERE | HAVING> [NOT]  
] <выражение> <оператор сравнения> <ANY | ALL> (<подзапрос>)
```

Оператор GROUP BY

Оператор **GROUP BY** языка AQL эквивалентен оператору с таким же именем в языке SQL.

Синтаксис

GROUP BY <выражение без агрегатов> [, <выражение без агрегатов>]...

Рекомендации по записи выражений

Оператором **GROUP BY** указываются поднаборы таблицы. Поднаборы определяются в операторе **GROUP BY** выражением, которое может быть, например, именем поля.

Если в список выбора инструкции **SELECT** включены групповые функции, оператором **GROUP BY** выполняется поиск результирующего значения для каждого поднабора. Данные результирующие значения могут использоваться в операторе **HAVING**.

Если в запросе используется оператор **GROUP BY**, каждым выражением в списке выбора должно обеспечиваться по одному значению для каждого поднабора.

Оператор GROUP BY - Пример

Приведенным ниже запросом возвращается полное число брендов, присутствующих в базе данных. Для каждого актива со связанным брендом Asset Manager возвращает экземпляр бренда.

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

С помощью оператора **GROUP BY** получается список брендов и число активов для каждого бренда:

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

Оператор **HAVING**

Оператор **HAVING** языка AQL эквивалентен оператору с таким же именем в языке SQL.

Синтаксис

HAVING <условия поиска>

Отличия от оператора **WHERE**

Условия поиска указываются им также, как и оператором **WHERE**. Однако два данных оператора имеют следующие отличия:

- Оператором **HAVING** указываются ограничения, которые должны применяться для групповых функций в списке выбора. Ограничения влияют на число результирующих строк, но не на расчеты, связанные с групповыми функциями.
- Если в запросе используется оператор **WHERE**, условия поиска накладывают ограничения на строки, указанные для расчетов с помощью групповых функций, но не влияют на результирующие строки.

Примеры

Пример запроса, в котором оператор **WHERE** эквивалентен оператору **HAVING**:

Приведенным ниже запросом возвращается список брендов, имя которых начинается с буквы после буквы B, и число активов для каждого из этих брендов:

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

Можно также выразить тот же самый запрос с помощью оператора **WHERE**:

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

Пример запроса, в котором используется оператор **HAVING**:

Оператор **HAVING** позволяет использовать групповые функции (например, **Count**); это не так в случае оператора **WHERE**. Таким образом, приведенным ниже запросом возвращаются все бренды, представленные несколькими активами:

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

Оператор ORDER BY

Оператор **ORDER BY** языка AQL эквивалентен оператору с таким же именем в языке SQL.

Позиции могут сортироваться:

- В восходящем порядке: **ASC**. Это порядок сортировки по умолчанию.
- В нисходящем порядке: **DESC**.

Синтаксис

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

Оператор INSERT

Этот оператор позволяет вставить одну или несколько записей в таблицу базы данных.

Синтаксис

INSERT INTO <Имя таблицы> [псевдоним таблицы] (<Имя поля> [, <Имя поля>]...) **VALUES** (<выражение> [, выражение]...) | Подзапрос AQL

Данный оператор включен в интерфейс API Asset Manager AmDbExecAql.

Дополнительные сведения об интерфейсе API в Asset Manager можно найти в руководстве Programmer's reference, глава Alphabetic reference («Алфавитный указатель»).

Пример

С помощью оператора **INSERT** можно упростить программный код мастера. Дополнительные сведения о поставке:

Программный код мастера без использования оператора INSERT

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



```

|Err = AmSetFieldLongValue(hrAlarm, "dtTrig1", AmGetFieldLongValue(hrAsset,
2)-IDaysBefore*86400)
|Err = AmSetFieldLongValue(hrAlarm, "IAction1Id", IActionId)
|Err = AmSetFieldLongValue(hrAlarm, "IMonitObjId", IAstId)
|Err = AmSetFieldStrValue(hrAlarm, "MonitoredField", "dWarrEnd")
|Err = AmSetFieldStrValue(hrAlarm, "MonitoredTable", "amAsset")
|Err = AmSetFieldLongValue(hrAlarm, "sDaysBefore1", IDaysBefore)
|Err = AmInsertRecord(hrAlarm)

```

Программный код мастера с использованием оператора **INSERT**

```

|Err = AmDbExecAql("insert into amDateAlarm (bSecondLevel, dtTrig1, IActionId,
IMonitObjId, MonitoredField, MonitoredTable, sDaysBefore1) values ( 0, " & AmG
etFieldLongValue(AmGetFieldLongValue(hrAsset, 2)-IDaysBefore*86400 & ", " &
IAstId & ", 'dWarrEnd', 'amAsset', " & IDaysBefore & ")")

```

оператор **UPDATE**

Этот оператор позволяет обновлять одно или несколько полей записей в таблице базы данных.

Синтаксис

UPDATE <имя таблицы> [псевдоним таблицы] **SET** (<имя поля> [, <имя поля>...]) [[оператор FROM](#) [стр. 52]] [[Оператор WHERE](#) [стр. 53]]

Пример

Оператор **UPDATE** может помочь упростить программный код действия, которым инициируется действие заказа:

Код действия без использования оператора **UPDATE**

```

|hr = AmGetRecordFromMainId("amPOrder", [IPOrdId])
|Err = AmSetFieldLongValue(hr, "seStatus", "${IDS_POSTATUS_ORDERED}")
|Err = AmUpdateRecord(hr)

```

Код действия с использованием оператора **UPDATE**

```

|Err = AmDbExecAql("update amPOrder set seStatus = 21 where IPOrdId = " &
[IPOrdId])

```

оператор **DUPLICATE**

Этот оператор позволяет дублировать запись, существующую в таблице базы данных.

Данная функция является особенностью Asset Manager.

Дополнительную информацию можно найти в руководстве Интерфейс пользователя, глава Операции с записями, раздел Дублирование записи.

Синтаксис

DUPLICATE <имя таблицы> [псевдоним таблицы] **SET** (<имя поля> [, <имя поля>...]) [[оператор FROM](#) [стр. 52]] [[Оператор WHERE](#) [стр. 53]]

оператор **DELETE**

Этот оператор позволяет удалять поля записей в таблице базы данных.

Синтаксис

DELETE [[оператор FROM](#) [стр. 52]] [[Оператор WHERE](#) [стр. 53]]

Справочная информация по функциям языка AQL

В запросах и формулах можно использовать следующие функции языка AQL:

- Функции AQL группового типа
- Функции AQL строкового типа
- Функции AQL типа даты
- Функции AQL числового типа
- Функции AQL проверочного типа



ЗАМЕЧАНИЕ:

Также можно использовать собственные функции SQL из СУБД. В таком случае, получающийся в результате код не будет переносимым.

Функции AQL группового типа

Таблица 3.4. AQL - функции группового типа

Функция	Описание
Avg(<столбец>)	Возвращается среднее значение столбца числового типа. Возвращает «0», если столбец не содержит никаких записей.
Count(<Столбец>)	Подсчитываются ненулевые значения в столбце.
Countdistinct(<Столбец>)	Подсчитываются отличающиеся ненулевые значения в столбце.
Max(<столбец>)	Возвращается максимальное значение столбца типа «Число», «Текст» или «Дата». Если в столбце нет никаких записей, возвращается значение «0» (столбец типа «Число»), «пустая строка» (столбец типа «Текст»), или «пустая дата» (столбец типа «Дата»).
Min(<Столбец>)	Возвращается минимальное значение в столбце типа «Число», «Текст» или «Дата». Если в столбце нет никаких записей, возвращается значение «0» (столбец типа «Число»), «пустая строка» (столбец типа «Текст») или "пустая дата" (столбец типа «Дата»).
Sum(<Столбец>)	Возвращается сумма значений столбца числового типа. Возвращает «0», если столбец не содержит никаких записей.

Данными функциями вместе используются операторы GROUP BY и HAVING

Функции AQL строкового типа

Таблица 3.5. AQL - функции строкового типа

Функция	Описание
Ascii(<строка>)	Возвращается значение ASCII первого символа <строки>.
Char(<n>)	Возвращается символ с кодом ASCII «n».
Length(<Строка>)	Возвращается длина <строки>.
CharIndex(<Строка1>, <Строка2>)	Возвращается позиция <строки1> внутри <строки2>. Первый символ в <string1> находится в положении 1. Если <string1> не встречается в <string2>, функция возвращает 0.
DateToText(<Дата>)	<Дата> преобразовывается в данные текстового типа (в строку, которой описывается дата в стандартном формате ISO)
EmptyString()	Создается ненулевая пустая строка.
LikeParam(<Поле>)	Используется при сравнении двух полей с помощью операнда Like языка AQL. Пример: Field1 like LikeParam(Field2)

Функция	Описание
Left(<Строка>, <n>)	Возвращаются первые n символов <строки>.
Lower(<Строка>)	Возвращается <строка> в нижнем регистре.
Ltrim(<Строка>)	Удаляются пробелы слева от <строки>.
NullBlob()	Создается объект Blob со значением NULL.
ЗАМЕЧАНИЕ: Программно он представляет собой значение NULL, приведенное к типу объекта Blob.	
NullMemo()	Создается объект «Заметка» со значением NULL.
ЗАМЕЧАНИЕ: Программно он представляет собой значение NULL, приведенное к типу объекта «Заметка».	
NullString()	Создается объект «Строка» со значением NULL.
ЗАМЕЧАНИЕ: Программно она представляет собой значение NULL, приведенное к типу объекта «Строка».	
NumberToText(<Число>)	<Число> преобразовывается в данные текстового типа.
Right(<Строка>, <n>)	Возвращаются последние n символов <строки>
Rtrim(<Строка>)	Удаляются пробелы справа от <строки>.
Substring(<Строка>, <n1>, <n2>)	Извлекается подстрока, начинающаяся с символа n1 в <строке> и с длиной n2 (первый символ <строки> рассматривается как символ номер 1).
TimeStampToText(<Дата+время>)	Преобразовывается объект типа Дата+Время в данные текстового типа, соответствующие следующему международному стандарту: YYYY-MM-DD HH24:MI:SS
TimeToText(<Время>)	Преобразовывается объект типа времени в данные текстового типа, соответствующие следующему международному стандарту: HH:MM:SS
Upper(<Строка>)	Возвращается <строка> в верхнем регистре.

Функции AQL типа даты

Таблица 3.6. AQL - функции типа даты

Функция	Описание
AddDays(<дата>, <число>)	Добавляется указанное число дней к полю типа «Дата» или «Дата и время».

Функция	Описание
AddHours(<дата>, <число>)	Добавляется указанное число часов к полю типа «Дата» или «Дата и время».
AddMinutes(<дата>, <число>)	Добавляется указанное число минут к полю типа «Дата» или «Дата и время».
AddSeconds(<дата>, <число>)	Добавляется указанное число секунд к полю типа «Дата» или «Дата и время».
Day(<дата>)	Возвращается число дней в месяце для поля типа «Дата» или «Дата и время» (1-31).
DayOfYear(<дата>)	Возвращается число дней в году для поля типа «Дата» или «Дата и время» (1-366).
DaysDiff(<дата1>, <дата2>)	Число дней между датами дата1 и дата2 (десятичное число с плавающей запятой)
DbToLocalDate(<дата>)	Преобразовывается значение даты, выраженное для часового пояса сервера базы данных, в значение даты, выраженное для часового пояса, определенного на уровне клиентского компьютера.
Getdate()	Возвращается текущая системная дата сервера.
Hour(<время>)	Возвращается число часов в дне для поля типа «Время» или «Дата и время» (0-23).
HoursDiff(<дата1>, <дата2>)	Число часов между датами дата1 и дата2 (десятичное число с плавающей запятой)
LocalToDbDate(<дата>)	Преобразовывается значение даты, выраженное для часового пояса клиентского компьютера, в значение даты, выраженное для часового пояса сервера базы данных.
Minute(<время>)	Возвращается число минут для поля типа «Время» или «Дата и время» (0-59).
MinutesDiff(<дата1>, <дата2>)	Число минут между датами дата1 и дата2 (десятичное число с плавающей запятой)
Month(<дата>)	Возвращается число месяцев для поля типа «Дата» или «Дата и время» (1=январь, ..., 12=декабрь).
NullDate()	Создается объект «Дата» со значением NULL.
ЗАМЕЧАНИЕ: Программно он представляет собой значение NULL, приведенное к типу объекта «Дата».	
NullTime()	Создается объект «Время» со значением NULL.
ЗАМЕЧАНИЕ: Программно он представляет собой значение NULL, приведенное к типу объекта «Время».	
NullTimeStamp()	Создается объект «Дата и время» со значением NULL.
ЗАМЕЧАНИЕ: Программно он представляет собой значение NULL, приведенное к типу объекта «Дата и время».	

Функция	Описание
NumberToTime(<число>)	Преобразовывается указанное число в дату типа «Дата и время».
Second(<время>)	Возвращается число секунд для поля типа «Время» или «Дата и время» (0-59).
SecondsDiff(<дата1>, <дата2>)	Число секунд между датами дата1 и дата2 (десятичное число с плавающей запятой)
TextToTime(<текст>, <формат>, <язык>)	Преобразовывается текст в дату. Дополнительные (и необязательные) параметры языка и формата дают доступ к форматам конкретной базы данных и позволяют указывать ожидаемый формат для преобразования и настройки поддержки языка. Дополнительные сведения см. в документации по конкретной базе данных.
WeekDay(<дата>)	Возвращается число дней в неделе для поля типа «Дата» или «Дата и время». Данное число зависит от того, каким образом сконфигурирован сервер. Например, конфигурация по умолчанию для Microsoft SQL Server: 1=воскресенье, 2=понедельник, ..., 7=суббота. Конфигурация по умолчанию для Oracle: 1=понедельник, ..., 7=суббота.
Year(<дата>)	Возвращается число, представляющее собой год, для поля типа «Дата» или «Дата и время» (например, 2000).

Таблица 3.7. AQL - примеры использования функции типа даты

Описание	Язык запросов Asset Manager
Все записи, измененные за последнюю неделю	AddDays(dtLastModif,7)>=Getdate()
Все заказы на работу, о которых было известно за последний час	HoursDiff(Getdate(), dtNotif) <= 1 или AddHours(dtNotif, 1) >= Getdate()
Все заказы на работу, закрытые за последние полчаса	MinutesDiff(Getdate(), dtActualFixed) <= 30 или AddMinutes(dtActualFixed, 30) >= Getdate()

Приведенным ниже запросом возвращается список заказов на работу, фактически выполненных и завершенных в течение того же дня. Учитывается часовой пояс клиентского компьютера.

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

Следующим запросом возвращается список всех заказов на работу, фактически начатых сегодня:

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

Функции AQL числового типа

Таблица 3.8. AQL - функции числового типа

Функция	Описание
Abs(<Число>)	Возвращается абсолютное значение числа.
Ceil(<Число>)	Возвращается наименьшее целое число, большее указанного числа или равное ему.
DataLength (<Данные>)	Возвращается длина <Данных> в байтах.
Floor(<Число>)	Возвращается наибольшее целое число, меньшее указанного числа или равное ему.
Length(<Данные>)	Возвращается длина <Данных> в символах.
Mod(<a>,)	Возвращается остаток от деления «a» на «b» ($a = qb + r$, где q - целое число и $0 < r < q$).
NullNumeric()	Создается объект «Число» со значением NULL.
ЗАМЕЧАНИЕ: Программно он представляет собой значение NULL, приведенное к типу объекта «Число».	
NumberToNumber(<Число>)	Позволяет выполнять преобразование между числами различных типов, если оно не выполняется на уровне РСУБД.
Round(<a>, <n>)	Округляет «a» до «n» десятичных знаков.
Sign(<Число>)	Позволяет определять знак параметра <Число>: ■ Если функция возвращает 1, <Число> положительно. ■ Если функция возвращает -1, <Число> отрицательно. ■ Если функция возвращает 0, <Число> равно нулю.
TextToNumber (<Текст>)	<Текст> преобразовывается в данные числового типа.
Trunc(<a>, <n>)	Усекает «a» до «n» десятичных знаков.

Примеры применения:

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.

Функции AQL проверочного типа

Таблица 3.9. AQL - функции проверочного типа

Функция	Описание
IsNull(<a>,)	Если «а» имеет значение «Null», «а» замещается на «b». Типы «а» и «b» должны быть совместимыми.

Примеры запросов:

Каждый из приведенных ниже примеров относится к конкретному аспекту составления запросов. Можно изменять или объединять данные примеры, чтобы использовать их как основу для своих собственных запросов.

В данных примерах приводится полный синтаксис запросов. Для проверки их в неизменном виде необходимо использовать Asset Manager Export Tool. Необходимо изменить синтаксис данных примеров, чтобы использовать их, например, в фильтре запросов.

Таким образом, полный запрос:

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

при использовании в фильтре запроса (когда в явной форме дается только оператор WHERE) для таблицы активов запишется следующим образом:

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

Дополнительные примеры запросов включены в демонстрационную базу данных, поставляемую с Asset Manager.

ЗАМЕЧАНИЕ:

Для просмотра преобразования запроса в соответствующий код SQL используемой СУБД необходимо отобразить вкладку Предварительный просмотр в сведениях о запросе.

Сравнение поля в основной таблице с каким-либо значением

Пример: все позиции портфеля с брендом Compaq.

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


Сравнение ссылки в основной таблице с другой ссылкой

Пример: все позиции портфеля с тем же местоположением, что и их родительский актив.

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

Сравнение ссылки в основной таблице с каким-либо значением

Пример: все подразделения и сотрудники, непосредственно связанные с компанией Burbank Agency.

```
SELECT Self FROM amEmplDept WHERE Parent.Name = 'Burbank Agency'
```

Сравнение на соответствие полю в таблице, связанной с основной таблицей

Пример: все позиции портфеля с тем же именем местоположения, что и их родительские позиции.

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

Иерархические таблицы

Использование поля «FullName (ПолноеИмя)»

Пример: все суб-местоположения местоположения с именем Ariane Building:

```
SELECT Self FROM amLocation WHERE FullName LIKE '/Ariane Building/%'
```

Использование полей «FullName (ПолноеИмя)» и «sLvl»

Запросами к иерархическим таблицам часто используются поля FullName и sLvl.

Пример: все суб-местоположения местоположения Ariane Building с уровнем иерархии меньше 3.

Значение иерархии корневого уровня древовидной структуры равно 0.

```
SELECT Self FROM amLocation WHERE (FullName LIKE '/Ariane Building/%') A  
ND (sLvl < 3)
```

Необходимо обратить особое внимание на символы «/», которые должны ставиться в начале и в конце полных имен.

Запрос, объединяющий два условия

Пример: все сотрудники с должностью Account executive и местом работы Burbank site.

```
SELECT Self FROM amEmplDEpt WHERE (Title = 'Commercial') AND (Location  
.Name = 'Madison site')
```

Сравнение поля с числами, датами или текстовыми величинами

Например: все заказы на работу, выполненные между 1 января 2006 года и 31 декабря 2006 года.

```
SELECT self FROM amWorkOrder WHERE (dtActualFixStart >= #2006-01-01 00  
:00:00#) AND (dtActualFixStart <= #2006-12-31 00:00:00#)
```

Запрос относительно признаков

Пример: все позиции портфеля, для которых значение признака с именем SQL **fv_Size** больше или равняется 150 см.

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

Поиск записей, соответствующих выражению

Пример: все активы, покупная цена которых равна наибольшей покупной цене, содержащейся в базе данных. Примечание: в основном запросе используется подзапрос для выявления максимальной цены.

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

Поиск незаполненного поля

Пример: все сотрудники, не имеющие номера телефона. Примечание: пустая строка представлена двумя одинарными кавычками ' '.

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

Поиск отсутствия ссылки

Случай с одной ссылкой

Пример: все позиции портфеля, не назначенные какому-либо пользователю.

Примечание: отсутствие ссылки обозначается нулем (0).

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

Случай с несколькими ссылками

Пример: все модели, с которыми не связаны какие-либо активы:

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

Данным запросом сканируется таблица моделей, и для каждой модели, одной за другой, сравнивается с 0 число активов, относящихся к данной модели.

Пример, в котором объединяется проверка для одной ссылки и для нескольких ссылок

Пример: все модели без родительских модели или подмодели:

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

Данным запросом выполняются:

- Проверка для одной ссылки (Parent = 0), чтобы выбрать модели без родительских активов.
- Проверка для нескольких ссылок ("0 = (SELECT COUNT(a.IModelId) FROM amModel a WHERE a.IParentId = IModelId)"), чтобы выбрать модели без подмоделей. При проверке для нескольких ссылок для каждой модели выбирается ее идентификатор IModelId и подсчитываются все модели, идентификатор IParentId которых равен IModelId.



ЗАМЕЧАНИЕ:

Оператором **SELECT COUNT** подсчитываются все записи базы данных; это больше влияет на производительность, чем в случае оператора **EXIST**.

Еще один пример

Все модели без подмодели с видом «Компьютер»:

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



ЗАМЕЧАНИЕ:

Если попытаться использовать такой запрос с Asset Manager Export Tool, появится сообщение об ошибке. Можно проигнорировать данное сообщение. На самом деле этот запрос выполняется правильно.

Запрос с псевдонимом

Пример: все сотрудники, проходящие программы обучения HP Software и Database.

Начальная таблица: таблица подразделений и сотрудников.

Запрос при этом следующий:

```
SELECT Self FROM amEmplDept WHERE (Trainings_1.Title = 'HP Software') AND (Trainings_2.Title = 'Database')
```

Псевдонимы, обозначенные как Training_1 и Training_2, позволяют определить 2 условия, касающиеся 2 различных записей, связанных ссылкой Training.

Если бы мы написали:

```
SELECT Self FROM amEmplDept WHERE (Trainings.Title = 'HP Software') AND (Trainings.Title = 'Database')
```

Были бы выбраны все сотрудники, проходящие курсы обучения с обоими названиями.

Если бы мы написали:

```
SELECT Self FROM amEmplDept WHERE (Trainings.Title = 'HP Software') OR (Trainings.Title = 'Database')
```

Были бы выбраны все сотрудники, проходящие один курс обучения с одним названием из двух.

Чтобы сократить запрос, можно использовать символ двоеточия:

```
SELECT amPortfolio.self, amModel_FullName:self FROM amPortfolio portfolio
```

Такой запрос эквивалентен запросу:

```
SELECT amPortfolio.self, FullName.semself FROM amPortfolio portfolio, amModel FullName
```

Запрос с переменной

Синтаксис 1

Например: все модели, имя которых равно значению, указанному пользователем при выборе фильтра, использующего данный запрос.

Начальная таблица: **Модели** (amModel).

```
Имя = $Nom
```

Пользователь фильтра запроса должен заполнить поле **Имя** перед выполнением запроса.

Если пользователь вводит Тест, генерируемый запрос будет:

```
SELECT M1.Name FROM amModel M1 WHERE M1.Name = 'Test'
```

Синтаксис 2

Например: все модели, имя которых содержит значение, указанное пользователем при выборе фильтра, использующего данный запрос.

Начальная таблица: **Модели** (amModel).

```
Name LIKE amDbVal(Имя, 'Имя содержит', 'Указать значение, которое должно содержаться в поле')
```

Пользователь фильтра запроса должен заполнить поле **Имя содержит** перед выполнением запроса.

Поле **Имя содержит** отображает по умолчанию Указать значение, которое должно содержаться в поле.

Если пользователь вводит Тест, генерируемый запрос будет:

```
SELECT M1.Name FROM amModel M1 WHERE M1.Name LIKE 'Test' ESCAPE '\'
```

4 Crystal Reports

В данной главе объясняется, как отображать и печатать отчеты в решении Asset Manager при помощи Crystal Reports.

Ограничения

Использование отчетов Crystal Reports зависит от того, получаете ли вы к ним доступ через веб-клиент или Windows-клиент, а также от того, используете ли вы отчеты, сохраненные в базе данных Asset Manager или базе данных Crystal Reports Server.

► Ограничения описаны в руководстве Интерфейс пользователя, глава Различия между Windows-клиентом и веб-клиентом, раздел Составление отчетов.

Общий обзор

Цель использования Crystal Reports

Crystal Reports используется для создания и модификации отчетов (файлы .rpt).

Файлы .rpt, созданные при помощи Crystal Reports, можно просматривать и распечатывать в веб-клиенте или Windows-клиенте Asset Manager.

Отчеты Crystal Reports нельзя создавать или модифицировать при помощи Asset Manager.

Место хранения отчетов

Файлы .rpt можно хранить разными способами:

- В базе данных Asset Manager.

К отчетам, сохраненным таким образом, можно получить доступ только через Windows-клиент.

- В базе данных Crystal Reports Server.

К отчетам, сохраненным таким образом, можно получить доступ через Windows-клиент или веб-клиент.

Когда пользователь запрашивает отчет Crystal Reports (для просмотра или печати) через веб- или Windows-клиент Asset Manager, запрос отправляется на Crystal Reports Server.

Данные извлекаются сервером Crystal Reports Server из базы данных Asset Manager, и отчет отображается в веб-клиенте или в Internet Explorer (клиент Windows).

Место ссылок на отчеты

На каждый файл .rpt ссылается запись в таблице **Отчеты** (amReport), независимо от способа его хранения:

- Если отчет сохранен в базе данных Asset Manager: файл .rpt импортируется в базу данных через Windows-клиент (кнопка **Импорт** на странице сведений об отчетах).
- Если отчет сохранен в базе данных Crystal Reports Server: на файл .rpt ссылается поле **Файл** (FileName), расположенное на странице сведений об отчетах.

Типы отчетов

Существует несколько типов отчетов:

- Отчеты со сведениями

Эти отчеты используются для отображения информации о записи в определенной таблице.

Доступ к этим отчетам можно получить, если выбрать запись из контекстной таблицы отчета.

Эти отчеты доступны только для Windows-клиентов.

- Графики и отчеты со списками

Эти отчеты используются для отображения информации о выборе записей в определенной таблице.

Доступ к этим отчетам можно получить всегда.

Эти отчеты доступны для Windows-клиентов и веб-клиентов.

Установка и настройка для активации доступа Windows-клиентов Asset Manager к отчетам, сохраненным в базе данных Asset Manager

Необходимо установить одно из следующих приложений на каждой рабочей станции Windows-клиента:

- полная версия Crystal Reports.

Для определения поддерживаемых версий Crystal Reports: ► матрица поддержки Asset Manager

- Версия OEM (Original Equipment Manufacturer - производитель оригинального оборудования) Crystal Reports (Designer) для Asset Manager, предоставленная на диске Сервисные программы Asset Manager.
- Ограниченная версия (время прогона), предоставленная с Asset Manager

Для установки данной версии необходимо выбрать соответствующую опцию при установке Asset Manager.

Данная ограниченная версия может использоваться для предварительного просмотра и печати существующих отчетов, содержащих данные текущей базы данных Asset Manager при помощи Windows-клиента, но не веб-клиента. Нельзя создавать новые отчеты или модифицировать существующие отчеты.



ПОДСКАЗКА:

Если вы уже установили или планируете установить полную версию или версию OEM Crystal Reports, в Asset Manager не нужно устанавливать ограниченную версию Crystal Reports.

Установка и настройка для активации доступа веб-клиентов Asset Manager к отчетам, сохраненным в базе данных Crystal Reports Server

Предпосылки

Создание базы данных Asset Manager

Необходима база данных Asset Manager, созданная с помощью СУБД, выбранной на ваше усмотрение.

Эта база данных должна быть активирована с помощью действительных лицензионных ключей HP AutoPass. Она также должна быть доступна с сервера Crystal Reports Server.

Руководство ► Asset Manager Администрирование, глава Установка лицензионных ключей.

Установка необходимых компонентов

Установите следующие компоненты:

- Следующие модули Asset Manager:
 - Windows-клиент Asset Manager
 - Драйвер ODBC Asset Manager

Руководство ► Asset Manager Установка и обновление и руководство Asset Manager Администрирование, глава Доступ к базе данных через ODBC.

- Клиентский уровень СУБД, используемый для базы данных Asset Manager
 - Руководство Администрирование Asset Manager, глава Создание, изменение и удаление базы данных Asset Manager, раздел Создание пустой оболочки с помощью СУБД.

Создание подключения к базе данных Asset Manager

- 1 Запустите Windows-клиент Asset Manager.
- 2 Создайте подключение к базе данных Asset Manager (меню **Файл/Управление подключениями...**).
 - Руководство Администрирование Asset Manager, глава Создание, изменение и удаление базы данных Asset Manager, раздел Создание подключения при помощи Asset Manager.Необходимо запомнить имя данного подключения. Оно понадобится позже.
- 3 Протестируйте подключение (кнопка **Тест** в окне управления подключениями).

Импорт Портфеля - Бизнес-данных в вашу базу данных

Портфель - Бизнес-данные должны быть импортированы для обеспечения надлежащей работы интеграции.

- ▶ Руководство Портфель, глава ИТ-портфель, раздел Компьютеры/ Просмотр компьютеров в базе данных HP Discovery and Dependency Mapping Inventory из Asset Manager/ Предварительные задачи/ Импорт Портфеля - Бизнес-данных в вашу базу данных.

Установка Crystal Reports Server и Crystal Reports (Designer)

Crystal Reports Server и Crystal Reports (Designer) можно установить несколькими способами:

- При помощи версии OEM, предоставляемой на диске Сервисные программы Asset Manager.

Установку можно выполнить вручную или автоматически.

- При помощи полной коммерческой версии, которая уже имеется.
 - ▶ Документация Crystal Reports.

Для определения поддерживаемых версий Crystal Reports: ▶ матрица поддержки Asset Manager



ЗАМЕЧАНИЕ:

В Crystal Reports Server не нужно выполнять никаких специальных настроек.

Настройки будут устанавливаться для каждого из отчетов.



ЗАМЕЧАНИЕ:

С Asset Manager может использоваться только Crystal Reports, установленная на базе ОС Windows. Это связано с тем, что в Crystal Reports используется драйвер ODBC Asset Manager.

В этом разделе описывается установка версии OEM Crystal Reports для Asset Manager и Crystal Reports Server для Asset Manager из предоставленных Сервисных программ Asset Manager.



ВАЖНО:

Продукты бизнес-объектов (Crystal Reports для Asset Manager и Crystal Reports Server для Asset Manager), предоставленные с Asset Manager, можно использовать только с продуктами ВТО (Business Technology Optimization - оптимизация бизнес-технологий).

Бизнес-объекты для предоставленной лицензии Crystal Reports Server имеют 20 клиентских лицензий. Только 20 пользователям разрешено использовать Crystal Reports Server одновременно.

Установка Crystal Reports для Asset Manager и Crystal Reports Server для Asset Manager (автоматический режим)

Установка в автоматическом режиме выполняется с использованием предварительно определенных опций.

- 1 Запустите скрипт AssetManagerReportingInstall.bat, расположенный в корневом каталоге предоставленного диска Сервисные программы Asset Manager.
- 2 Имеются два варианта выбора:
 - Установить Crystal Reports Server
 - Установить Crystal Reports DesignerВыберите нужную опцию.
- 3 Возможен выбор между следующими вариантами:
 - Установка в автоматическом режиме
 - Ручная установкаВыберите установку в автоматическом режиме.
- 4 Установка будет выполнена с использованием предварительно определенных опций. Процесс установки будет приостанавливаться несколько раз; его необходимо будет возобновлять вручную.

Выполните эти этапы для установки обоих продуктов (Crystal Reports для Asset Manager и Crystal Reports Server для Asset Manager).

Установка Crystal Reports Server для Asset Manager и Crystal Reports для Asset Manager вручную

Если необходимо настроить установку Crystal Reports Server для Asset Manager и/или Crystal Reports для Asset Manager, нужно выполнить установку вручную.

Подробную информацию можно найти в руководствах по бизнес-объектам, которые можно найти на сайте: <http://support.businessobjects.com/documentation>.

Если вы не можете получить доступ к серверу документации, изучите примечания к версии и руководство по установке Crystal Reports Server на установочном диске Asset Manager на английском языке в папке Doc.

Для выполнения ручной установки Crystal Reports Server для Asset Manager или Crystal Reports для Asset Manager выполните следующие действия:

- 1 Запустите скрипт AssetManagerReportingInstall.bat, расположенный в корневом каталоге предоставленного диска Сервисные программы Asset Manager.
- 2 Имеются два варианта выбора:
 - Установить Crystal Reports Server

- Установить Crystal Reports Designer
Выберите нужную опцию.
- 3 Возможен выбор между следующими вариантами:
 - Установка в автоматическом режиме
 - Ручная установка
Выберите ручную установку.
- 4 Далее выполните процедуру, указанную мастером установки.
Выполните эти этапы для установки обоих продуктов (Crystal Reports для Asset Manager и Crystal Reports Server для Asset Manager).
Ручная установка Crystal Reports Server для Asset Manager требует выполнения дополнительных процедур настройки вручную. ► [Настройка \(ручная установка версии OEM или установка полной коммерческой версии Crystal Reports Server\)](#) [стр. 77]

Настройка (ручная установка версии OEM или установка полной коммерческой версии Crystal Reports Server)

Если вы выполнили установку предоставленной версии OEM Crystal Reports Server для Asset Manager в автоматическом режиме, не нужно выполнять процедуру настройки, описанную в этом разделе (информация о настройке была введена автоматически во время установки).

Однако, если вы выполнили ручную установку предоставленной версии OEM Crystal Reports Server для Asset Manager, или установили полную коммерческую версию Crystal Reports Server, для интеграции необходим дополнительный этап настройки.

Файл u2lamlib.dll

- Если Asset Manager и Crystal Reports Server были установлены на одном и том же сервере, для файла u2lamlib.dll настройки не требуется.
- Если Asset Manager и Crystal Reports Server не установлены на одном и том же сервере:
 - 1 Откройте проводник Windows.
 - 2 Отобразите содержимое папки lib\XX (где XX соответствует языку установки Asset Manager) на диске Сервисные программы Asset Manager.
 - 3 Скопируйте файл u2lamlib.dll.
 - 4 Вставьте данный файл в папку <Полный путь к папке установки Crystal Reports Server>\Common\3.5\bin. Например, C:\Program Files\Business Objects\Common\3.5\bin.

Импорт отчетов Asset Manager в Crystal Reports Server

Отчеты Asset Manager можно импортировать в Crystal Reports Server несколькими способами.

Импорт файлов .rpt Asset Manager вручную

Для функционирования данного типа импорта Crystal Reports Server должен быть установлен и должен иметь доступ к серверу, на котором размещено решение Asset Manager (для получения доступа к отчетам).

В этом случае:

- 1 Запустите Мастер публикаций Crystal Report Server (меню Windows **Пуск/ Программы/ Бизнес-объекты/ Crystal Reports Server/ Мастер публикаций**).
- 2 Необходимо выполнить мастер, выбрав импортируемые отчеты и указав местоположение в Crystal Reports Server, куда их нужно поместить.

Импорт отчетов при помощи предоставленного файла BIAR

- Если вы выполнили установку Crystal Reports Server для Asset Manager в автоматическом режиме при помощи предоставленной версии OEM, отчеты Asset Manager автоматически импортируются посредством развертывания файла BIAR (Business Intelligence Archive Resource - ресурс архива бизнес-аналитики) в базе данных Crystal Reports Server с использованием скрипта deployReports.bat (находящегося в папке установки диска Сервисные программы Asset Manager).



ЗАМЕЧАНИЕ:

Этот скрипт работает, только если Crystal Reports Server был установлен с использованием заданных по умолчанию параметров Asset Manager.

- В противном случае, используйте мастер импорта Crystal Reports Server:
 - 1 Запустите Мастер импорта Crystal Report Server (меню Windows **Пуск/ Программы/ Бизнес-объекты/ Crystal Reports Server/ Мастер импорта**).
 - 2 На странице Среда источника выберите Файл ресурса архива бизнес-аналитики (BIAR) в качестве источника, а затем выберите его местоположение на диске Сервисные программы Asset Manager:
<Media>\lib\reports\am_reports_EN.biar.
 - 3 На странице Среда назначения введите параметры подключения к базе данных Crystal Reports Server.
 - 4 На странице Выбрать импортируемые объекты:

- Выберите опцию Импортировать пользователей и группы пользователей, но не выбирайте никаких дополнительных опций.
 - Выберите опцию Импортировать папки и объекты, но не выбирайте никаких дополнительных опций.
 - Выберите опцию Импортировать объекты репозитория.
 - Очистите все остальные опции.
- 5 Появится предупреждение о том, что Crystal Reports Server не поддерживает семантический срез данных. Пропиригнорируйте это предупреждение.
 - 6 На странице Поэтапный импорт выберите все опции.
 - 7 На странице Пользователи и группы выберите пользователя Asset Manager.
 - 8 На странице Папки и объекты выберите папку Asset Manager (при этом выбираются все подпапки).
 - 9 Выполните выход из мастера.
- Теперь все отчеты Asset Manager и пользователь Asset Manager были импортированы в базу данных Crystal Reports Server.

Создание пользователя Asset Manager в Crystal Reports Server

- Если вы импортировали отчеты с помощью предоставленного файла BIAR (Business Intelligence Archive Resource - ресурс архива бизнес-аналитики), пользователь Crystal Reports Server Asset Manager также был импортирован.
- В противном случае, вы можете:
 - Создать пользователя вручную ► [Ручное создание](#) [стр. 79]
 - Автоматически создать пользователя в процессе единой регистрации ► [Автоматическое создание с помощью процесса единой регистрации](#) [стр. 81]

Ручное создание

Создание пользователя Crystal Reports Server Asset Manager состоит из двух этапов:

- 1 [Создание пользователя](#) [стр. 79]
- 2 [Назначение прав](#) [стр. 80]

Создание пользователя

Для создания пользователя Crystal Reports Server Asset Manager

- 1 Запустите панель запуска администрирования Java в Crystal Reports Server (меню Windows **Пуск/ Программы/ Бизнес-объекты/ Crystal Reports Server/ Панель запуска администрирования Java**).

- 2 Щелкните ссылку Центральная панель управления и укажите себя.
 - 3 Щелкните Пользователи (в группе Организовать).
 - 4 Отобразится окно управления пользователями. Щелкните **Новый пользователь**.
 - 5 Отобразится окно создания нового пользователя. Введите следующие параметры:
 - Тип аутентификации: предприятие
 - Имя учетной записи: AssetManager
 - Полное имя: AssetManager
 - Пароль: выберите любой пароль
 - Выберите Пароль с неограниченным сроком действия.
 - Очистите все остальные опции настройки пароля.
 - Выберите тип подключения Одновременно работающий пользователь.
- Щелкните **ОК**. Теперь был создан менеджер AssetManager.

Назначение прав

Отчеты Asset Manager уже должны были импортироваться в Crystal Reports Server.

Чтобы назначить все права для отчетов Asset Manager пользователю AssetManager:

- 1 Запустите модуль Центральная панель управления Crystal Reports Server (меню Windows **Пуск/ Программы/ Бизнес-объекты/ Crystal Reports Server/ Панель запуска администрирования Java**, затем - ссылку Центральная панель управления).
- 2 Щелкните Папки (в группе Организовать).
- 3 Щелкните папку, из которой вы экспортировали отчеты (например, Asset Manager).
- 4 Во вкладке Права предоставляются права для Asset Manager.
- 5 Щелкните **Добавить/Удалить....**
- 6 В поле Выбрать операцию выберите Добавить пользователей. Затем в поле Доступные пользователи выберите пользователя AssetManager и щелкните **>**.
Пользователь AssetManager теперь связан с папкой Asset Manager.
Щелкните **ОК**.
- 7 Для пользователя AssetManager в раскрывающемся списке, который находится в столбце Уровень доступа, выберите Полный контроль.
- 8 Для сохранения сделанных изменений нажмите **Обновить**.

Теперь у пользователя AssetManager есть возможность полного контроля над отчетами Asset Manager.

Автоматическое создание с помощью процесса единой регистрации

Если пользователь Crystal Reports Server Asset Manager не был создан, а единая регистрация была проведена (► [Настройка единой регистрации](#) [стр. 84]) при попытке пользователя подключиться к отчету, механизм единой регистрации автоматически создаст пользователя и добавит его/ее в соответствующую группу. Эти параметры определены в файле reporting.properties по умолчанию:

- Имя пользователя: AssetManager
- Группа: Администратор

Это также соответствует конфигурации, импортированной файлом BIAR.

Эти параметры можно изменить, отредактировав файл reporting.properties и изменив соответствующую информацию в базе данных Crystal Reports Server.

Обновление исходной базы данных отчетов Asset Manager

Исходная база данных отчетов Asset Manager соответствует базе данных, в которой хранятся данные отчетов.

Исходные данные отчетов можно обновлять несколькими способами.

Модификация исходной базы данных при помощи файла .rpt.

- 1 Запустите Crystal Reports (Designer).
- 2 Откройте файл .rpt для модификации.
- 3 В меню База данных выберите Установить местоположение источника данных.
- 4 Выберите Раскрыть отчеты и Базы данных Asset Manager.
- 5 Отредактируйте поле Код пользователя, указав новые данные: <Код пользователя>@<База>, например, Admin@AMDemo51en.
- 6 Появится сообщение, запрашивающее пароль.
Введите пароль (если пользователь - Admin, пароль является пустым по умолчанию).
Далее щелкните Открыть.
- 7 Появится новое сообщение, на этот раз из ODBC (RDO), запрашивающее пароль для базы данных.
Введите пароль (если пользователь - Admin, пароль является пустым по умолчанию).
Далее щелкните Завершить.
- 8 Если существуют исходные базы данных для дополнительных отчетов, выполните аналогичную операцию для каждой из них.

9 Закройте окно.

Ручная модификация исходной базы данных при помощи модуля Центральная панель управления Crystal Reports Server

Запустите модуль Центральная панель управления Crystal Reports Server (меню Windows **Пуск/ Программы/ Бизнес-объекты/ Crystal Reports Server/ Панель запуска администрирования Java**, затем - ссылку Центральная панель управления).

Поочередно откройте каждый отчет Asset Manager и для каждого отчета:

- 1 Выберите опцию Использовать указанную здесь пользовательскую информацию для входа в базу данных и драйвер ODBC.
- 2 В поле Сервер введите Базы данных Asset Manager.
- 3 В поле База данных введите имя базы данных (например, AMDemo51en).
- 4 Выберите опцию Предлагать пользователя для входа в базу данных (эта опция требуется для механизма единой регистрации).

Автоматическая модификация исходной базы данных

Отчеты Asset Manager уже должны были импортироваться в Crystal Reports Server.

Чтобы автоматически модифицировать исходную базу данных:

- 1 Вставьте диск Сервисные программы Asset Manager в привод CD-ROM.
- 2 На диске выберите папку установки и откройте командную строку DOS (щелкните правой кнопкой мыши и выберите Открыть командное окно здесь).
- 3 Если Crystal Reports Server не был установлен в заданной по умолчанию папке (C:\Program Files\Business Objects), определите папку установки Crystal Reports Server посредством ввода следующей команды:

```
set BOBJ_InstallDir=<Папка установки Crystal Reports Server>
```

Например:

```
set BOBJ_InstallDir=C:\my directory\Business Objects
```

- 4 Если вы импортировали отчеты в папку, имя которой - не Asset Manager, определите папку импортирования отчетов посредством ввода следующей команды:

```
set AMRootReportsFolder=<папка импортирования отчетов>
```

Например:

```
set AMRootReportsFolder=Asset Manager Reports
```

- 5 Вы собираетесь выполнить скрипт buildReportsList.bat для компилирования списка отчетов Asset Manager, загруженных в Crystal Reports Server. По

умолчанию результат этого скрипта хранится в файле C:\am_reports.dat. Если необходимо изменить местоположение или имя файла результатов, введите следующую команду:

```
set ReportsListFile=<путь и имя файла>
```

Например:

```
set ReportsListFile=C:\temp\asset_manager_reports.dat
```

- 6 Затем выполните скрипт buildReportsList.bat посредством ввода следующей команды:

```
buildReportsList.bat
```

buildReportsList.bat генерирует файл и предоставляет имя файла по окончании выполнения.

- 7 Чтобы исключить отчеты из импортирования в базу данных Asset Manager, откройте файл, сгенерированный buildReportsList.bat, и вставьте символ # в начало каждой строки отчета, который необходимо исключить.
- 8 В командной строке DOS введите следующую команду для определения исходного языка данных:

```
set BOBJ_ClientLanguage=<Язык базы данных Asset Manager>
```

Выберите язык, используя один из следующих двухбуквенных языковых кодов: EN, FR, DE, ES, JP, CHS, CHT, KO, NL, SV, IT, PT.

- 9 Откройте проводник Windows.
- 10 Перейдите в папку \lib\reports.
- 11 Скопируйте файл am_reportsDSDef_xxx.dat, где xxx представляет язык установки Crystal Reports.
- 12 Вставьте его в любую папку.
- 13 Отредактируйте файл am_reportsDSDef_xxx.dat.
- 14 Введите параметр после базы данных, указав имя подключения к базе данных Asset Manager ([► Создание подключения к базе данных Asset Manager](#) [стр. 74]).
Не модифицируйте другие параметры.
- 15 Вернитесь к окну командной строки DOS.
Обозначьте ссылкой файл am_reportsDSDef_xxx.dat, который вы только что модифицировали, посредством ввода следующей команды:

```
set DataSourceDefFile=<Полное имя модифицированного файла am_reportsDSDef_xxx.dat>
```

- 16 Введите следующую команду для выполнения скрипта changeReportsDS.bat:

```
changeReportsDS.bat
```

- 17 Теперь вы связали пользовательскую исходную базу данных с каждым отчетом Asset Manager в Crystal Reports Server.

Настройка единой регистрации

Во избежание необходимости нескольких аутентификаций (при подключении к веб-клиенту Asset Manager во время доступа к отчету, или в случае необходимости получения данных из базы данных Asset Manager для отчета) необходимо настроить единую регистрацию.

Единая регистрация позволяет пользователям получить доступ ко всем службам и не требует, чтобы пользователи несколько раз вводили информацию, использующуюся для аутентификации.

Если вы установили Crystal Reports Server для Asset Manager автоматически, единая регистрация уже была автоматически настроена.

Однако если вы выполнили установку вручную, необходимо вручную настроить единую регистрацию, выполнив следующее:

- 1 Скопируйте файлы am-reporting.common-51.jar и am-reporting.serverside-51.jar (которые можно найти в папке lib\ssso на диске Сервисные программы Asset Manager) в папку lib веб-приложения Crystal Reports Server для Asset Manager. Например: C:\Program Files\Business Objects\Tomcat\webapps\businessobjects\enterprise115\desktoplaunch\WEB-INF\lib
- 2 Откройте файл веб-приложения Crystal Reports web.xml, находящийся в: C:\Program Files\Business Objects\Tomcat\webapps\businessobjects\enterprise115\desktoplaunch\WEB-INF. Скопируйте содержимое файла reporting-SsoWeb.xml (находящегося в папке lib\ssso на диске Сервисные программы Asset Manager) в этот файл.
- 3 Скопируйте файл reporting.properties (находящийся в папке lib\ssso на диске Сервисные программы Asset Manager) в папку WEB-INF/classes веб-приложения Crystal Reports.

Пример.

```
C:\Program Files\Business Objects\Tomcat\webapps\businessobjects\enterprise115\desktoplaunch\WEB-INF\classes
```

Вы можете отредактировать этот файл и настроить его параметры, включая:

Имя параметра	Значение по умолчанию	Описание параметра
server.bo.shared_secret	cradmin	Общий секрет - пароль для единой регистрации.
server.bo.admin_account	Administrator	Учетная запись администратора Crystal Reports (используется только для автоматического создания пользователя).

Имя параметра	Значение по умолчанию	Описание параметра
server.bo.cms_port	6400	Номер порта базы данных Crystal Reports Server.
server.bo.default_user	AssetManager	Если этот параметр определен, подключение к Crystal Reports Server будет выполнено с использованием этого имени пользователя, а не имени пользователя, предоставленного механизмом единой регистрации. Если имя пользователя не существует, оно автоматически создается, если был активирован параметр on_the_fly_creation. В противном случае, не удастся осуществить попытку подключения.
serverbo.user.on_the_fly_creation	true	Авторизовать или отклонить автоматическое создание пользователя в Crystal Reports Server (пользователь будет создан или не будет создан в базе данных Crystal Reports Server во время своего первого подключения, если пользователь еще не существует в базе данных).
serverbo.user.can_change_password	true	Свойства Crystal Reports для пользователя, созданного автоматически.
serverbo.user.must_change_password	false	Idem
serverbo.user.password_expires	false	Idem
server.bo.user.is_named	false	Idem
server.bo.user.password	Welcome	Пароль пользователя по умолчанию для пользователя, созданного автоматически.
server.bo.user.description	user auto-created by the Hp Ers Sandbox	Описание пользователя по умолчанию для пользователя, созданного автоматически.

- 4 Настройте пароль для единой регистрации:
 - a Откройте центральную панель управления Crystal Reports: меню Windows Пуск/ Программы/ Бизнес-объекты/ Crystal Reports Server/ Панель запуска администрирования Java.
 - b Щелкните Центральная панель управления.
 - c Имя для входа по умолчанию - Администратор; пароль не требуется.
 - d В разделе Управлять щелкните Аутентификация.
 - e Во вкладке Предприятие выберите опцию Активирована доверенная аутентификация и заполните поле Общий секрет, используя пароль, аналогичный применяемому в файле reporting.properties и тому, который вы копировали ранее, в переменной server.bo.shared_secret. По умолчанию - cradmin.
 - f Щелкните **Обновить**.



ВНИМАНИЕ:

Механизм единой регистрации не сработает, если хост, используемый для приложения Asset Manager, не использует ту же область DNS, что и хост для Crystal Reports Server. Это происходит вследствие определенных ограничений, установленных Интернет-стандартами, такими как заполнение файлов cookie.

Если оба хоста находятся в двух разных областях DNS, пользователи должны будут выполнять аутентификацию при каждом подключении: при подключении к Asset Manager, при подключении к Crystal Reports Server и при подключении к базе данных Asset Manager.

Кроме того, для корректного функционирования единой регистрации необходимо ввести имя области в:

- Имя сервера Crystal Reports Server: [Объявление Crystal Reports Server](#) [стр. 87]
- Имя сервера Asset Manager Web Tier: http://<Имя сервера HP Asset Manager Web Tier с именем области>:<Порт, используемый HP Asset Manager Web Tier>/AssetManager

Настройка вычисляемого поля sysCoreWebCrystal

Чтобы модифицировать вычисляемое поле sysCoreWebCrystal:

- 1 Запустите Windows-клиент Asset Manager.
- 2 Откройте экран вычисляемые поля (меню **Администрирование/Вычисляемые поля...**).
- 3 Выберите вычисляемое поле Отобразить отчет (sysCoreWebCrystal).

Вы можете модифицировать скрипт, используемый для этого вычисляемого поля, если вы находитесь в одной из описанных ниже ситуаций.

Если вы используете предыдущую версию Crystal Reports Server

Если вы используете версию, предшествующую версии, предоставленной с Asset Manager, модифицируйте скрипт, используемый для вычисляемого поля sysCoreWebCrystal, следующим образом:

```
bHasOldCrystal = TRUE  
bHasSSO = FALSE
```

Механизм единой регистрации (SSO) не доступен для предыдущих версий Crystal Reports Server.

Если вы не хотите использовать единую регистрацию

Если по какой-либо причине вы не хотите использовать механизм единой регистрации (SSO), вы можете отключить его, модифицировав вычисляемое поле sysCoreWebCrystal следующим образом:

```
bHasOldCrystal = FALSE  
bHasSSO = FALSE
```

Если приложение Asset Manager напрямую сообщается с Crystal Reports Server и пользователю нужно выполнить аутентификацию при подключении к Crystal Reports Server и базе данных Asset Manager.

Объявление Crystal Reports Server

- 1 Запустите Windows- или веб-клиент Asset Manager.
- 2 Подключитесь к базе данных.
- 3 Запустите мастер Изменение URL-адресов серверов приложений... (BstBackEndOpt) (ссылка на навигационной панели **Администрирование/ Изменение URL-адресов серверов приложений...**).
- 4 Введите сведения в мастер следующим образом:

Поле	Значение
Страница Добавить и/или изменить свойства серверов приложений	
Имя добавляемого сервера приложений	Если в нижней части страницы нет строки, столбец Приложение которой равняется Crystal, заполните данное поле значением Crystal, затем щелкните Добавление сервера приложения .
Список приложений, строка, чей столбец Приложение равняется Crystal, столбец Адрес URL сервера	http или https://<Crystal Reports Server имя сервера>:<порт, используемый Crystal Reports Server> Пример: http://BusinessObjectServer:8088 ВНИМАНИЕ: Для корректного функционирования единой регистрации необходимо вести имя области в имя сервера Crystal Reports Server.
Страница Сводка изменений	Проверьте содержимое страницы.
Щелкните Завершить .	

Поле	Значение
Windows-клиент: нажмите ОК	

- 5 Windows-клиент: повторно подключитесь к базе данных (Меню **Файл/Подключиться к базе данных**).

Отображение отчета

Чтобы отобразить отчет

Способ отображения отчета зависит от типа отчета (сведения, список или график):

Отчеты со сведениями



ВНИМАНИЕ:

Отчеты со сведениями доступны через Windows-клиент, а не через веб-клиент.

Отчет со сведениями может отображаться разными способами:

Отображение отчета посредством его выбора на навигационной панели

- 1 Отобразите список, содержащий запись, которая является объектом отчета.
- 2 Выберите запись, которая является объектом отчета.



ВАЖНО:

Можно выбрать только одну запись.

- 3 Разверните ссылку навигационной панели, которая соответствует функциональной области отчета.



ПОДСКАЗКА:

Функциональная область отчета определена в поле **Область (Domain)** в сведениях об отчете.

- 4 Щелкните имя отчета в навигационной панели.

Отображение отчета из списка отчетов (только для Windows-клиента)

- 1 Отобразите отчеты (ссылка **Администрирование/Составление отчетов/Отчеты** на навигационной панели).
- 2 Отобразите сведения об отчете.
- 3 Нажмите кнопку **Предварительный просмотр**.
- 4 Выберите запись, которая является объектом отчета.
- 5 Щелкните **ОК**.

Отображение отчета при помощи меню **Файл/ Печать** (только для Windows-клиента)

- 1 Отобразите список записей, которые являются объектом отчета.
- 2 Выберите записи, которые являются объектом отчета.
- 3 Выберите пункт меню **Файл/Печать**.
- 4 Занесите в поле **Тип** значение Отчет со сведениями (Crystal Reports).
- 5 Выберите отчет в списке **Отчет**.
- 6 Щелкните **Предварительный просмотр**.

Графики и отчеты со списками

Отчет в виде списка или графика может отображаться разными способами:

Отображение отчета посредством его выбора на навигационной панели

- 1 Разверните ссылку навигационной панели, которая соответствует функциональной области отчета.



ПОДСКАЗКА:

Функциональная область отчета определена в поле **Область (Domain) в сведениях об отчете.**

- 2 Щелкните имя отчета в навигационной панели.

Отображение отчета из списка отчетов (только для Windows-клиента)

- 1 Отобразите отчеты (ссылка **Администрирование/Составление отчетов/Отчеты** на навигационной панели).
- 2 Отобразите сведения об отчете.
- 3 Щелкните **Предварительный просмотр**.

Отображение отчета при помощи меню **Файл/ Печать** (только для Windows-клиента)

- 1 Выберите пункт меню **Файл/Печать**.
- 2 Занесите в поле **Тип** значение Неконтекстный отчет (Crystal Reports).
- 3 Выберите отчет в списке **Отчет**.
- 4 Щелкните **Предварительный просмотр**.

Отображение отчета при помощи автоматического обновления (только в Windows-клиенте)

- 1 Выберите меню **Инструменты/ Составление отчетов/ Статистика Crystal Reports**
- 2 Заполните поле **Вид**, если вы хотите отфильтровать список отчетов (список справа от поля **Вид**).
- 3 Выберите отчет в списке справа от поля **Вид**.
- 4 Для обновления отчета щелкните значок  .
Для установления частоты автоматического обновления отчета щелкните правой кнопкой мыши значок  .
Для модификации коэффициента масштабирования (3 уровня) щелкните значок  .

Доступные отчеты

Ниже представлены условия, которым должен соответствовать отчет, прежде чем его можно будет использовать в Windows-клиенте или веб-клиенте.

Условие	Клиент Windows	Веб-клиент
База данных, в которой сохранен отчет	■ База данных Asset Manager ■ База данных Crystal Reports Server	База данных Crystal Reports Server
ВНИМАНИЕ: Для веб-клиента и клиента Windows отчет доступен через Crystal Reports Server, если Crystal Reports Server был объявлен с помощью мастера Изменение адресов URL серверов приложений... (BstBackEndOpt), и если поле отчета Файл (база данных) содержит численное значение.		

Условие	Клиент Windows	Веб-клиент
Тип отчета	■ Отчет со сведениями ВНИМАНИЕ: Чтобы отчет появился на навигационной панели, необходимо выбрать допустимый контекст. ■ Отчет в виде списка ■ График	■ Отчет в виде списка ■ График
Видно в меню (bVisible) - флажок	Установлен	Установлен

Как следует отображать отчет

Клиент Windows:

- Если файл .rpt хранится в базе данных Asset Manager, отчет отображается клиентом Windows.
- Если файл .rpt хранится не в базе данных Asset Manager, а на сервере Crystal Reports, отчет отображается в Интернет-браузере.

Веб-клиент: отчет отображается веб-клиентом.

Как Asset Manager создает URL-адреса отчетов

Asset Manager использует вычисляемое поле Отображение отчета (sysCoreWebCrystal) для создания привязки HTML <A> с атрибутом HREF, метка которого - метка отчета. Она указывает на URL-адрес, для формирования которого сцеплением использовались:

- URL-адрес Crystal Reports Server, определенный мастером Изменение URL-адресов серверов приложений... (BstBackEndOpt)
- Текстовая строка, указанная в скрипте вычисляемого поля
- Поле отчета **Файл** (FileName)

При щелчке по ссылке, указанной после отчета, Asset Manager обрабатывает URL-адрес.



ЗАМЕЧАНИЕ:

По умолчанию вычисляемое поле **Отобразить отчет (sysCoreWebCrystal)** настраивается для получения доступа к **Business Object Crystal Report Server**, версия 11.

Если при использовании версии, предшествующей версии 11, сгенерированная ссылка не работает, необходимо вручную модифицировать вычисляемое поле:

- 1 Отобразите сведения о вычисляемом поле **Отобразить отчет (sysCoreWebCrystal)** (меню **Администрирование/ Вычисляемые поля...**).
- 2 Замените следующий URL-адрес в скрипте:

```
"/businessobjects/enterprise115/desktoplaunch/opendoc/openDocument.jsp?  
iDocID="
```

На:

```
"/businessobjects/viewers/crystalreportviewers115/htmlviewerbridge.jsp?id="
```

- 3 Сохраните сделанные изменения.

Обеспечение доступности отчетов

Чтобы отчет можно было использовать Windows- и/или веб-клиентами, необходимо создать файл .rpt отчета и сохранить отчет в одной из следующих баз данных:

- База данных Asset Manager
- База данных Crystal Reports Server

Чтобы определить, в какой базе данных нужно сохранить отчет Crystal Reports:

► [Место хранения отчетов](#) [стр. 72].

В этой главе содержится информация по следующим вопросам:

- [Получение файлов .rpt](#) [стр. 92]
- [Сохранение отчетов в базе данных Asset Manager](#) [стр. 93]
- [Сохранение отчетов в базе данных Crystal Reports Server](#) [стр. 95]

Получение файлов .rpt

Для Crystal Reports существуют несколько источников:

- отчеты, созданные пользователем с помощью Crystal Reports;
- отчеты, уже хранящиеся в базе данных Asset Manager.

Если вы хотите переместить их в базу данных Crystal Reports Server, необходимо поочередно экспортировать их при помощи Windows-клиента, нажимая кнопку **Экспорт** на странице со сведениями об отчете.

- Отчеты о бизнес-данных Asset Manager.

Данные отчеты находятся в подпапке datakit папки установки Asset Manager.



ЗАМЕЧАНИЕ:

Список доступных отчетов содержится в файле reports.txt в подпапке datakit\standard\reports, расположенной в папке установки Asset Manager.

Отчеты Crystal Reports хранятся в папке \datakit\standard\reports\rpt.

Не следует менять структуру файла Reports.txt, так как он используется скриптом импорта отчетов. С другой стороны, если необходимо только импортировать выбор отчетов в рабочую базу данных, можно удалять строки целиком из данного файла перед выполнением скрипта импорта. Можно также добавлять свои собственные отчеты.

Сохранение отчетов в базе данных Asset Manager

К отчетам, сохраненным таким образом, можно получить доступ только через Windows-клиент.

Поочередный импорт отчетов в существующую базу данных Asset Manager

Для вставки отчетов друг за другом:

- 1 Запустите Windows-клиент Asset Manager.
- 2 Подключитесь к базе данных.
- 3 Отобразите отчеты (ссылка **Администрирование/Составление отчетов/Отчеты** на навигационной панели).
- 4 Создайте новый отчет.
- 5 Нажмите кнопку **Импорт**.
- 6 Выберите необходимый файл с расширением .rpt в подпапке \datakit\standard\reports\rpt папки установки Asset Manager.

Импорт всех отчетов, предоставленных с Asset Manager, при создании новой базы данных

Чтобы импортировать отчеты при создании базы данных:

- 1 Запустите Asset Manager Application Designer.
- 2 Выберите пункт меню **Файл/Открыть**.

- 3 Выберите параметр **Открыть файл описания базы данных - создать новую базу данных**.
- 4 Выберите файл gbbase.xml, расположенный в подпапке config папки установки Asset Manager.
- 5 Запустите мастер создания баз данных (меню **Действие/Создать базу данных**).
- 6 Следуйте указаниям руководства Администрирование, раздел Создание, изменение и удаление базы данных Asset Manager.

На странице **Данные для импорта** выберите **Crystal Reports**.

Импорт всех отчетов, предоставленных с Asset Manager, в существующую базу данных

Чтобы импортировать отчеты в существующую базу данных:

- 1 Запустите Asset Manager Application Designer.
- 2 Выберите пункт меню **Файл/Открыть**.
- 3 Выберите параметр **Открыть файл описания базы данных - создать новую базу данных**.
- 4 Выберите файл gbbase.xml, расположенный в подпапке config папки установки Asset Manager.
- 5 Запустите мастер создания баз данных (меню **Действие/Создать базу данных**).
- 6 Заполните страницы мастера следующим образом (переходы между страницами мастера выполняются с помощью кнопок **Далее** и **Назад**):

Страница **Создать скрипт SQL/Создать базу данных**:

Поля	Значение
База данных	Выберите подключение к базе данных, в которую следует импортировать отчеты.
Создание	Импорт бизнес-данных.
Используйте расширенные параметры создания	Не следует использовать этот параметр

Страница **Параметры создания**:

Поля	Значение
Пароль	Введите пароль администратора: ЗАМЕЧАНИЕ: Администратор базы данных Asset Manager представляет собой запись в таблице Сотрудники и подразделения (amEmplDept), для которой поле Имя (Name) имеет значение Admin. Имя входа подключения к базе данных хранится в поле Имя пользователя (UserLogin). Именем администратора является Admin. Пароль хранится в поле Пароль (LoginPassword).

Страница **Данные для импорта:**

Поля	Значение
Доступные данные	Выберите параметр Crystal Reports .
Остановить импорт при возникновении ошибки	Выберите этот параметр для прекращения импорта в случае возникновения проблемы.
Файл журнала	Полное имя файла, в который будет записываться журнал всех операций импорта, включая ошибки и предупреждения.

- 7 Выполнить все варианты, определенные с помощью мастера (кнопка **Готово**).

Сохранение отчетов в базе данных Crystal Reports Server

К отчетам, сохраненным таким образом, можно получить доступ через Windows-клиент или веб-клиент.

Для каждого отчета, который необходимо сделать доступным для пользователей:

- 1 Запустите модуль центральной консоли управления Crystal Reports Server.
- 2 Импортируйте отчет.
- 3 Откройте отчет (меню **Объекты**).
- 4 Обратите внимание на следующие свойства:
 - Имя
 - Файл

Пример:

```
frs://Input/a_117/002/000/629/1f048f4996817a4.rpt
```

В данном примере необходимо обратить внимание на значение 629. Это идентификатор отчета, который необходимо сохранить в базе данных Asset Manager.



ПОДСКАЗКА:

Чтобы получить идентификаторы отчетов, можно также выполнить запрос, подобный приведенному ниже, с помощью конструктора запросов Crystal Reports Server:

```
SELECT SI_NAME, SI_ID FROM CI_INFOOBJECTS WHERE SI_PROGI  
D = 'CrystalEnterprise.Report'
```

Еще можно использовать файл, сгенерированный **buildReportsList.bat**: ► [Автоматическая модификация исходной базы данных](#) [стр. 82].

- 5 Заполните вкладку **Процессы**, подвкладку **База данных**.

Имя свойства	Значение
Сервер	Asset Manager Databases
База данных	Имя подключения Asset Manager
Пользователь	Оставьте это свойство пустым
Пароль	Оставьте это свойство пустым

- 6 Запустите Windows- или веб-клиент Asset Manager.
- 7 Подключитесь к базе данных.
- 8 Отобразите отчеты (ссылка **Администрирование/Составление отчетов/Отчеты** на навигационной панели).
- 9 Создайте отчет, заполнив следующие поля:

Имя поля или ссылки	SQL-имя поля или ссылки	Комментарий
Метка	Title	Имя отчета в базе данных Crystal Reports Server

ЗАМЕЧАНИЕ:

Это поле необязательно, но может оказаться полезным.

Имя поля или ссылки	SQL-имя поля или ссылки	Комментарий
Файл	FileName	Идентификатор отчета в базе данных Crystal Reports Server

Изменение отчета Seagate Crystal

Отчеты, сохраненные в базе данных Asset Manager

- 1 Отобразите отчеты (ссылка **Администрирование/Составление отчетов/Отчеты** на навигационной панели).
- 2 Отобразите сведения об отчете.
- 3 Экспортируйте отчет (кнопка **Экспорт**), чтобы создать файл .rpt.
- 4 Измените файл .rpt отчета с помощью Crystal Reports и сохраните его.
- 5 Снова отобразите сведения об отчете.
- 6 Импортируйте файл .rpt, который не был модифицирован (кнопка **Импорт**).
- 7 Сохраните изменения (кнопка **Изменить**).

Отчеты, сохраненные в базе данных Crystal Reports Server

Измените отчет непосредственно при помощи Crystal Reports, не меняя его код в базе данных Crystal Reports Server (► [Сохранение отчетов в базе данных Crystal Reports Server](#) [стр. 95]).

Создание отчета со сведениями

Отчет со сведениями - отчет, в котором печатается подробная информация об одной или нескольких записях, выбранных в списке.

Пример использования

- 1 Отобразите список активов.
- 2 Выберите актив.
- 3 Выберите пункт меню Файл/Печать.
- 4 Установите для поля Тип значение «Отчет со сведениями (Crystal Reports)».
- 5 Выберите отчет.
- 6 Напечатайте его.

Эта процедура распечатает отчет со сведениями для каждой выбранной записи.

Конфигурация отчетов в Crystal Reports

Чтобы получить отчет со сведениями, используйте приведенную ниже процедуру (пример выполнялся в Crystal Reports Professional 5.0):

- 1 Создайте поле формулы с помощью элемента меню Вставка/Поле формулы. Его имя должно удовлетворять следующему синтаксису:

<Имя SQL таблицы, для которой отчет является контекстным>Id



ЗАМЕЧАНИЕ:

Необходимо соблюдать регистр имен SQL таблиц.

Например, формула для создания контекстного отчета для таблицы активов:

amAssetId



ЗАМЕЧАНИЕ:

Не следует путать синтаксис имени поля формулы с именем SQL поля первичного ключа. Например, первичный ключ таблицы активов - lAstId, то есть отличается от amAssetId.

Формула CurrentUserId (с соблюдением регистра) позволяет идентифицировать пользователя, печатающего отчет. При печати данной формулой используется значение номера идентификатора (то есть значение поля с именем SQL lEmpIDeptId для текущего имени входа) пользователя, подключившегося к базе данных Asset Manager.

Если необходимо просмотреть результаты отчета для данной записи в контекстной таблице, следует изменить поле формулы, введя первичный ключ таблицы для существующей записи в базе данных Asset Manager.

Например:

512



ЗАМЕЧАНИЕ:

Поле формулы должно изменяться в окне, которое автоматически отображается при подтверждении имени нового поля формулы. Если поле формулы уже существует, щелкните кнопку **Изменить** для его изменения.

- 2 Для изменения формулы выбора используется элемент меню Отчет/Изменить формулу выбора/Запись. Используется следующий синтаксис:

```
{<Имя SQL таблицы контекста>.<Имя SQL поля - первичного ключа>} =  
@<Имя поля формулы>}
```

Регистр, используемый для имен SQL таблиц и полей, несущественен.
Пример:

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

С помощью описанной выше процедуры Asset Manager автоматически идентифицирует отчет как контекстуальный при его импорте в базу данных. Для его просмотра необходимо выполнить следующую процедуру:

- 1 откройте список отчетов с помощью элемента меню Инструменты/Составление отчетов/Отчеты.
- 2 Создайте новый отчет.
- 3 Импортируйте файл Crystal Reports (с расширением .rpt), щелкнув кнопку **Импорт**.
- 4 Сразу после добавления данного файла можно будет заметить, что в поле Таблица (имя SQL: TableName) показывается имя SQL контекстной таблицы. Если это не так, проверьте поле формулы и формулу выбора в Crystal Report.

Печать отчета

Способ печати отчета зависит от типа отчета (сведения, список или график):

Отчет со сведениями (только в Windows-клиенте)



ВНИМАНИЕ:

Отчеты со сведениями доступны через Windows-клиент, а не через веб-клиент.

Существуют разные способы печати отчета со сведениями:

Выберите отчет в навигационной панели

- 1 Отобразите список, содержащий запись, которая является объектом отчета.
- 2 Выберите запись, которая является объектом отчета.



ВАЖНО:

Можно выбрать только одну запись.

- 3 Разверните ссылку навигационной панели, которая соответствует функциональной области отчета.



ПОДСКАЗКА:

Функциональная область отчета определена в поле **Область (Domain)** в сведениях об отчете.

- 4 Щелкните имя отчета в навигационной панели.
- 5 Щелкните значок **Печать**.

Из списка отчетов

- 1 Отобразите отчеты (ссылка **Администрирование/Составление отчетов/Отчеты** на навигационной панели).
- 2 Отобразите сведения об отчете.
- 3 Щелкните **Предварительный просмотр**.
- 4 Выберите запись, которая является объектом отчета.
- 5 Щелкните **ОК**.
- 6 Щелкните значок **Печать**.

Через меню **Файл/ Печать**.

- 1 Отобразите список записей, которые являются объектом отчета.
- 2 Выберите записи, которые являются объектом отчета.
- 3 Выберите пункт меню **Файл/Печать**.
- 4 Занесите в поле **Тип** значение Отчет со сведениями (Crystal Reports).
- 5 Выберите отчет в списке **Отчет**.
- 6 Щелкните **Печать**.

Графики и отчеты со списками

Отчет в виде списка или графика можно распечатать разными способами:

Выберите отчет в навигационной панели

- 1 Разверните ссылку навигационной панели, которая соответствует функциональной области отчета.



ПОДСКАЗКА:

Функциональная область отчета определена в поле **Область** (Domain) в сведениях об отчете.

- 2 Щелкните имя отчета в навигационной панели.
- 3 Щелкните значок **Печать**.

Из списка отчетов (только для Windows-клиента)

- 1 Отобразите отчеты (ссылка **Администрирование/Составление отчетов/Отчеты** на навигационной панели).
- 2 Отобразите сведения об отчете.
- 3 Щелкните **Предварительный просмотр**.
- 4 Щелкните **ОК**.
- 5 Щелкните значок **Печать**.

Через меню **Файл/ Печать** (только для Windows-клиента)

- 1 Выберите пункт меню **Файл/Печать**.
- 2 Занесите в поле **Тип** значение Неконтекстный отчет (Crystal Reports).
- 3 Выберите отчет в списке **Отчет**.
- 4 Щелкните **Печать**.

Идентификация отчетов Crystal Reports, свойственных данному модулю

Чтобы идентифицировать отчеты Crystal Reports, свойственные данному модулю:

- 1 Запустите Windows-клиент Asset Manager.
- 2 Отобразите отчеты (ссылка **Администрирование/Составление отчетов/Отчеты** на навигационной панели).
- 3 Щелкните правой кнопкой мыши по списку в открывшемся окне.
- 4 Выберите пункт **Сервис/Настройка списка** из контекстного меню.
- 5 Добавьте ссылку **Область** (Domain) к столбцам списка.
- 6 Нажмите кнопку **ОК**.
- 7 Отсортируйте список по столбцу **Область**.
- 8 Отчеты для какого-либо модуля идентифицируются по имени области.

Привязка отчета к кнопке на экране (Windows-клиент) или к действиям на экране (веб-клиент)

Отчет можно связать с кнопкой на экране (Windows-клиент) или с действиями на экране (веб-клиент) несколькими способами.

Дополнительные сведения см.:

- Руководство Tailoring, часть 1 Customizing client computers, глава Customizing a client workstation, раздел Customizing buttons.
- Руководство Tailoring, глава Customizing the database, раздел Customizing existing objects/ Customizing objects / Customizing a detail/ Buttons.
- Руководство Tailoring, глава Customizing the database, раздел Creating new objects/ Creating action buttons.

5 Инфопанели

Asset Manager позволяет создавать инфопанели, которые включают отчеты и статистику. Инфопанелью обеспечивается общее представление информации для упрощения принятия решений. Подобно панели управления автомобиля или самолета, она отображает необходимые и критически важные данные, помогая предвидеть проблемы и добиться ровного функционирования компании. Таким образом, полезность инфопанели прямо связана с выбором отображаемых на ней индикаторов.

Создание инфопанели

Инфопанель определяется с помощью действия типа **Инфопанель**.

Для создания инфопанели:

- 1 Выберите ссылку **Администрирование/ Действия** на навигационной панели и щелкните **Создать** на экране сведений о действии.
- 2 Выберите имя инфопанели и выберите **Инфопанель** как **Тип** действия.
- 3 Если вы хотите, чтобы инфопанель появилась в древовидной структуре навигационной панели, определите **Область** для инфопанели.



ВНИМАНИЕ:

Если **Область** для инфопанели не указана, вы не сможете отобразить ее в веб-клиенте.

Можно также указать другие свойства, которые являются общими для всех действий.



ЗАМЕЧАНИЕ:

Дополнительную информацию можно найти в главе [Действия](#) [стр. 129] данного руководства.

- 4 На вкладке **Индикаторы** выберите индикаторы, которые необходимо добавить к инфопанели.

Информация о привязке индикатора хранится в таблице **Элемент инфопанели** (amDashboardItem).

В качестве индикаторов для инфопанелей в Asset Manager используются объекты двух типов: отчеты и статистика. В этом разделе не описывается создание статистики или отчетов; создание Crystal Reports подробно описывается в главе [Crystal Reports](#) [стр. 71] этого руководства, а статистика - в главе [Статистика](#) [стр. 107].

Для привязки индикатора:

- 1 Выберите в сведениях об инфопанели вкладку **Индикаторы** и щелкните кнопку .
- 2 Выберите **Имя**, чтобы указать индикатор в инфопанели.
- 3 Определите **Тип** индикатора (отчет или статистика).
- 4 Наконец, выберите **Отчет** или **Статистика**.



ЗАМЕЧАНИЕ:

Щелкните  для удаления индикатора из инфопанели. Отображается сообщение о том, что привязка индикатора к инфопанели будет удалена. Это означает, что объект, сохраненный в таблице **Элемент инфопанели** (amDashboardItem), будет удален.

- 5 Можно также установить флажок **Обновить по умолчанию**. Если установлен данный флажок, данные из инфопанели обновляются с периодичностью, определенной параметром

Статистика/Периодичность обновления статистики.



ЗАМЕЧАНИЕ:

Параметры приложений доступны в меню **Изменить/Параметры**.

Если данный флажок не установлен, инфопанель является статической и данные не обновляются динамически.

Пример инфопанели

В приведенном ниже примере описывается создание инфопанели, используемой отделом кадров компании. Как необходимые для инфопанели были определены следующие индикаторы:

- соотношение численности мужчин и женщин в компании;
- разбивка сотрудников по подразделению и месту размещения;
- разбивка сотрудников по времени в компании.

Эти индикаторы были созданы в разделе [Статистика, не заданная скриптом](#) [стр. 114] главы [Статистика](#) [стр. 107]. Для создания статистики, необходимой для данного примера, см. этот раздел.

Этап 1: создание инфопанели

- 1 Выберите меню **Инструменты/Действия/Изменить** и щелкните **Создать** для создания нового действия.
- 2 Заполните поля новой записи следующими данными:

Поле	Значение
Имя	Инфопанель ОК
Тип	Инфопанель

- 3 Проверьте созданное, нажав на кнопку **Создать**.

Этап 2: привязка индикаторов

Выберите вкладку **Индикаторы** в сведениях о вновь созданной инфопанели. Будут последовательно добавлены различные индикаторы статистики в соответствии со следующей процедурой:

- 1 Щелкните .
- 2 Заполните поле **Имя** индикатора. Можно использовать имена статистики, которую вы хотите добавить.
- 3 Выберите **Статистику** как **Тип** индикатора.
- 4 Щелкните на списке выбора  рядом с полем **Статистика**, а затем выберите статистику, которую необходимо добавить.
- 5 Щелкните **Добавить** (или кнопку **Добавить** с двойными стрелками, чтобы добавить несколько индикаторов).

Далее щелкните **Изменение**.

Этап 3: открытие инфопанели и переход к данным

Запустите инфопанель с помощью меню **Инструменты/ Действия**, или щелкнув ее имя в навигационной панели Windows-клиента или веб-клиента, в которой вы определили ее в поле **Область** (Domain).

Статистики инфопанелей можно настраивать сразу после их отображения с помощью параметров контекстного меню. Можно также щелкнуть на областях данных, чтобы отобразить сведения о примере статистики. Например, если щелкнуть по области 3-мерного сектора, представляющего собой численность мужчин, Asset Manager открывает окно подразделений и сотрудников и фильтрует данные в поле **Г-н/Г-жа** (MrMrs), используя **Г-н** как значение для фильтра.



ЗАМЕЧАНИЕ:

Для отчетов навигация недоступна.

6 Статистика

Статистика используется для отображения информации о базах данных в виде диаграмм.

Статистика не может отображаться обособленно; она должна отображаться через инфопанель или домашнюю страницу компании. ► [Просмотр статистики](#) [стр. 112]

Не следует путать статистику, описанную в этой главе, со статистикой Crystal Reports, описанной в главе [Отображение отчета при помощи автоматического обновления \(только в Windows-клиенте\)](#) [стр. 90]:

- Статистика Crystal Reports используется для отображения отчета Crystal Reports, который можно обновлять автоматически.
- При отображении диаграммы статистики можно изменять масштаб элементов диаграммы для того, чтобы отобразить записи, представленные определенным элементом. Изменение масштаба невозможно со статистикой Crystal Reports.

Создание статистики

Чтобы создать статистику:

- 1 Отобразите статистику (ссылка **Администрирование/ Статистика...** на навигационной панели).
- 2 Создайте новую статистику (кнопка **Создать**) и заполните различные поля.

- 3 Если вы хотите сохранить результаты статистики в базе данных, активируйте опцию **Сохранить результаты** (bStored).
 - [Оптимизация работы](#) [стр. 110].
- 4 Определить статистику можно одним из двух способов:
 - Используя стандартный запрос:
 - a Не устанавливайте флажок в ячейке **Скрипт** (bScript).
 - b Во вкладке **Параметры** введите разные параметры для создания статистики:
 - Необходимые компоненты:
 - Укажите таблицу с записями, которые необходимо анализировать в поле **Таблица** (TableName).
 - Для указания поля или ссылки, используемых для группировки элементов, используйте поле **Группировать по** (GroupBy).
 - Необязательные компоненты:
 - Если вы хотите использовать в таблице для создания статистики не все записи, в поле **Фильтр** укажите фильтр, который необходимо использовать для отображения записей, которые должны быть включены в статистику.

Если это поле не заполнено, учитываются все записи.
 - Если вы хотите осуществлять группировку по второму значению, после заполнения поля **Группировать по** можно использовать поле **Дополнительно группировать по** (SubGroupBy) для группировки по второму значению.
 - Если вы хотите использовать выражение для генерирования значений диаграмм, установите флажок **Использовать выражение** (bUseAgregExpr).

Затем заполните поле **Выражение** (AqlAgregExpr).
Это позволит вам выполнять операции (например, со средними показателями) или осуществлять разделение по партям.
 - Написав сложный скрипт для отображения статистики
 - 1 Установите флажок в ячейке **Скрипт** (bScript).
 - 2 Во вкладке **Скрипт** укажите скрипт, используемый для вычисления статистических данных.
 - Прочитайте параграф [Использование скрипта вместо запроса](#) [стр. 109], чтобы узнать о различиях между скриптами и запросами.
 - Прочитайте параграф [Статистика, заданная скриптом](#) [стр. 118], чтобы ознакомиться с примерами статистики, заданной скриптом.

ВНИМАНИЕ:

Если вы не указываете **Область** для статистики, она не будет доступна через веб-клиент (даже после привязки к инфопанели или домашней странице компании ► [Просмотр статистики](#) [стр. 112]).

ВНИМАНИЕ:

Веб-клиент: если выбран тип представления (поле **seType**) Кривые, статистика может отображаться, только если отображаемые значения являются числами. Поля Группировать по (**GroupBy**) и, возможно, Дополнительно группировать по (**SubGroupBy**) должны указывать на числовые значения.

- 5 После заполнения полей щелкните **Создать**.
- 6 Для отображения диаграммы щелкните **Тест**.
- 7 Щелкните **ESC**, чтобы закрыть диаграмму.

Использование скрипта вместо запроса

Как поясняется в параграфе [Создание статистики](#) [стр. 107], существует два способа описания статистики:

- При помощи простого запроса.
- При помощи сложного скрипта.

Использование скрипта вместо запроса

Стандартные запросы используются для отображения данных в таблицах:

- При помощи одного или двух критериев группировки,
- В полях баз данных и определенных вычисляемых полях,
- Возможно, при помощи фильтра,
- Если необходимо, применяется метод вычисления (выражение).

Если необходимо определить сложную статистику, вам потребуется определить ее при помощи скрипта.

Например, чтобы группировать записи по периоду, сравнивать записи в разных таблицах или группировать их при помощи данных из поля типа полное имя, необходимо написать скрипт.

Синтаксис скриптов

- ▶ Ознакомьтесь со следующими материалами:
 - примеры в разделе [Статистика, заданная скриптом](#) [стр. 118], чтобы узнать, какой синтаксис используется при написании скриптов.
 - [Определение настраиваемых цветов для статистики в виде диаграммы](#) [стр. 126]

Оптимизация работы

Причины проблем в работе

Статистика отображается посредством инфопанелей.

Когда отображается инфопанель, Asset Manager осуществляет доступ к соответствующим таблицам посредством одной или нескольких статистик.

Это может вызвать следующее:

- Замедление клиентской станции.
- Замедление получения доступа к определенным таблицам для других пользователей.
- Замедление отображения результатов в том виде, в каком они вычисляются в режиме реального времени. Кроме того, для определенной инфопанели может потребоваться вычисление нескольких статистик.

Способ устранения проблем в работе

Для сокращения проблем с производительностью можно регулярно автоматически генерировать статистику и сохранять результаты в таблице Заметки по статистике (amStatMemo).

В таком случае, если отображается инфопанель, Asset Manager проверяет эту таблицу, а не фактическую таблицу, из которой, в противном случае, данные собирались бы в режиме реального времени. Это существенно сокращает время вычисления.

Преимущества этой опции:

- Отсутствие ограничения доступа к таблице для других пользователей, подающих запросы.
- Запросы запускаются Asset Manager Automated Process Manager, а не другими пользовательскими станциями.

Внедрение этого способа

В сведениях о статистике активируйте опцию **Сохранить результаты** (bStored).

Настройте Asset Manager Automated Process Manager для регулярного автоматического сохранения результатов:

- 1 Запустите Asset Manager Automated Process Manager
- 2 Подключитесь к базе данных (меню **Файл/Подключение к базе данных**).
- 3 Откройте страницу настройки модулей (меню **Инструменты/Настройка модулей**).
- 4 Выберите модуль Обновление статистики для таблиц (Stats)
- 5 Установите флажок Включено.
- 6 Определите интервалы выполнения обновления.
- 7 Сохраните модификации (кнопка **Изменение**).

Затем Asset Manager Automated Process Manager будет запускать статистику через установленные графиком интервалы для всей статистики, для которой активирована опция **Сохранить результаты**. Затем результаты сохраняются в таблице Заметки по статистике (amStatMemo).

Масштабирование сведений о статистике

При отображении диаграммы можно получить доступ к каждой зоне сведений посредством нажатия на нужную зону. При этом открывается таблица статистики, отфильтрованная при помощи критериев группировки и фильтрации.

Определение максимального количества групп для индивидуального представления

Windows-клиент

Рекомендуется не использовать более двадцати групп значений, чтобы сохранять читаемость диаграммы.

Оставшиеся группы можно сгруппировать в группу Другие значения.

По умолчанию, а также для того, чтобы сохранять читаемость диаграмм, количество индивидуально представленных групп не должно превышать 20.

В таком случае будут представлены первые 20 групп значений, а остальные, если таковые имеются, будут сгруппированы в группу под названием Другие значения.

Отображенные на диаграмме группы значений сортируются по значению (кроме группы Другие значения, которая ставится на последнее место и включает в себя все группы, не вошедшие в число отображаемых групп).

Чтобы модифицировать максимальное количество представленных групп:

- 1 Откройте группы опций (меню **Редактировать/ Опции**).
- 2 Расширьте ветвь **Статистика/ Максимальное количество учитываемых значений**.
- 3 Определите количество групп для индивидуального представления.

Просмотр статистики

На инфопанелях

Прежде чем отобразить статистику, ее можно связать с инфопанелью следующим образом:

- 1 Отобразите действия (ссылка **Администрирование/ Действия** на навигационной панели).
- 2 Создайте новое действие (кнопка **Создать**).
- 3 Обязательно заполните поле **Тип** (seActionType), выбрав значение Инфопанель и поле **Область** (Domain), чтобы указать узел навигационной панели, на котором будет доступна данная инфопанель.

ВНИМАНИЕ:

Если не ввести значение в поле Область, инфопанель не будет отображаться в веб-клиенте.

- 4 Проверьте создание (кнопка **Создать**).
- 5 Во вкладке **Индикаторы** создайте индикаторы статистики, выполнив следующую процедуру:
 - a Щелкните .
 - b Заполните поле **Имя** (Name) индикатора. Можно использовать имена статистики, которую вы хотите вставить.
 - c В поле **Тип** (seType) выберите Статистика.
 - d Далее введите ссылку **Статистика**.

Выполняйте эти этапы каждый раз при добавлении статистики.

6 Обновите инфопанель (кнопка **Изменение**).

Инфопанель создана и ее можно использовать.

Чтобы использовать эту статистику в инфопанелях, прочитайте следующую главу: [Инфопанели](#) [стр. 103].

На домашних страницах компаний

Для использования статистики можно связать ее с домашней страницей компании следующим образом:

- 1 Создайте домашнюю страницу (ссылка **Администрирование/ Домашние бизнес-страницы** на навигационной панели).
- 2 Во вкладке **Статистика** введите необходимую статистику, щелкнув значок  (Windows-клиент) или кнопку **Добавить** (веб-клиент), а затем выбрав статистику.
 - Чтобы узнать больше о домашних страницах компании, прочитайте руководство Tailoring, раздел Customizing Web clients, глава Business home pages.

Отображение статистики

Статистика не может отображаться обособлено и должна отображаться при помощи инфопанели.

Прочитайте главу [Инфопанели](#) [стр. 103] этого руководства, чтобы узнать о том, как статистика используется в инфопанели.

Примеры статистики

В своих бизнес-данных Asset Manager приводит примеры статистики, сгенерированной в базе данных при помощи инфопанелей.

Эти инфопанели доступны в разных бизнес-данных.

Они доступны по умолчанию в демонстрационной базе данных.

Однако, чтобы они находились в рабочей базе данных, необходимо импортировать бизнес-данные, частью которых они являются.

- Пример импорта бизнес-данных приводится в руководстве Закупки, глава Общий обзор, раздел Предварительные этапы, параграф Импорт бизнес-данных в существующую базу данных.

Статистика, не заданная скриптом

Пример 1: статистика по средней цене компьютеров по центру затрат и частоте ЦП

Необходимо просмотреть статистику по средней цене компьютеров по центру затрат и частоте ЦП. Эта статистика:

- Представляет две группы: центр затрат и частоту ЦП
- Фильтр, используемый для компьютеров
- Использует выражение для вычисления средней цены

Чтобы создать статистику:

- 1 Отобразите статистику (ссылка **Администрирование/ Статистика...** на навигационной панели).
- 2 Создайте новую статистику (кнопка **Создать**).
- 3 Заполните поля и ссылки, показанные в следующей таблице:

Поле или ссылка (метка)	Поле или ссылка (имя SQL)	Значение
Имя	Name	Средняя цена компьютеров по центру затрат и частоте ЦП
Вкладка Общее		
Тип представления	seType	Горизонтальная гистограмма
Условные обозначения	bLegend	Установите этот флажок
Вкладка Запрос		
Таблица	TableName	Компьютеры (amComputer)
Группировать по	GroupBy	Portfolio.CostCenter.Title
Если ведено значение для предыдущего поля вместо его выбора из списка, необходимо проверить ввод или переместить курсор в другое поле, чтобы отобразилось следующее поле.		
Дополнительно группировать по	SubGroupBy	ICPUSpeedMHz
Использовать выражение	bUseAgregExpr	Установите этот флажок
Выражение	AqlAgregExpr	Avg(Portfolio.mAvgPrice)

- 4 Создайте статистику (кнопка **Создать**).
- 5 Для отображения диаграммы щелкните **Тест**.
- 6 Щелкните **ESC**, чтобы закрыть диаграмму.
- 7 Закройте все окна.

Пример 2: разбивка по половому признаку

Необходимо увидеть разбивку сотрудников компании по половому признаку. Группировку нужно выполнить, используя только один критерий (должность

сотрудника). Однако необходимо создать фильтр, используемый для отделения услуг, определенных в той же таблице, для того, чтобы эта информация не использовалась в статистике.

Начните с создания этого фильтра:

- 1 Отобразите запросы (ссылка **Администрирование/Запросы** на навигационной панели).
- 2 Создайте новый фильтр, заполнив поля и ссылки, указанные в следующей таблице:

Поле или ссылка (метка)	Поле или ссылка (имя SQL)	Значение
Имя	Имя	Только сотрудники
Таблица	TableName	Сотрудники и подразделения (amEmplDept)
Вкладка Запрос		
Query	memQueryText	bDepartment = 0

- 3 Щелкните Создать.
- 4 Закройте все окна.

Теперь создайте статистику:

- 1 Отобразите статистику (ссылка **Администрирование/ Статистика...** на навигационной панели).
- 2 Создайте новую статистику (кнопка **Создать**).
- 3 Заполните поля и ссылки, показанные в следующей таблице:

Поле или ссылка (метка)	Поле или ссылка (имя SQL)	Значение
Имя	Name	Разбивка по половому признаку
Вкладка Общее		
Условные обозначения	bLegend	Установите этот флажок
Тип представления	seType	Сектора
Объемное отображение	b3D	Установите этот флажок
Вкладка Запрос		
Таблица	TableName	Сотрудники и подразделения (amEmplDept)
Фильтр	Filter	Только сотрудники
Группировать по	GroupBy	MrMrs

- 4 Щелкните Создать.
- 5 Для отображения диаграммы щелкните **Тест**.
- 6 Щелкните **ESC**, чтобы закрыть диаграмму.
- 7 Закройте все окна.

Пример 3: разбивка сотрудников по подразделению и месту размещения

В этом примере необходимо увидеть разбивку сотрудников по подразделению и месту размещения. Информация группируется при помощи двух критериев (подразделение и место размещения), а фильтр применяется только к сотрудникам (как и в предыдущем примере).

- 1 Отобразите статистику (ссылка **Администрирование/ Статистика...** на навигационной панели).
- 2 Создайте новую статистику (кнопка **Создать**).
- 3 Заполните поля и ссылки, показанные в следующей таблице:

Поле или ссылка (метка)	Поле или ссылка (имя SQL)	Значение
Имя	Name	Разбивка сотрудников по местоположению
Вкладка Общее		
Тип представления	seType	Горизонтальная гистограмма
Условные обозначения	bLegend	Установите этот флажок
Вкладка Запрос		
Таблица	TableName	Сотрудники и подразделения (amEmplDept)
Фильтр	Filter	Только сотрудники
Группировать по	GroupBy	Parent.Name
Если ведено значение для предыдущего поля вместо его выбора из списка, необходимо проверить ввод или переместить курсор в другое поле, чтобы отобразилось следующее поле.		
Дополнительно группировать по	SubGroupBy	Location.FullName
Вернитесь к вкладке Общее .		
Многоуровневое представление	bStacked	Установите этот флажок

- 4 Щелкните Создать.
- 5 Для отображения диаграммы щелкните **Тест**.
- 6 Щелкните **ESC**, чтобы закрыть диаграмму.
- 7 Закройте все окна.

Пример: 4 разбивка сотрудников по времени работы в компании

В этом примере необходимо разделить данные по приоритету. Как и в предыдущих примерах, сотрудники фильтруются и группируются по приоритету, если это необходимо. Приоритет, не является значением, сохраненным в базе данных. Но в таблице **Подразделение и сотрудники** есть поле, указывающее начало работы сотрудника в компании (**Дата приема на работу**). Это значение может использоваться для вычисления

приоритета сотрудника. Это выполняется с использованием вычисляемого поля, значение которого затем используется для группировки информации.

Чтобы создать вычисляемое поле:

- 1 Отобразите вычисляемые поля (ссылка **Администрирование/Система/Вычисляемые поля** на навигационной панели).
- 2 Создайте новую запись со следующей информацией:

Поле или ссылка	Значение
Имя	Время работы сотрудника в компании
Таблица	Сотрудники и подразделения (amEmplDept)
Тип поля	AQL
Тип результата	Числовое
Синтаксис AQL	Round((DaysDiff(GetDate(), dHire) / 365), 0) Запросом AQL используется функция DaysDiff() , с помощью которой рассчитывается продолжительность времени между: ■ временем выполнения расчета: функция GetDate(); ■ и датой приема на работу: значением поля Дата приема на работу (dHire). Продолжительность указывается в днях. Результат делится на 365, что соответствует количеству дней в году. Затем результат округляется до ближайшего целого числа при помощи функции Округлить() .

- 3 Проверьте создание (кнопка **Создать**).
 - 4 Закройте все окна.
- Теперь создайте статистику:
- 1 Отобразите статистику (ссылка **Администрирование/ Статистика...** на навигационной панели).
 - 2 Создайте новую статистику (кнопка **Создать**).
 - 3 Заполните поля и ссылки, показанные в следующей таблице:

Поле или ссылка (метка)	Поле или ссылка (имя SQL)	Значение
Имя	Имя	Разбивка сотрудников по времени в компании
Вкладка Общее		
Условные обозначения	bLegend	Установите этот флажок
Тип представления	seType	Кривые
Вкладка Запрос		
Таблица	TableName	Сотрудники и подразделения (amEmplDept)
Фильтр	Filter	Только сотрудники

Поле или ссылка (метка)	Поле или ссылка (имя SQL)	Значение
Группировать по	GroupBy	Выберите вычисляемое поле Приоритет, созданное ранее

- 4 Щелкните **Создать**.
- 5 Щелкните **Тест...** для отображения диаграммы.
- 6 Щелкните **ESC**, чтобы закрыть окно диаграммы.
- 7 Закройте все окна.

Статистика, заданная скриптом

Ниже приведены примеры статистики, заданной скриптом, которые можно тестировать или модифицировать с целью создания собственной статистики.

Другие примеры статистики, заданной скриптом, можно найти в демонстрационной базе данных (выберите ссылку **Администрирование/ Статистика...** на навигационной панели).

Пример 1 (базовый)

Целью этого примера является демонстрация способа создания простой статистики (данные не извлекаются из базы данных) и способа управления представлением информации.

В этом примере создаются три группы (Франция, Германия и Англия). В каждой группе определенное количество сотрудников (10, 20 и 18); нам нужно, чтобы диаграмма отображала количество сотрудников в стране.

Необходимо сгенерировать следующий запрос:

```
<group Name="France" filtertable="amEmplDept" filter="Location.FullName like '/France/%'" value="10"/> <group Name="Germany" filtertable="amEmplDept" filter="Location.FullName like '/Germany/%'" value="20"/> <group Name="England" filtertable="amEmplDept" filter="Location.FullName like '/England/%'" value="18"/>
```

Для генерирования этого запроса будет использоваться опция Скрипт (установите соответствующий флажок). Эта опция позволяет писать скрипт, генерирующий статистические данные. Интерфейс API XmlAttribute будет использоваться для преобразования предварительно определенных сущностей XML (► см. Programmer's Reference для получения подробной информации).

- 1 Отобразите статистику (ссылка **Администрирование/ Статистика...** на навигационной панели).
- 2 Создайте новую статистику (кнопка **Создать**).
- 3 Заполните поля и ссылки, показанные в следующей таблице:

Поле или ссылка (метка)	Поле или ссылка (имя SQL)	Значение
Имя	Name	Количество сотрудников в стране
Скрипт	bScript	Установите этот флажок
Вкладка Общее		
Тип представления	seType	Вертикальная гистограмма
Условные обозначения	bLegend	Установите этот флажок
Вкладка Скрипт		
Скрипт данных	Скрипт	RetVal = "<group " & XmlAttribute("Name", "France") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/France/%') & " " & XmlAttribute("value", "10") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<group " & XmlAttribute("Name", "Germany") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/Germany/%') & " " & XmlAttribute("value", "20") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<group " & XmlAttribute("Name", "England") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/England/%') & " " & XmlAttribute("value", "18") & " />" & Chr(13) & Chr(10)

- 4 Щелкните Создать.
- 5 Щелкните **Тест...** для отображения диаграммы.
- 6 Щелкните **ESC**, чтобы закрыть окно диаграммы.
- 7 Закройте все окна.

Пример 2 (с подгруппой)

В предыдущий пример была добавлена подгруппа, соответствующая обращению, которое используется в отношении каждого сотрудника (Г-н или Г-жа).

Теперь необходимо просмотреть диаграмму, в которой представлено количество сотрудников по типу обращения и стране.

Необходимо сгенерировать следующий запрос:

```
<group Name="Mr."> <subgroup Name="France" filtertable="amEmplDept" filter="Location.FullName like '/France/%' and MrMrs='Mr'" value="5"/> <subgroup Name="Germany" filtertable="amEmplDept" filter="Location.FullName like '/Germany/%' and MrMrs='Mr'" value="9"/> <subgroup Name="England" filtertable="amEmplDept" filter="Location.FullName like '/England/%' and MrMrs='Mr'" value="10"/> </group>

<group Name="Mrs."> <subgroup Name="France" filtertable="amEmplDept" filter="Location.FullName like '/France/%' and MrMrs='Mrs'" value="3"/> <subgroup Name="Germany" filtertable="amEmplDept" filter="Location.FullName like '/Germany/%' and MrMrs='Mrs'" value="9"/> <subgroup Name="England" filtertable="amEmplDept" filter="Location.FullName like '/England/%' and MrMrs='Mrs'" value="10"/> </group>
```

```
ame="Germany" filtertable="amEmplDept" filter="Location.FullName like '/Germany/%' and MrMrs='Mrs'" value="7"/> <subgroup Name="England" filtertable="amEmplDept" filter="Location.FullName like '/England/%' and MrMrs='Mrs'" value="7"/> </group>

<group Name="Miss"> <subgroup Name="France" filtertable="amEmplDept" filter="Location.FullName like '/France/%' and MrMrs='Miss'" value="2"/> <subgroup Name="Germany" filtertable="amEmplDept" filter="Location.FullName like '/Germany/%' and MrMrs='Miss'" value="4"/> <subgroup Name="England" filtertable="amEmplDept" filter="Location.FullName like '/England/%' and MrMrs='Miss'" value="1"/> </group>
```

Как и предыдущем примере, используется опция Скрипт для кодирования этой информации.

- 1 Отобразите статистику (ссылка **Администрирование/ Статистика...** на навигационной панели).
- 2 Создайте новую статистику (кнопка **Создать**).
- 3 Заполните поля и ссылки, показанные в следующей таблице:

Поле или ссылка (метка)	Поле или ссылка (имя SQL)	Значение
Имя	Name	Количество сотрудников по типу обращения и стране
Скрипт	bScript	Установите этот флажок
Вкладка Общее		
Условные обозначения	bLegend	Установите этот флажок
Тип представления	seType	Вертикальная гистограмма
Вкладка Скрипт		

Поле или ссылка (метка)	Поле или ссылка (имя SQL)	Значение
Скрипт данных	Скрипт	<pre>RetVal = "<group " & XmlAttribute("Name", "Mr.") & " ">"; Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "France") & " "& XmlAttribute("filtertable", "amEmplDept") & " "& XmlAttribute("filter", "Location.FullName like '/France/%' and MrMrs like '/Mr/%' ") & " "& XmlAttribute("value", "5") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "Germany") & " "& XmlAttribute("filtertable", "amEmplDept") & " "& XmlAttribute("filter", "Location.FullName like '/Germany/%' and MrMrs like '/Mr/%' ") & " "& XmlAttribute("value", "9") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "England") & " "& XmlAttribute("filtertable", "amEmplDept") & " "& XmlAttribute("filter", "Location.FullName like '/England/%' and MrMrs like '/Mr/%' ") & " "& XmlAttribute("value", "10") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "</group>"</pre> <pre>RetVal = RetVal & "<group " & XmlAttribute("Name", "Mrs.") & " ">"; Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "France") & " "& XmlAttribute("filtertable", "amEmplDept") & " "& XmlAttribute("filter", "Location.FullName like '/France/%' and MrMrs like '/Mrs/%' ") & " "& XmlAttribute("value", "3") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "Germany") & " "& XmlAttribute("filtertable", "amEmplDept") & " "& XmlAttribute("filter", "Location.FullName like '/Germany/%' and MrMrs like '/Mrs/%' ") & " "& XmlAttribute("value", "7") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "England") & " "& XmlAttribute("filtertable", "amEmplDept") & " "& XmlAttribute("filter", "Location.FullName like '/England/%' and MrMrs like '/Mrs/%' ") & " "& XmlAttribute("value", "7") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "</group>"</pre>

Поле или ссылка (метка)	Поле или ссылка (имя SQL)	Значение
		RetVal = RetVal & "<group " & XmlAttribute("Name", "Miss") & " >" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "France") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/France/%' and MrMrs like '/Miss/%' ") & " " & XmlAttribute("value", "2") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "Germany") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/Germany/%' and MrMrs like '/Miss/%' ") & " " & XmlAttribute("value", "4") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "<subgroup " & XmlAttribute("Name", "England") & " " & XmlAttribute("filtertable", "amEmplDept") & " " & XmlAttribute("filter", "Location.FullName like '/England/%' and MrMrs like '/Miss/%' ") & " " & XmlAttribute("value", "1") & " />" & Chr(13) & Chr(10) RetVal = RetVal & "</group>"

- 4 Щелкните Создать.
- 5 Щелкните **Тест...** для отображения диаграммы.
- 6 Щелкните **ESC**, чтобы закрыть окно диаграммы.
- 7 Закройте все окна.

Пример 3 (продвинутый пользователь)

В этом более сложном примере данные экспортируются из базы данных Asset Manager.

В примере отображается сводка всех сеансов обучения в базе данных, отсортированных по модели обучения:

- Количество дней обучения
- Стоимость программ обучения
- Количество сеансов обучения: один сеанс обучения на человека, участвующего в данном сеансе

Эти типы статистики, заданной скриптом, всегда пишутся по одинаковой схеме. В этом примере представлена сопровождающаяся комментариями модель статистики, заданной скриптом, которую можно тестировать и модифицировать так, как необходимо для вашей статистики.

Как и в предыдущих примерах, создайте новую статистику, заданную скриптом (установите флажок Скрипт).

Этап 1: определение необходимых переменных

Сначала нужно определить переменные, которые будут использоваться для создания статистики.

Должна быть построена структура статистики:

1 Выбрана основная таблица, используемая для извлечения и отображения данных. В этом примере выбрана таблица **Обучение** (amTraining).

2 Выберите один из вариантов построения:

- Группировка при помощи нескольких выражений.
- Группировка при помощи нескольких критериев в одном выражении.

В этом примере данные будут группироваться при помощи одного критерия (модель обучения) и трех выражений (количество дней обучения, стоимость сеансов обучения и количество сеансов обучения).

Столбцы соответствуют моделям обучения, используемым для основной группы.

Сериі соответствуют разным выражениям (количество сеансов обучения, количество дней обучения и стоимость сеансов обучения).

Введите следующий текст в поле Скрипт данных, расположенное во вкладке статистики Скрипт:

```
SetMaxInst(1000000000) const CONST_NumberSecondsPerMonth = 2592000 c
onst CONST_NumberSecondsPerDay = 86400

Dim strRC as String strRC = Chr(13) & Chr(10)

RetVal = ""

Dim strTable, strColumnField, strSerieField as String

strTable = "amTraining"

Dim strColumns, strSeries as String strColumns = AmDbGetStringEx("SELECT D
ISTINCT Model.Name FROM amTraining WHERE Model.Name <> "" , "а", "|") str
Series = "Number of training sessions" & "|" & "Number of training days" & "|" &
; "Cost"
```

 ЗАМЕЧАНИЕ:

Можно определить:

- Основной фильтр для выбора определенных записей в данной таблице.
- Основное выражение используется для выполнения операций с записями.

В таком случае, после получения значений для разных серий (в параграфе **Этап 2: получение значений** [стр. 124]) необходимо добавить основной фильтр к фильтрам, используемым для получения значений, а также добавить основное выражение к каждому выражению подгруппы.

В примере эти функции не используются, так как они не нужны. Примеры использования этих функций см. в базе данных.

 ЗАМЕЧАНИЕ:

При написании статистики, если необходимо выполнить операции с датами, а не с выражениями, как описано в этом примере, используйте библиотеку скриптов StatLib (доступ к этой библиотеке можно получить по ссылке **Администрирование/ Скрипты** на навигационной панели).

В решении имеется три предварительно определенных функции, которые можно использовать для определения столбцов (strColumns): CURRENTFISCALYEAR, LAST12MONTHS и NEXT12MONTHS.

Чтобы узнать, как используются эти функции, прочитайте их описание в библиотеке скриптов (ссылка **Администрирование/ Скрипты** на навигационной панели).

Этап 2: получение значений

На этом этапе осуществляется получение значений для разных групп и серий, которые были определены.

- Для получения этих значений для группы моделей обучения в переменной strColumns введите следующее:

```
Dim iColumn, iColumnCount as Long
iColumnCount = CountValues(strColumns,"|")

For iColumn = 1 to iColumnCount
Dim strColumn as String
strColumn = GetListItem(strColumns, "|", iColumn)
Dim iColumnValue as Long
```

- Для каждой из трех серий, определенных в strSeries (количество сеансов обучения, количество дней обучения и стоимость сеансов обучения),

необходимо определить фильтр (strFilter) и выражение (strQuery) для вычисления данных и получения значений:

```
Dim iSerie, iSerieCount as Long iSerieCount = CountValues(strSeries, "|")

dim strFilter as String dim strQuery as String dim ISerieValue as Long

for iSerie = 1 to iSerieCount Dim strSerie as String strSerie = GetListItem(strSeries, "|", iSerie)

if "Number of training sessions" = strSerie then strFilter = " Model.Name = " &
; AmSQLTextConst(strColumn) strQuery = "SELECT Count(ITrainingId) FROM amTraining WHERE " & strFilter

elseif "Number of training days" = strSerie then strFilter = " Model.Name = " &
; AmSQLTextConst(strColumn) strQuery = "SELECT Sum(tsDuration)/" & CONST_NumberSecondsPerDay & " FROM amTraining WHERE " & strFilter

elseif "Cost" = strSerie then strFilter = " Model.Name = " & AmSQLTextConst(strColumn) strQuery = "SELECT Sum(mCost+mTax) FROM amTraining WHERE " & strFilter

end if

ISerieValue = AmDbGetLong(strQuery)
```

Этап 3: определение групп и подгрупп

Данные о столбцах определяются по группам, а серии - по подгруппам.

```
dim subgroup as String
subgroup = subgroup & "<subgroup " & XmlAttribute("name", strSerie) & " " & XmlAttribute("filter", strFilter) & " " & XmlAttribute("value", ISerieValue) & "/> "
IColumnValue = IColumnValue + ISerieValue
next

RetVal = RetVal & "<group " & XmlAttribute("name", strColumn) & " " & XmlAttribute("value", IColumnValue) & "> "
RetVal = RetVal & subgroup
RetVal = RetVal & "</group> "
next
```

Этап 4: просмотр статистики

Щелкните **Тест...** для отображения статистики.

Определение настраиваемых цветов для статистики в виде диаграммы

Можно задать настраиваемые цвета, используемые для запроса со скриптами, который возвращает результаты статистики в виде диаграммы для набора подгрупп или серий. (Имеет смысл определять цвета только в запросах, включающих несколько подгрупп).

Чтобы определить цвет для серии, используйте параметр color= с инструкцией определения подгруппы. Пример.

```
<subgroup Name="Germany" filtertable="amEmplDept" filter="Location.FullName like '/Germany/%' and MrMrs='Mr'" value="9" color="#cc0066"/>
```

 ЗАМЕЧАНИЕ:

Чтобы обеспечить совместимость с клиентами Windows и веб-клиентами Asset Manager, цвета необходимо указывать в шестнадцатеричном эквиваленте их значений RGB, например, #6f006f для фиолетового цвета.

Стандартные графические цвета представлены следующими значениями (тем не менее, можно указывать другие цвета с помощью его шестнадцатеричного эквивалента значения RGB):

 standard graphic colors

Color specifications for standard graphics		
 Blue R:28 G:107 B:227 (Hex:1C6BE3)	 Aqua R:51 G:166 B:182 (Hex:33A6B6)	 Gray R:156 G:156 B:156 (Hex:9C9C9C)
 R:186 G:210 B:246 (Hex:BAD2F6)	 R:168 G:215 B:223 (Hex:A8D7DF)	 R:215 G:215 B:215 (Hex:D7D7D7)
 Dark Blue R:15 G:71 B:128 (Hex:0F4780)	 Magenta R:204 G:0 B:102 (Hex:CC0066)	 Dark Green R:51 G:102 B:51 (Hex:336633)
 R:183 G:197 B:212 (Hex:B7C5D4)	 R:255 G:173 B:198 (Hex:FFADC6)	 R:193 G:209 B:193 (Hex:C1D1C1)
 Purple R:138 G:40 B:145 (Hex:8A2891)	 Brown R:146 G:93 B:73 (Hex:925D49)	
 R:202 G:163 B:205 (Hex:CAA3CD)	 R:212 G:183 B:174 (Hex:D4B7AE)	

Обработка ошибок

Для разрешения отклонений в определении цветов используются следующие правила.

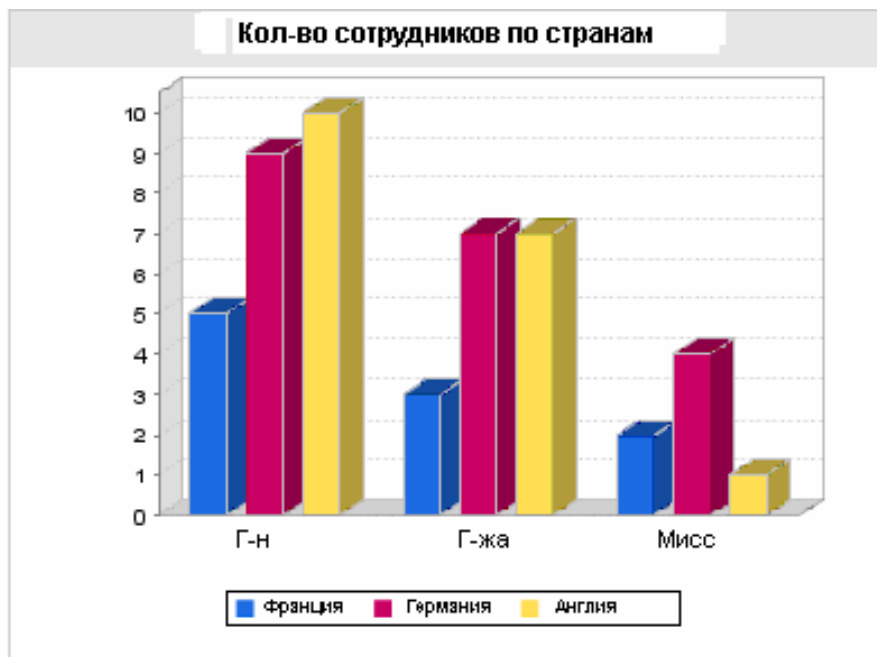
- Если параметр color= не определен для одной или нескольких подгрупп в запросе, к данным подгруппам будут применяться цвета по умолчанию.
- Если параметр color= использует нераспознанное значение, будет применяться цвет по умолчанию.
- Цвета для подгруппы можно определить только один раз. (Любая попытка переопределить цвета будет проигнорирована).

Практический пример

В приведенном ниже запросе определены следующие настраиваемые цвета: синий, фиолетовый и желтый для трех подгрупп группы Mr.

```
<group Name="Mr."> <subgroup Name="France" filtertable="amEmplDept" filter="Location.FullName like '/France/%' and MrMrs='Mr'" value="5" color="#1c6be3" /> <subgroup Name="Germany" filtertable="amEmplDept" filter="Location.FullName like '/Germany/%' and MrMrs='Mr'" value="9" color="#cc0066"/> <subgroup Name="England" filtertable="amEmplDept" filter="Location.FullName like '/England/%' and MrMrs='Mr'" value="10" color="#ffde53"/> </group>
```

В результате запроса получается следующая диаграмма.



7 Действия

В данной главе объясняется, как определять действия с помощью Asset Manager.

Для создания действий используется пункт меню

Инструменты/Действия/Изменить.

Можно выполнять действия через меню Инструменты/Действия или контекстный список «Действия» на панели инструментов.

Рекомендации по разработке

Использование тестовой функциональной области для настройки действий

Рекомендация

Рекомендуется связывать новое действие или новую версию действия (например, мастера) с тестовой функциональной областью перед тем, как сделать их видимыми.

Можно также маскировать тестовую функциональную область, чтобы скрывать от пользователей разрабатываемые действия.

Обоснование

Пока действие неправильно, функциональная область, с которой оно связано, не создается как веб-служба для Asset Manager Web.

Соответственно, все объекты (даже и действительные), связанные с функциональной областью, исчезнут из веб-служб, предоставляемых Asset Manager Web.

Реализация

Чтобы связать действие с тестовой функциональной областью:

- 1 Запустите клиент Windows или веб-клиент.
- 2 Отобразите функциональные области (ссылка **Администрирование/Функциональные области** на навигационной панели).
- 3 Создайте тестовую функциональную область.
Чтобы убедиться, что для данной функциональной области не создано веб-службы, заполните поле **Веб-служба** (seWebService) значением нет. Запомните имя SQL тестовой функциональной области.
- 4 Отобразите действия (ссылка **Администрирование/Действия** на навигационной панели).
- 5 Свяжите действие с тестовой функциональной областью (ссылка **Область** (Domain)).
- 6 После того, как объект был протестирован и проверен, свяжите его с соответствующей предоставляемой функциональной областью.

Пометка веб-служб

Если вы модифицируете действия в отношении скрипта или мастера, обязательно повторно отметьте веб-службы, если вы выполняли развертывание Asset Manager Web.

► Руководство Tailoring, глава Customizing the database, раздел Development best practices/ Tag the Web services.

Определение действия.

Действие позволяет автоматизировать, или полностью, или частично, задачи, выполняемые для базы данных Asset Manager.

Имеются несколько типов действий:

- исполняемые;

- DDE
- обмен сообщениями;
- Скрипт: модификация объекта в базе данных Asset Manager
- мастер;
- печать;
- распределение ПО;
- действие;
- инфопанель;

Действия должны определяться первыми, чтобы выполняться при их выборе в списке.

ЗАМЕЧАНИЕ:

Можно определить область для действия, а также распределять по категориям области в соответствии с их функциями с помощью функциональных областей.

Функциональная область

Asset Manager позволяет определять области, в которых группируются вместе функции ПО. По умолчанию с ПО поставляются некоторые функциональные области: они соответствуют модулям, которые можно активировать или деактивировать с помощью элемента меню «Файл/Активировать модули».

Функциональные области используются для создания и распределения по категориям информации, отображаемой в навигационной панели клиента Windows и веб-клиента. Таким образом, после выбора для действия функциональной области данное действие появится в навигационной панели клиента Windows и веб-клиента под заголовком данной функциональной области.

ЗАМЕЧАНИЕ:

Содержимое навигационной панели реорганизуется и изменяется в соответствии с контекстом. Если действие контекстное (например, оно не может выполняться, если не открыт конкретный экран), то оно не появится в навигационной панели, если текущий контекст не соответствует его определениям (например, если в настоящее время не отображается конкретный экран).

Чтобы определить функциональную область:

- 1 Выберите пункт меню **Администрирование/Функциональные области**.

- 2 Нажмите кнопку **Создать**.
- 3 Назначьте **Имя** своей функциональной области. Данное имя - имя, которое будет появляться в навигационной панели клиентов Windows и веб-клиентов. По умолчанию, Asset Manager назначает данной функциональной области **Имя SQL**; можно изменить это значение, если необходимо.
- 4 Можно также выбрать **Родительскую область** для функционального домена, если необходимо.
- 5 Проверьте созданное, нажав на кнопку **Создать**.



ВНИМАНИЕ:

Если доступ к функциональной области полностью запрещен (для считывания и для записи) для каких-либо пользователей, они не смогут вызывать действия, представления или отчеты в данной области. На практике рекомендуется разрешить доступ для чтения к следующим таблицам при определении функциональной области:

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

Создание действия

В данном разделе описывается создание действия:

- типы действий;
- общий метод;
- заполнение вкладки DDE;
- заполнение вкладки «Обмен сообщениями».

Типы действий

Asset Manager позволяет определять несколько типов действий:

- исполняемые;
- DDE
- обмен сообщениями;

- скрипт;
- мастер;
- распределение ПО;
- печать;
- действие;
- инфопанель;

Исполняемые действия

Исполняемое действие вызывает выполнение программ.

Им запускаются файлы приложений с расширениями .exe, .com, .bat или .pif. Можно также давать ссылку на любой тип документа, если расширение его имени связано с каким-либо приложением в диспетчере файлов.

Действия DDE

Действием DDE отправляется запрос DDE к приложению сервера DDE (или совместимому с DDE приложению), способному обрабатывать запросы DDE.

Аббревиатура DDE, означающая Dynamic Data Exchange (динамический обмен данными), обозначает метод для динамического обмена информацией между программами. Asset Manager использует DDE для команд, выполняемых другим приложением.

Пример: с помощью DDE можно дать приложению Microsoft Word команду открыть файл с указанными именем и содержимым.

Действия обмена сообщениями

Действия обмена сообщениями позволяют отправлять сообщения с помощью:

- внутренней системы обмена сообщениями Asset Manager;
- внешней системы обмена сообщениями стандарта VIM (Lotus Notes, Lotus cc:Mail и так далее);
- внешней системы обмена сообщениями стандарта MAPI (Microsoft Exchange, Microsoft Outlook и так далее);
- внешней системы обмена сообщениями стандарта SMTP.



ВНИМАНИЕ:

Можно отправлять сообщения только через те системы обмена сообщениями, к которым возможно подключение.

Для отправки сообщения по стандарту VIM, MAPI или SMTP Asset Manager использует:

- поля Учетная запись (имя SQL: MailLogin) и Пароль (имя SQL: MailPassword) вкладки Обмен сообщениями сведений о сотруднике, открывающем базу

данных Asset Manager (таблица подразделений и сотрудников), чтобы идентифицировать отправителя сообщения;

- поле Электронная почта (имя SQL: EMail) на вкладке Общее сведений о сотруднике, чтобы идентифицировать получателя сообщения.

Для отправки сообщений с помощью внутренней системы обмена сообщениями Asset Manager, Asset Manager использует поля Имя входа и Пароль на вкладке Профиль сведений отправителя и получателя.



ЗАМЕЧАНИЕ:

Адрес пользователя Asset Manager во внутренней системе обмена сообщениями такой же, что и его Имя входа.



ВНИМАНИЕ:

Администратор должен создать пользователя с именем Admin и заполнить поля Учетная запись, Электронная почта и Пароль, чтобы использовать внешнюю систему обмена сообщениями и удостовериться в правильном функционировании Asset Manager Automated Process Manager.

Действия типа «Скрипт»

Действия типа Скрипт позволяют выполнять любую операцию для базы данных Asset Manager. Они дают опытным пользователям обширные возможности управления базой данных, позволяя им выполнять операции, которые не могут выполняться с действиями других типов, в частности:

- создание записей;
- удаление записей;
- дублирование записей;
- изменение одного или нескольких объектов в базе данных Asset Manager, например, всех записей в таблице, в поле или в ссылке.

Операции, выполняемые действием данного типа, описываются скриптом на языке Basic; это позволяет использовать сложные функции, подобные функциям в Asset Manager API.



ЗАМЕЧАНИЕ:

Сложность функций, используемых действием типа Скрипт, связанная с возможностью вносить глубокие изменения в базу данных, делают действия такого типа потенциально опасными для целостности базы данных. Поэтому их следует использовать только опытным пользователям.

Для изменения значения объектов базы данных используются различные функции, в зависимости от контекста действия:

- Если у действия нет контекста, необходимо использовать функции, производные от Asset Manager API, например, AmSetFieldStringValue() или AmSetFieldLongValue().
- Если контекст действия - таблица, можно использовать функцию Set(); она имеет следующий синтаксис:

`Set [<Link.Link.Field>]=<Value>`

Действия типа «Мастер»

Мастера представляют собой сложные действия. Дополнительную информацию можно найти в главе Мастера данного руководства.

Мастера предназначены для управления пошаговым выполнением сложных или повторяющихся задач в Asset Manager. Мастера создаются с помощью специального языка программирования.

Действие типа «Печать»

Действие типа Печать позволяет напечатать отчет или форму.

Необходимо заполнить следующие поля для действия такого типа:

- Поле **Тип** (seFormType): позволяет указывать тип отчета для печати.
- Поле **Отчет**: позволяет указывать отчет, который необходимо использовать;



ЗАМЕЧАНИЕ:

Контекст действия определяется контекстом, определенным для отчета.

Действие типа «Распределение ПО»

Действие типа Распределение ПО позволяет создавать запланированную задачу для нескольких компьютеров.

Контекст действий таких типов включает таблицу **Компьютеры** (amComputer). Действия таких типов могут выполняться, только если выбран компьютер.

Изменять контекст нельзя.

Поля, которые необходимо заполнять для действий такого типа, находятся на вкладке **Распределение**.

Дополнительную информацию о распределении ПО можно найти в руководстве ► Software distribution.

Для выполнения действия типа Распределение ПО:

- 1 Отобразите компьютеры (ссылка **Управление портфелем/ИТ/Оборудование/Компьютеры** на навигационной панели).
- 2 Выберите целевые группы компьютеров (в таком случае, будут выбраны все компьютеры в группе, даже если они не появляются в интерфейсе) и (или) целевые компьютеры для создаваемой запланированной задачи.
- 3 Выполните действие (меню **Инструменты/Действия/<Имя действия>**).

Создается только одна запланированная задача. Целевым для данной задачи является каждый из компьютеров, выбранных до выполнения действия, с заполненным полем **Идентификатор распределения ПО** (SWDID).

Действие типа «Действие»

Тип действия Действие позволяет выполнять действие для набора записей, выбранных запросом.

Поля, которые необходимо заполнять для действий такого типа, находятся на вкладке **Действие**.

Пример приложения:

- Создайте действие типа Распределение ПО.
Данным действием создается запланированная задача для нескольких компьютеров.
- Создайте действие типа Действие, чтобы выполнить действие типа Распределение ПО для компьютеров, выбранных запросом выбора действия типа Действие.
Запрос выбора используется для автоматического выбора целевых компьютеров.

Действие типа «Инфопанель»

Действие типа Инфопанель позволяет создать инфопанель, содержащую статистику и отчеты. У данного действия нет контекста.

Вкладка **Индикаторы** содержит список объектов, отображаемых инфопанелью. Можно добавлять объекты на данную вкладку или удалять их с нее.

Создание действия

Чтобы создать действие:

- 1 Выберите пункт меню **Инструменты/Действия/Изменить**.
- 2 Нажмите кнопку **Создать**.
- 3 Введите имя действия.

- 4 В поле Тип (имя SQL: seActionType) укажите тип действия, которое необходимо создать. Типом выбираемого действия управляется отображение одной из следующих вкладок:
- Исполняемое или DDE
 - Обмен сообщениями
 - Скрипт
 - Мастер
 - Печать
 - Распределение ПО
 - Действие
 - Индикаторы
- 5 Можно заполнить поле Имя SQL (имя SQL: SQLName) сведений о действии, но не обязательно. Имя SQL представляет собой уникальный способ идентификации действия; в частности, он используется, когда действие выполняется с помощью команды DDE (в случаях, когда Asset Manager используется в качестве сервера команд DDE).



ЗАМЕЧАНИЕ:

Если поле Имя SQL не заполнено, оно автоматически заполняется приложением.

- 6 Заполните поле Контекст (имя SQL: ContextTable):
- Если выбирается таблица из раскрывающегося списка, действие зависит от контекста: оно будет предлагаться только при отображении списка записей в данной таблице или сведений об одной из данных записей.
 - Если действие не зависит от таблицы, выберите параметр (Нет таблиц) наверху раскрывающегося списка.
- 7 Заполните поле **Область**, которое позволяет указывать функциональную область, к которой принадлежит действие. Данное действие появится для данного домена в навигационной панели клиента Windows или веб-клиента.
- 8 Можно связать с действием значок, но не обязательно.
- Для этого используется квадрат в верхней левой части экрана сведений о действии. Данное изображение будет затем появляться в контекстном списке «Действия» на панели инструментов. Активный значок в списке (значок, отображаемый на экране по умолчанию) - значок последнего действия, выполненного с помощью панели инструментов.
- 9 Заполните поля на вкладке Описание и поля на соответствующей вкладке для типа действия, которое необходимо создать.
- 10 Нажмите кнопку **Создать**.



ЗАМЕЧАНИЕ:

Администратор Asset Manager может просматривать все действия, независимо от того, предназначены ли они для совместного использования или нет, и были ли они созданы администратором или кем-либо еще.

Заполнение вкладки DDE

Информация о конкретном действии DDE находится на вкладке DDE сведений о действии.

Данная вкладка отображается, только если назначено значение DDE полю Тип (имя SQL: seActionType) в основной информации о действии.

Механизмы DDE основаны на службах, предоставляемых программами. Для выполнения механизмов DDE необходимо определить тему, которой указывается контекст, в котором должны выполняться команды.

Поэтому необходимо указать:

- В поле Служба (имя SQL: DDEService) - имя службы DDE, обеспечиваемой исполняемым файлом, который необходимо вызвать. Обычно это уникальная служба для исполняемого файла. Список предоставляемых служб можно найти в документации на исполняемые файлы.
- В поле Тема (имя SQL: DDETopic) - контекст, в котором должно выполняться действие.
- В поле Команда (имя SQL: DDECommand) - команды, которые должны выполняться внешним приложением.

В случае приложения Word команда может быть командой на языке Word Basic или Visual Basic.

Можно указать несколько команд, если это допускается службой DDE приложения.

Необходимо соблюдать синтаксис, требуемый внешним приложением.

- Если служба не представлена, в поле Файл (имя SQL: ActionFile) необходимо указать файл, используемый для запуска приложения, которым активизируется служба. Это основное приложение, которое реагирует на команды DDE.
- В поле Папка указывается путь к файлу, которым активизируется служба.
- В поле Параметры указываются параметры, передаваемые в программу, выполняющую службу DDE.

Важное примечание

Команды, отправляемые во внешнее приложение, заключаются в квадратные скобки («[», «]»). Например (для Microsoft Word):

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

- Если действие является контекстным, можно использовать переменные для ссылки на значение поля в базе данных. Так как такие переменные тоже заключаются в квадратные скобки, Asset Manager не может различать команды и переменные автоматически. Необходимо идентифицировать команды, указывая символ обратной черты «\» перед квадратными скобками. Соответственно предыдущий пример записывается (в случае контекстного действия) как:

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

Можно объединять команды и переменные, как показано ниже (в данном случае контекстом является таблица активов):

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

- Если действие не является контекстным, такая проблема не возникает. Текст, заключенный в квадратные скобки, все также рассматривается как команды, отправляемые внешнему приложению.

Заполнение вкладки «Обмен сообщениями»

Информация о действии обмена сообщениями находится на вкладке «Обмен сообщениями» сведений о действии.

Данная вкладка отображается, только если полю Тип в основной информации о действии присвоено значение Обмен сообщениями.



ВНИМАНИЕ:

Чтобы система функционировала правильно, системная переменная PATH должна включать папку, содержащую VIM DLL (VIM.DLL) и MAPI (MAPI.DLL).

Использование поля «Ссылочный объект»

Данное поле используется для выбора ссылки из таблицы, выбранной в поле Контекст.

Данное поле используется только для сообщений, пересылаемых через внутреннюю систему обмена сообщениями Asset Manager. Оно позволяет непосредственно обращаться к объекту, который инициирует отправку сообщения, просто щелкнув кнопку **Ссылочный объект** в сведениях о сообщении. Если ссылочный объект - непосредственно запись, которой

инициируется действие, поле Ссылочный объект (имя SQL: RefObject) не заполняется.

Получение подтверждения

Если необходимо, чтобы отправитель сообщения получил подтверждение через свою обычную службу обмена сообщениями, следует установить флажок Подтверждение (имя SQL: bAcknowledgment).

Это подтверждение будет отослано по адресу, указанному в поле Электронная почта (имя SQL: EMail) во вкладке Общее и относящемуся к сотруднику, открывшему базу данных Asset Manager (в таблице «Сотрудники и подразделения»).



ЗАМЕЧАНИЕ:

Нельзя получить подтверждение на сообщение, посланное через внутреннюю систему обмена сообщениями Asset Manager или через систему обмена сообщениями MAPI или SMTP.

Указание адреса

Есть несколько способов указания адреса:

Адрес в форме <Механизм обмена сообщениями>:<Адрес для отправки сообщения>

<Механизм обмена сообщениями> может быть:

- AM: использование внутренней системы обмена сообщениями Asset Manager.
- MAPI: использование системы обмена сообщениями по стандарту MAPI (Internet Mail, Microsoft Outlook и так далее).
- VIM: использование системы обмена сообщениями по стандарту VIM (Lotus Notes и так далее).
- SMTP: использование системы обмена сообщениями по стандарту SMTP (стандарт Интернета).

Компонент <Адрес для отправки сообщения> имеет обычную форму, соответствующую выбранной системе обмена сообщениями. Адреса внутренней системы обмена сообщениями совпадают с именами входа пользователей.

Примеры адресов:

- AM:Admin
- MAPI:OlegPutin@janus.ru
- VIM: Poletaev/JANUS
- SMTP:cbernard@taltek.com

Адрес в форме <Asset Manager имя входа>

В таком случае используемой системой обмена сообщениями будет та, которая указана в поле Электронная почта (имя SQL: EMail) вкладки Общее сведений о сотруднике, чье Имя входа (вкладка Профиль в сведениях о сотруднике) указано в адресе.

Если поле Электронная почта не заполнено, сообщение пересылается через внутреннюю систему обмена сообщениями.

Например:

- 1 Сообщение отправляется для следующих имен входа Asset Manager: Cathy, Gerald и Philip.
- 2 Поля Электронная почта показывают MAPI:CathyBernard@taltek.com для Cathy и VIM:Gerald Colombo/Taltek для Gerald. Поле Электронная почта для Philip пусто.
- 3 Если у отправителя - учетная запись MAPI, сообщение будет отправлено Cathy через интерфейс MAPI, а двум остальным получателям - через внутреннюю систему обмена сообщениями Asset Manager.
- 4 Если у отправителя - учетная запись VIM, сообщение будет отправлено Cathy через VIM, а двум остальным получателям - через внутреннюю систему обмена сообщениями Asset Manager.

Адрес с контекстными переменными

Если действие является контекстным, можно использовать переменные в квадратных скобках []. Данными переменными вызываются значения полей в базе данных Asset Manager.

Например: чтобы отправить сообщение пользователю актива, выбранного в таблице активов, можно использовать выражение [User.Email].

Примеры действий

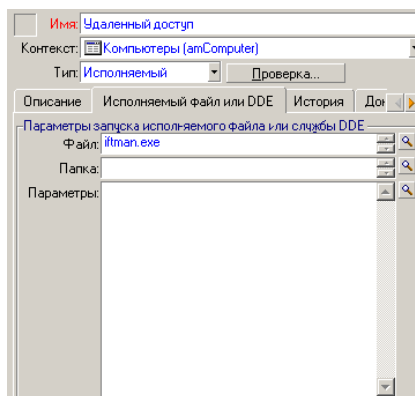
В данном разделе приведены следующие примеры действий Asset Manager:

- пример действия исполняемого типа;
- пример действия типа DDE;
- пример действия типа «Обмен сообщениями»;
- пример действия типа «Скрипт».

Пример действия исполняемого типа

На следующем экране описывается неконтекстное действие, которым запускается Asset Manager Automated Process Manager и выполняется подключение к базе данных amdemo:

Рисунок 7.1. Действие исполняемого типа - окно сведений



Пример действия типа DDE

Действия типа DDE применяются во многих случаях:

- вставка данных Asset Manager в таблицу Microsoft Excel;
- вставка информации, связанной с заказом на поставку, в пакет программного обеспечения для учета;
- автоматическая отправка подтверждающего сообщения по факсу при закрытии квитанции;
- автоматическая отправка запроса заказа на работу;
- и т.п.

Пример действия типа «Обмен сообщениями»

Отправляется сообщение из списка активов, чтобы указать истечение срока договора аренды актива, пользователю данного актива. Данный актив должен иметь указание на аренду как на метод приобретения и должен быть связан с договором аренды (вкладка **Приобр.**). Чтобы ссылочный объект был заявкой на поставку, а не строкой заказа, сведения о действии необходимо сконфигурировать следующим образом:

Рисунок 7.2. Действие типа «Обмен сообщениями» с ссылочным объектом - окно сведений

Имя: Expiration

Контекст: Ресурсы (amAsset)

Тип: Обмен сообщениями Проверка...

Описание Обмен сообщениями История Документы

Ссылочный объект:

Приоритет: Обычный

☐ Подтверждение

По: [Позиции портфеля, Пользователь, Email]

Копия:

Скрытая копия:

Тема: Истечение контракта

Сообщение

The [AcquContract.cf_sell] expires on [AcquContract.dEnd]

Пример действия типа «Скрипт»

Создание действия типа Скрипт заключается в основном в написании скрипта на языке Basic, которым изменяется база данных Asset Manager.



ЗАМЕЧАНИЕ:

Использование функций, соответствующих данным действиям, разрешается в данных скриптах. Такие функции указаны в руководстве «Programmer's Reference», глава «Index of functions by field of application», раздел «Built-in functions».

Предисловие

Для подготовки к созданию действия выполните следующие шаги:

- 1 Выберите пункт меню Инструменты/Действия/Изменить и нажмите кнопку **Создать** для создания нового экрана со сведениями о действии.
- 2 Назначьте имя действия, которое необходимо создать, например, Test, и установите для поля Тип (имя SQL: seActionType) значение Скрипт. Откажитесь от выбора контекста для данного действия. Нажмите кнопку **Создать**.
- 3 На вкладке Скрипт щелкните кнопку , чтобы отобразить окно составителя скриптов. Для программируемой функции **Success()**, используемой для

таких действий, не нужен явный код возврата. В приведенном далее примере будет создана новая запись в таблице «Виды» на основе информации, содержащейся в таблице ниже.

Метка поля	Имя SQL поля	Значение поля
Имя	Имя	Mini-computer
Создано	seBasis	Позиция портфеля
Может соединяться	blsCnxClient	Установлен данный флажок

Написание скрипта

Введите следующий пример

```
Dim lrec As Long Dim lres As Long lrec=AmCreateRecord("amNature") lres=AmSetFieldStrValue(lrec, "Name", "Mini-computer") lres=AmSetFieldStrValue(lrec, "seBasis", 1) lres=AmSetFieldStrValue(lrec, "blsCnxClient", 1) AmInsertRecord(lres)
```



ЗАМЕЧАНИЕ:

Данным действием создается необходимый вид без всякого вмешательства пользователя.

Демонстрация применения функции Set()

Теперь тот же самый вид будет создан с помощью действия Скрипт с указанием таблицы «Виды» как контекста действия. В этом случае скрипт записывается следующим образом:

```
Set [Name]="Mini-computer" Set [seBasis]=1 Set [blsCnxClient]=1
```



ЗАМЕЧАНИЕ:

Чтобы выполнить данное действие, пользователь должен открыть таблицу «Виды» и щелкнуть **Новый**. После выполнения скрипта пользователь должен также щелкнуть **Создать**, чтобы подтвердить создание.

Совет

Если необходимо сделать недействительным выполнение действия в скрипте, достаточно просто установить для кода возврата значение, отличное от 0 (например, 12001). Данное значение рассматривается как код ошибки. Следующей командой прерывается действие и отменяются все уже сделанные изменения:

Использование переменных

На вкладках Исполняемое, DDE или Обмен сообщениями сведений о контекстном действии можно использовать переменные, которые ссылаются на содержимое полей, признаки или вычисляемые поля в базе данных.

Ими используется форма [Ссылка.Ссылка.Поле].

Для упрощения ввода таких переменных щелкните значок увеличения  справа от заполняемого поля.

Все, что не заключено в квадратные скобки [], считается текстом.

Например: выражением [Link.Link.Field].doc вызывается значение поля Поле в таблице, связанной с основной таблицей через ссылки Link.Link.



ВНИМАНИЕ:

Чтобы данные переменные действовали, полем Контекст сведений о действии должна показываться таблица в Asset Manager и перед выполнением действия необходимо выбрать запись в списке записей таблицы.

Тестирование действия

Для тестирования действия при его создании используется кнопка **Тест** в верхнем правом углу сведений о тестируемом действии.

Asset Manager отображает окно предварительного просмотра, в котором можно выбрать контекст для предварительного просмотра действия. Данное окно содержит две кнопки: **Вычислить** и **Выполнить**.

Кнопка **Вычислить**

После выбора контекста щелкните кнопку **Вычислить**. При этом заполняются поля на вкладках Исполняемое, DDE или Обмен сообщениями. Проверьте правильность извлечения переменных из записи, выбранной в поле Контекст (имя SQL: ContextTable).

Кнопка **Выполнить**

Данная кнопка позволяет выполнить действие непосредственно из экрана.

Выполнение действия

Можно выполнить действие одним из нескольких способов.

- С помощью раскрывающегося списка  на панели инструментов.
 - Кнопка  заменяется значком, связанным с последним действием, использованным на данной рабочей станции, если такой значок существует. Если действие уже выполнено, необходимо щелкнуть значок  или значок, его заменяющий, чтобы запустить действие снова.
 - С помощью кнопки  отображается список доступных действий.
 - Чтобы вставить такой всплывающий список в панель инструментов, используется элемент меню «Инструменты/Настройка панели инструментов»: он представляет собой часть категории «Инструменты».
- С помощью пункта меню Инструменты/Действия: щелкнуть необходимое действие в подменю.
- С помощью кнопки **Тест** в верхнем правом углу сведений о действии.
 - Если действие контекстное, необходимо указать контекст, (имя SQL: ContextTable), выбрав запись в ссылочной таблице действий.
 - Щелкнуть кнопку **Выполнить**, чтобы выполнить действие.
- С помощью контекстного меню (доступного с помощью щелчка правой кнопкой мыши). Если для открытой таблицы доступно, по крайней мере, одно действие, в контекстном меню показывается запись «Действия».

Множественный выбор в списках

Можно выбрать в списке несколько записей и применить к ним действие.

Таким способом можно выбрать несколько активов и послать их пользователям одно и то же сообщение.

Действие типа «Мастер»

Мастер состоит из последовательности страниц. На каждой такой странице отображается информация или требуется ввод пользователя, например, необходимо сделать выбор или ввести элементы данных.

Навигация между различными страницами мастера проста:

- После заполнения страницы так, как необходимо, можно перейти к следующей странице (определяемой переходом), щелкнув кнопку **Далее**. Данная кнопка на последней странице недоступна.
 - Всегда можно вернуться для внесения любых исправлений, нажав кнопку **Назад**.
 - Можно выполнить заключительное действие мастера в любой момент, нажав кнопку **Готово**. Если для мастера не хватает информации, чтобы выполнить назначенную задачу, отображается соответствующая страница.
- Можно полностью отменить выполнение мастера (и, как следствие, связанное с ним действие), щелкнув кнопку **Отмена**.

Действия исполняемого типа

Поведение действий исполняемого типа зависит от значения поля **Файл** (ActionFile) сведений о действии.

В таблице ниже собраны описания возможного поведения клиента Windows и веб-клиента в зависимости от ситуации:

Если поле **Файл** (ActionFile) в сведениях о действии начинается с http или https

Если действие инициировано клиентом Windows	Если действие инициировано веб-клиентом
Запустите на рабочей станции клиента Windows обозреватель Интернета по умолчанию и подключитесь по адресу URL, указанному в поле Файл (ActionFile).	Отобразите страницу, соответствующую адресу URL, указанному в поле Файл (ActionFile) рабочей области веб-клиента.

Если поле **Файл** (ActionFile) в сведениях о действии начинается с ftp

Если действие инициировано клиентом Windows	Если действие инициировано веб-клиентом
Запустите на рабочей станции клиента Windows Проводник Windows и подключитесь к узлу ftp, указанному в поле Файл (ActionFile).	Отобразите узел ftp, соответствующий адресу URL, указанному в поле Файл (ActionFile) рабочей области веб-клиента.

Если поле **Файл** (ActionFile) в сведениях о действии начинается с mailto

Запустите приложение обмена сообщениями по умолчанию на рабочей станции клиента Windows или веб-клиента и создайте новое сообщение с помощью параметров, указанных после mailto: в поле **Файл** (ActionFile).

Если поле **Файл (ActionFile)** в сведениях о действии начинается с другого значения

Если действие инициировано клиентом Windows	Если действие инициировано веб-клиентом
Исполнение файла, указанного в поле Файл (ActionFile)	По умолчанию, файл не выполняется ни на станции Asset Manager Web Service, ни на веб-клиенте и возвращается ошибка. Чтобы обеспечить исполнение файла на станции Asset Manager Web Service: 1 Отобразите параметры базы данных (меню Администрирование/Параметры базы данных... на клиенте Windows). 2 Выберите параметр Действия/Выполнение действий исполняемого типа (ExecuteAction). 3 Установите для этого параметра значение Сервер. ВНИМАНИЕ: Операции такого типа заключают в себе некоторый риск.

Связывание действия с кнопкой на экране

Существует несколько способов связывания действия с кнопкой на экране. Дополнительные сведения см.:

- Руководство Tailoring, часть 1 Customizing client computers, глава Customizing a client workstation, раздел Customizing buttons.
- Руководство Tailoring, глава Customizing the database, раздел Customizing existing objects/ Customizing objects / Customizing a detail/ Creating action buttons.
- Руководство Tailoring, глава Customizing the database, раздел Creating new objects/ Creating action buttons.

8 Обмен сообщениями

Asset Manager дает возможность управлять двумя типами сообщений:

- Сообщения, исходящие из Asset Manager и направляемые в базу данных Asset Manager через внутреннюю систему обмена сообщениями.
- Сообщения, созданные в Asset Manager и отправленные через внешнюю систему обмена сообщениями.

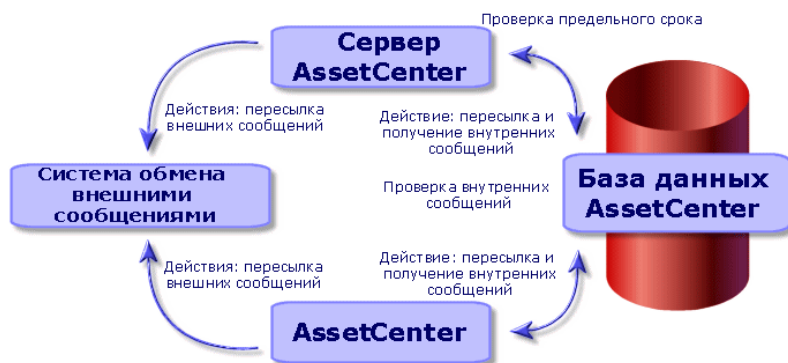
Обзор обмена сообщениями

Asset Manager управляет отправкой сообщений с помощью следующих типов протоколов:

- AM (Asset Manager)
- SMTP
- MAPI
- VIM

При получении сообщений Asset Manager управляет только теми сообщениями, которые используют протокол AM (Asset Manager).

Рисунок 8.1. Обмен сообщениями - обзор



Настройка системы обмена сообщениями более подробно рассматривается в руководстве Администрирование, глава Обмен сообщениями.

Отправка сообщений

Сообщения создаются с помощью действий Обмен сообщениями.

Чтобы можно было отправлять сообщения, сначала необходимо создать данные действия.

Действие инициируется различными способами:

- вручную, выбором действия в списке, предоставляемым пунктом меню **Инструменты/Действия**;
- Автоматически, посредством Asset Manager Automated Process Manager.
- автоматически, с помощью Asset Manager.

Создание действий типа «Обмен сообщениями» описывается в разделе [Заполнение вкладки «Обмен сообщениями»](#) [стр. 139].

Обращение к сообщениям

Обращение к сообщениям, отправленным во внутреннюю систему обмена сообщениями

Агент проверяет прибытие новых внутренних сообщений и информирует пользователей о наличии новых сообщений.

К данным сообщениям можно обращаться:

- с помощью пункта меню **Инструменты/Сообщения**;

- в окне сообщений, которым информируется о наличии новых сообщений.

Обращение к сообщениям, отправленным во внешнюю систему обмена сообщениями

Получатель таких сообщений может обращаться к ним обычным образом, из клиента внешней системы обмена сообщениями.

Подтверждение получения

ЗАМЕЧАНИЕ:

Нельзя получить подтверждение на сообщение, посланное через внутреннюю систему обмена сообщениями Asset Manager или через систему обмена сообщениями MAPI или SMTP.

Чтобы получить подтверждение, необходимо установить флажок

Подтверждение (имя SQL: bAcknowledgment) в сведениях о действии типа Обмен сообщениями.

Это подтверждение будет отослано по адресу, указанному в поле

Электронная почта (имя SQL: EMail) во вкладке **Общее** и относящемуся к сотруднику, открывшему базу данных Asset Manager (в таблице «Сотрудники и подразделения»).

9 Рабочий процесс

Предназначение рабочего процесса - организация бизнес-процессов и их внедрение. По определению организации WfMC (Workflow Management Coalition (Коалиция по управлению рабочими процессами) - некоммерческая организация, занимающаяся определением, разработкой и внедрением стандартов для решений рабочих процессов), процесс представляет собой «набор одной или нескольких связанных процедур или действий для совместной реализации какой-либо бизнес-цели или политической цели». Управление такими процессами включает моделирование рабочих процедур и учет всех элементов, связанных с работой организации.

Система рабочих процессов в Asset Manager:

- позволяет формализовать бизнес-процессы с помощью схем рабочих процессов: кем, что, когда и как должно выполняться;
- поддерживает выполнение экземпляров данных процессов в соответствии с формальным определением;
- обеспечивает доступ к информации и инструментам, необходимым для выполнения задач, определенных в действиях и связанных с процессами. Задача может выполняться автоматически или требовать внимания пользователя;
- предоставляет метрики для выполненных и еще выполняющихся экземпляров.

Например, с помощью методов рабочих процессов можно моделировать и автоматизировать следующие процессы:

- процедуры утверждения заявок на поставку;
- перемещения активов;

- и т.п.

При создании схемы рабочего процесса в Asset Manager определяется следующее:

- начальная операция, представляющая собой начальную точку процесса;
- события (содержащиеся в операции);
- переходы, создаваемые событиями;
- операции, инициируемые переходами;
- контекст;
- лимиты времени и оповещения.

Рисунок 9.1. Рабочий процесс - упрощенная схема



Определения

В данном разделе даются определения нескольких ключевых терминов, используемых в рабочем процессе:

- определение экземпляра рабочего процесса;
- определение операции рабочего процесса;
- определение события рабочего процесса;
- определение перехода рабочего процесса;
- задача рабочего процесса;
- определение назначенного для операции рабочего процесса;
- определение группы рабочих процессов.

Определение экземпляра рабочего процесса

Экземпляр рабочего процесса представляет собой рабочий процесс, исполняемый в соответствии с формальным процессом, определенным в схеме рабочего процесса.

Операция рабочего процесса

В состав рабочего процесса входят:

- задача, которую необходимо выполнить. Для выполнения задачи может понадобиться взаимодействие с пользователем, или она выполняется автоматически Сервером Asset Manager;
- События, которые инициируют переходы к другим операциям.

Событие рабочего процесса

События рабочих процессов являются результатами операций. Они, в свою очередь, позволяют активировать переходы, которыми инициируются другие операции.

Переход рабочего процесса

Переход рабочего процесса позволяет идти от одной операции к другой. Они инициируются событием.

Событие может связываться с несколькими переходами.

Задача рабочего процесса

Задача рабочего процесса представляет собой назначенную для выполнения задачу, явившуюся результатом инициирования операции.

Назначенный операции рабочего процесса

Назначенные операции - пользователи, ответственные за выполнение задач, явившихся результатом операций рабочих процессов.

Определение группы рабочих процессов

Группы выполнения рабочих процессов позволяют распределять по категориям определяемые схемы рабочих процессов. Группа выполнения, к которой относится данная схема рабочего процесса, указывается в сведениях о схеме рабочего процесса.

Общий обзор

Ключевым этапом внедрения рабочего процесса в Asset Manager является определение схем рабочих процессов с помощью графического редактора через пункт меню **Инструменты/Рабочие процессы/Схемы рабочего процесса**. Схемами рабочих процессов определяются операции, события и переходы. Они могут ссылаться на операции и сотрудников (назначенных рабочих процессов) Asset Manager.



ВНИМАНИЕ:

Схемы рабочего процесса можно редактировать только при помощи Windows-клиента Asset Manager.

Схемы рабочих процессов интерпретируются механизмами рабочих процессов. Рассматриваемые механизмы рабочих процессов выполняются или Сервером Asset Manager, или агентами Asset Manager.

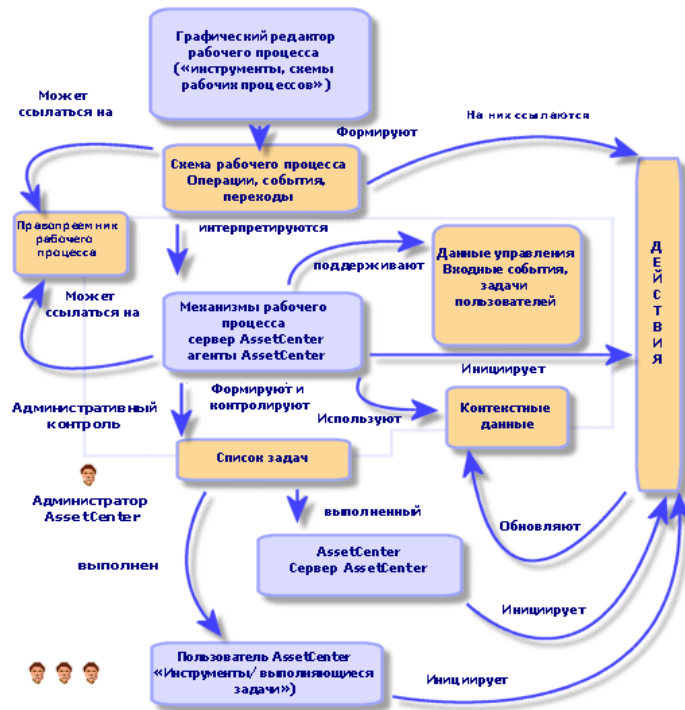
Реагируя на события, механизмы рабочих потоков инициируют и отслеживают экземпляры рабочих процессов:

- механизмы рабочего процесса создают задачи для выполнения;
- ими отслеживаются такие задачи и события, инициирующие операции;
- ими также сохраняется запись о ходе выполнения событий, с регистрацией входных событий и задач пользователей для выполнения.

Задачи рабочих процессов выполняются или механизмами рабочих процессов, или пользователями Asset Manager. По мере своего выполнения ими активируются события, которые затем учитываются механизмами рабочих процессов.

На следующей диаграмме дается обзор того, как реализуется рабочий процесс в Asset Manager:

Рисунок 9.2. Рабочий процесс в Asset Manager - обзор



Обработка экземпляров рабочих процессов изменяется в соответствии с видом операций и событий и тем, как настроены управляющие ими механизмы рабочих процессов.

Реализация рабочего процесса

Рабочий процесс в Asset Manager позволяет реализовывать процедуры компании. Ниже приводятся этапы выполнения:

- 1 Проанализируйте процедуры компании, которые необходимо формализовать.
- 2 Создайте:
 - 1 Роли в рабочем процессе.
 - 2 Действия.

- 3 Создайте схемы рабочего процесса, для которых определите:
 - 1 Операции, события и переходы.
 - 2 Оповещения.
- 4 Определите соответствующие группы выполнения рабочих процессов. Свяжите каждую схему рабочего процесса с группой выполнения рабочего процесса.
- 5 Запустите Asset Manager Automated Process Manager на одном или нескольких компьютерах. Для каждого экземпляра Asset Manager Automated Process Manager определите группы выполнения рабочих процессов, которые необходимо отслеживать, и параметры отслеживания.

Использование графического редактора рабочего процесса

Для доступа к схемам рабочих процессов используется пункт меню Инструменты/Рабочие процессы/Схемы рабочего процесса. Данный пункт меню зарезервирован для администраторов Asset Manager.



ВНИМАНИЕ:

Редактор рабочих процессов доступен только в Windows-клиенте Asset Manager.

Вкладка Операции в сведениях о схеме рабочего процесса включает графический редактор, который позволяет упростить создание схем рабочих процессов.

В данном разделе описывается, как использовать данный графический редактор для создания, изменения или удаления схем рабочих процессов:

- Операции
- События
- Переходы
- Прочая функциональность

Операции

Чтобы создать операцию:

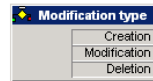
- щелкните правой кнопкой мыши по пустой области вкладки Операции, затем выберите в контекстном меню команду Добавить операцию. Отображаются сведения об операции;
- или в качестве альтернативы щелкните кнопку , затем щелкните в графической зоне. Отображаются сведения об операции.

Чтобы удалить операцию:

- или выберите операцию, щелкнув ее, а затем нажмите клавишу Delete;
- или выберите операцию и выберите команду Удалить в контекстном меню;
- или выберите операцию и щелкните кнопку .

Чтобы изменить сведения об операции:

- щелкните правой кнопкой мыши операцию и выберите пункт Сведения об операции в контекстном меню;
- или дважды щелкните операцию.



Щелкните здесь, чтобы
изменить сведения о событии

Щелкните здесь, чтобы
изменить сведения об операции

События

События, содержащиеся в операциях, бывают двух типов.

Событие типа «База данных» или «Периодическое»

Чтобы добавить исходящее событие к операции:

- щелкните правой кнопкой мыши операцию, затем выберите команду **Добавить событие** в контекстном меню;
- или выберите операцию, затем щелкните кнопку .

Чтобы удалить из операции исходящее событие типа База данных или Периодическое:

- выберите событие, затем нажмите клавишу Delete;
- или дважды щелкните событие и выберите команду Удалить в контекстном меню;
- или выберите событие и щелкните кнопку .

Чтобы изменить сведения о событии типа База данных или Периодическое:

- дважды щелкните событие;
- или выберите событие, затем выберите пункт Сведения о событии в контекстном меню.

Событие системы

Есть два различных типа событий в приложениях, события системы и события пользователей. События системы - события, инициируемые приложением как

результат операции. События пользователя инициируются пользователем, например, как результат события типа «вопрос».

Тип события определяется в поле **seType** (поле не показывается).

Переходы

Для создания перехода:

- 1 Щелкните начальное событие, чтобы его выбрать.
- 2 Удерживая кнопку мыши нажатой, перетащите его в целевую операцию.

Для удаления перехода:

- выберите переход, щелкнув его, затем нажмите клавишу Delete;
- или выберите переход и выберите команду Удалить в контекстном меню;
- или выберите переход и щелкните кнопку .

Чтобы изменить сведения о переходе:

- 1 Щелкните переход, чтобы его выбрать.
- 2 Выберите пункт Сведения о переходе в контекстном меню.

Чтобы изменить источник и (или) цель перехода:

- 1 Выберите переход.
- 2 Перетащите объект, который необходимо изменить.

Прочая функциональность

Графический редактор позволяет также:

- перетаскивать связанные операции и переходы;
- увеличивать или уменьшать схемы с помощью ползунка Масштаб или кнопки .

Пример рабочего процесса, используемого при утверждении заявки

В данном разделе описывается простой пример рабочего процесса.

- 1 [Предназначение](#) [стр. 161]
- 2 [Предпосылки](#) [стр. 162]
- 3 [Создание операций](#) [стр. 168]
- 4 [Настройка событий, создаваемых одновременно с операциями](#) [стр. 172]
- 5 [Создание начального события](#) [стр. 173]
- 6 [Создание переходов](#) [стр. 174]

Предназначение

Предназначение данной схемы рабочего процесса - автоматизация обработки заявки на поставку в соответствии со следующим:

Рисунок 9.3. Рабочий процесс - проверка заявки



Схема рабочего процесса состоит из следующих шагов:

- 1 Экземпляр рабочего процесса запускается, как только возникает необходимость проверки заявки на поставку (1), то есть когда поле **Статус заявки** (имя SQL: seStatus) в сведениях о заявке на поставку приобретает значение Ожидается утверждение.
- 2 Сначала заявка проходит функциональную проверку (2). Данный шаг заключается в представлении заявки ответственному подразделению на утверждение. Он информируется сообщением. Программируется оповещение с напоминанием, инициируемое, если заявка не рассмотрена утверждающим до конца следующего рабочего дня после отправки сообщения с запросом утверждения.
- 3 Если заявка на поставку проходит проверку ответственным лицом (3), следующий шаг - финансовая проверка.
В противном случае заявка отклоняется (3b).
- 4 Финансовая проверка (4) заключается в подаче заявки финансовому контролеру компании Gerald Colombo. Он также извещается письмом и программируется оповещение с напоминанием об утверждении.
- 5 Если заявка на поставку проходит проверку финансовым контролером, заявка на поставку является утвержденной (6).
В противном случае заявка отклоняется (5b).
- 6 Если заявка на поставку принимается, Asset Manager устанавливает для поля **Статус заявки** в сведениях о заявке на поставку значение Проверена.

Если заявка на поставку отклоняется (6b), Asset Manager устанавливает для поля **Статус заявки** в сведениях о заявке на поставку значение Отклонена.

Предпосылки

Необходимо подключиться к базе данных с именем входа **Admin** и настроить систему обмена сообщениями (► [Обмен сообщениями](#) [стр. 149]).

Создание назначенных рабочих процессов

В данной схеме рабочего процесса участвуют следующие назначенные:

- ответственный за подразделение запрашивающего;
- финансовый контролер компании Gerald Colombo.

Назначенные рабочих процессов - сотрудники, определенные в таблице «Сотрудники и подразделения», которые уполномочены взаимодействовать на одном или нескольких этапах рабочего процесса.

Назначенные рабочих процессов создаются и сохраняются в таблице ролей в рабочем процессе (имя SQL: amWfOrgRole). Чтобы их создать, выберите пункт меню **Инструменты/Рабочие процессы/Роли**, затем щелкните **Создать**.

Ответственный за подразделение запрашивающего

Данное лицо рассчитывается скриптом. Для его определения заполните экран сведений, как показано ниже.

Роли в рабочем процессе: Сведения о роли «Ответственный подра...

Назначение	Контекст	Ссылка
Администратор учетной записи	amCntrlEmpl	D000000

14 / 14

Назначение: , подающий заявку Ссылка: D001001

Контекст: Заявки (amReque) Тип: Рассчитанное лицо

Скрипт
RetVal = Requester.Parent.Supervisor

Создать
Создать +
Отмена

Финансовый контролер

Данное лицо назначается как Gerald Colombo. Чтобы определить его как назначенного, заполните экран сведений, как показано ниже.

The screenshot shows a Windows-style dialog box titled "Роли в рабочем процессе: Сведения о роли «Финансовый контролер»". It contains the following fields and controls:

- Назначение:** A text field containing "Финансовый контролер".
- Ссылка:** A text field containing "D001001".
- Контекст:** A dropdown menu with the selected option "(Таблица отсутствует)".
- Тип:** A dropdown menu with the selected option "Назначенное лицо".
- Назначенный:** A text field containing "Полетаев, Георгий".
- Buttons:** "Создать" (top right), "Дублировать" (middle right), and "Закрыть" (bottom right).

Создание действий

Схемами рабочих процессов вызываются многочисленные действия. Чтобы их создать, выберите пункт меню Инструменты/Действия/Изменить.

Отправка заявки на техническую проверку ответственному за подразделение запрашивающего

Данное действие используется в фазе технической проверки. Оно позволяет уведомить лицо, ответственное за техническую проверку, о необходимости рассмотреть данную заявку:

The screenshot shows a software window titled "Действия: Сведения о действии «Отправить напоминание о технической проверке»". The window contains the following fields and controls:

- Имя:** Отправить напоминание о технической проверке
- Контекст:** Задачи рабочего процесса (an/wf/workItem)
- Тип:** Обмен сообщениями (with a "Проверка..." button next to it)
- Таблица:** Three tabs are visible: "Описание", "Обмен сообщениями", and "История".
- Ссылочный объект:** (empty dropdown)
- Приоритет:** Обычный (dropdown menu)
- Подтверждение:** (checkbox, currently unchecked)
- По:** [Assignee.EMail] (with a search icon)
- Копия:** (empty field with a search icon)
- Скрытая копия:** (empty field with a search icon)
- Тема:** Утверждение заявки (with a search icon)
- Сообщение:** A text area containing the text "Необходимо отклонить или утвердить заказ на поставку". To the right of the text area are three icons: a magnifying glass, a checkmark, and a red X.
- Buttons:** "Изменение" and "Отмена" are located at the bottom right of the window.

Отправка заявки финансовому контролеру для финансовой проверки

Данное действие используется в процессе финансового утверждения. Оно отправляет сообщение лицу, ответственному за финансовую проверку, о необходимости рассмотреть данную заявку:

The screenshot shows a software window titled "Detail of action 'Send a financial-validation reminder'". It contains several fields and buttons for configuring an action. At the top, there is a "Name" field with the text "Send a financial-validation reminder", a "Context" dropdown menu set to "Workflow tasks (amw/fw/workItem)", and a "Type" dropdown menu set to "Messaging". To the right of these fields are "New" and "Duplicate" buttons. Below these is a "Preview..." button. A tabbed interface is visible with tabs for "Description", "Messaging", "History", and "Features". The "Messaging" tab is active, showing fields for "Referenced object", "Priority" (set to "Normal"), and an "Acknowledgment" checkbox. Below these are email fields: "To:" (containing "[Assignee.EMail]"), "Cc:", "Bcc:", and "Subject:" (containing "Request approval"). Each of these fields has a magnifying glass icon to its right. At the bottom is a "Message:" text area containing the text "Do you approve the request [ReqNumber]?", also with a magnifying glass icon. A "Close" button is located at the bottom right of the window.

Проверка заявки на поставку

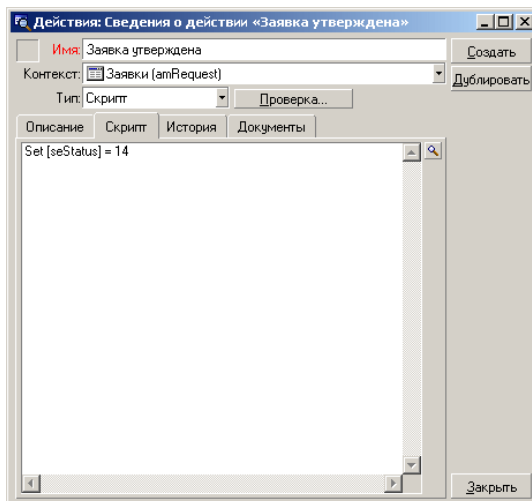
Данное действие используется на уровне операции **Заявка утверждена**, которая будет создана позже.

Им для поля **Статус заявки** (имя SQL: seStatus) в сведениях о заявке на поставку устанавливается значение Проверено. Данное действие является действием типа Скрипт.

Поле **Статус заявки** представляет собой системный подробный список. Для просмотра списка его значений отобразите справку по данному полю:

- 1 Щелкните правой кнопкой мыши поле **Статус заявки** в подробных сведениях о заявке.
- 2 Выберите пункт Справка по этому полю в контекстном меню: значение, отображаемое как Проверена, хранится в базе данных как **14**.

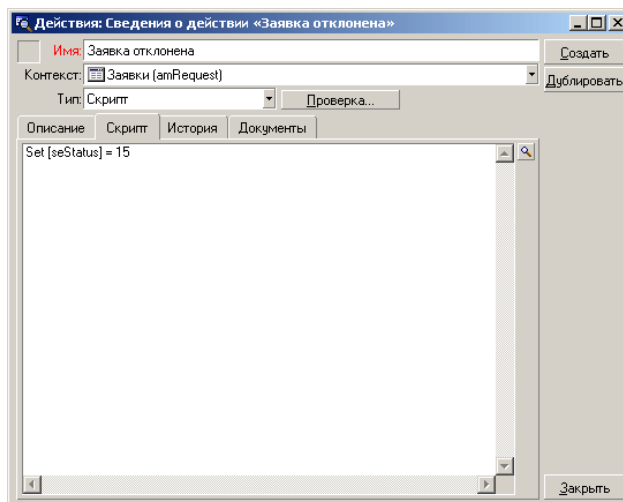
Выполняется следующее действие:



Отклонение заявки на поставку

Данное действие используется на уровне операции **Заявка отклонена**, которая будет создана позже.

Оно подобно действию **Заявка утверждена**, но для поля **Статус заявки** (имя SQL: seStatus) на вкладке **Общее** сведений о запросе должно быть установлено значение Отклонена.



Создание календаря

Для доступа к списку календарей используется пункт меню

Инструменты/Календари. Такие календари связываются с операциями схем рабочих процессов. Они позволяют устанавливать сроки для экземпляров рабочих процессов:

	0	4	8	12	16	20	24
Понедельник:							8:00-12:00;14:00-18:00
Вторник:							8:00-12:00;14:00-18:00
Среда:							8:00-12:00;14:00-18:00
Четверг:							8:00-12:00;14:00-18:00
Пятница:							8:00-12:00;14:00-18:00
Суббота:							
Воскресенье:							

Подготовка схемы рабочего процесса

- 1 Щелкните пункт меню **Инструменты/Рабочие процессы/Схемы рабочего процесса**.
- 2 Нажмите кнопку **Создать**.
- 3 Назначьте рабочему процессу имя **Утверждение заявки**.
- 4 Укажите контекст начального объекта, который будет применяться по умолчанию ко всем операциям в схеме рабочего процесса. В данном случае это таблица заявок (имя SQL: amRequest).
- 5 Щелкните **Создать**: начальная операция (**Запуск**) автоматически создается Asset Manager в графическом редакторе во вкладке Операции.

Имя: Утверждение заявки
Ссылка: W001001
Контекст объекта запуска: Заявки (amRequest)

Операции | Общее | История | Признаки | Документы | Рабочий процесс

Масшт.: [Slider]

Начало

Создание операций

Операции можно создавать графически на вкладке **Операции** сведений о рабочем процессе:

- 1 Щелкните за пределами объекта рабочего процесса.
- 2 Щелкните правой кнопкой мыши.
- 3 Выберите элемент меню Добавить операцию в контекстном меню: отображаются сведения об операции.

Создание операции **Техническая проверка**

- 1 Назначьте имя операции **Техническая проверка**.
- 2 Так как операция представляет собой подачу заявки на утверждение ответственному подразделению, выберите значение Вопрос в подробном списке Тип (имя SQL: seType).
- 3 Поле **Контекст** (имя SQL: ContextTable) на вкладке Общее не изменяется.

Настройка операции **Техническая проверка**

- 1 Заполните вкладку Параметры, как показано ниже:

Операция в схеме рабочего процесса «Утверждение заявки»

Имя: Техническая проверка Шаблон:

Тип: Вопрос Ссылка: A001001

Общие Параметры История Документы

Назначенный:

Назначенный: Менеджер подразделения, создающего заявку

☒ Уведомление лица

Действие: Заявка для технического утверждения

Вопрос или инструкции:

Вы утверждаете или отклоняете заявку [ReqNumber]

Возможные ответы:

Назначение: Утвердить

Имя SQL: Approve

Утвердить Отклонить

Добавить

Добавить >

Отмена

- 2 Укажите задаваемый вопрос:
 - 1 Текст вопроса ссылается на номер заявки на поставку.
 - 2 Возможны два ответа: отклонить или утвердить. Чтобы добавить подзакладку, описывающую ответ на вопрос, щелкните правой кнопкой мыши область метки подзакладок и выберите Добавить связанную запись или Копировать ссылку.

- 3 Укажите, кому адресован запрос, в поле Назначено (имя SQL: Assignee). В данном случае назначенным является менеджер запрашивающего. Данное лицо было создано в таблице ролей в рабочем процессе в предыдущих шагах.
- 4 Чтобы назначенный был уведомлен о необходимости рассмотреть заявку:
 - 1 Установите для поля Уведомить лицо (имя SQL: bNotifAssignee) значение Да.
 - 2 Укажите выполняемое действие: это действие **Заявка для технического утверждения**, созданное на промежуточном шаге. Данное действие автоматически инициируется при подаче заявки на поставку для технического утверждения.

Определение лимита времени для выполнения операции **Техническая проверка**

На вкладке Лимит времени сведений об операции:

- 1 Укажите календарь рабочих дней, прикрепленный к операции. Данный календарь учитывается при расчете лимитов времени. Выберите календарь **Milwaukee calendar**, созданный на предыдущем шаге.
- 2 Определите время, по прошествии которого должно приниматься решение относительно момента времени для инициации операции. В данном случае назначенный должен ответить на вопрос в течении 24 часов.

The screenshot shows a software window titled "Операция в схеме рабочего процесса «Утверждение заявки»". It contains several tabs: "Общее", "Параметры", "Лимит времени", "Оповещения", and "История". The "Лимит времени" (Time Limit) tab is currently selected. Within this tab, there is a section for "Календарь" (Calendar) with a dropdown menu showing "Стандартный календарь - Германия". Below this, there is a "Время разрешения" (Resolution time) section with a "Тип задержки" (Delay type) dropdown set to "Продолжительно" (Continuous) and a "Лимит времени" (Time limit) input field containing the value "24ч". At the bottom right of the window, there are two buttons: "Изменение" (Change) and "Отмена" (Cancel).

Определение оповещения для операции **Техническая проверка**

На вкладке Оповещения сведений об операции определяется оповещение с напоминанием в случае, если решение не принято до конца лимита времени, указанного на вкладке Лимит времени.

Проще говоря, оповещением будет инициироваться действие **Заявка для технического утверждения**:

Операция в схеме рабочего процесса «Утверждение заявки»

Имя: Техническая проверка Шаблон: [выбор]

Тип: Вопрос Ссылка: A001001

Общие | Параметры | Лимит времени | Оповещения | История

Имя: Заявка для технического утверждения

Ссылка: ALRM001001

Инициирование действия

Тип: Время, прошедшее с начала задачи

Лимит времени: 0ч

Активация: Действие

Действие: Заявка для технического утверждения

Заявка для технического утверждения

Изменение Отмена

Можно определить дополнительные оповещения с помощью команды Добавить связанную запись в контекстном меню.

События

После создания операции Asset Manager создает два события системы, Утверждение и Отказ, соответствующие двум возможным ответам на вопрос:



Когда происходят данные события, агент Asset Manager регистрирует их в таблице элементарных событий рабочих процессов (имя SQL: amWfOccurEvent).

Ниже указаны операции, инициируемые или Asset Manager непосредственно, или Asset Manager Automated Process Manager в соответствии с конфигурацией событий рабочих процессов:

- Если в поле Обработка (имя SQL: seProcessingMode) установлены значения **Записать событие в журнал и обработать немедленно** или **Обработать событие немедленно без записи в журнал**, Asset Manager инициирует следующее действие.
- Если в поле Обработка (имя SQL: seProcessingMode) установлено значение **Зарегистрировать событие и обработать сервером**, Asset Manager Automated Process Manager инициирует следующее действие.

По умолчанию, для поля Обработка (имя SQL: seProcessingMode) на вкладке Общее сведений о событии устанавливается значение Записать событие в журнал и обработать на сервере.

Создание операции **Финансовая проверка**

Данная операция аналогична предыдущей.

The screenshot shows a software window titled "Операция в схеме рабочего процесса «Утверждение заявки»". It contains several input fields and tabs. The "Имя" (Name) field is set to "Финансовая проверка". The "Тип" (Type) is a dropdown menu showing "Вопрос". The "Ссылка" (Link) field contains "A001005". There are buttons for "Создать" (Create) and "Дублировать" (Duplicate). Below these are tabs: "Общее" (General), "Параметры" (Parameters), "Лимит времени" (Time Limit), "Оповещения" (Notifications), and "История" (History). The "Общее" tab is active, showing a checked option "Записать задачу в журнал" (Record task in journal). The "Контекст" (Context) dropdown is set to "Заявки (apiRequest)". The "Рабочий процесс" (Workflow) is "Утверждение заявки". The "Условие ввода" (Input condition) is "OR". There is a large text area for "Комментарий" (Comment). A "Заккрыть" (Close) button is at the bottom right.

- Отличается назначенный: в данном случае, это финансовый контролер компании Gerald Colombo (назначенный отдельно). Он был создан в таблице ролей в рабочем процессе в предыдущих шагах с ролью финансового контролера. Для его уведомления выберите созданное ранее действие **Заявка для финансовой проверки** (поле **Действие**).
- Задержки и оповещения создаются, в основном, таким же образом, как для операции функционального утверждения.

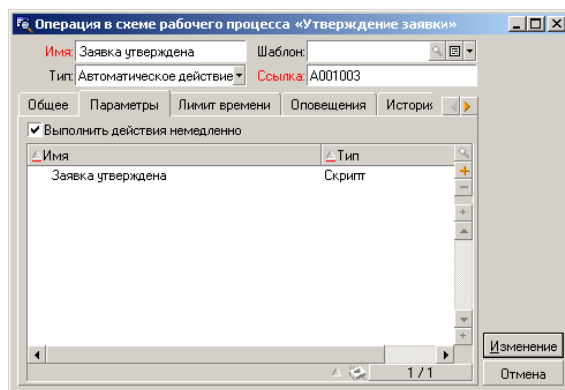
Создание операции **Заявка утверждена**

Если запрос успешно проходит два шага проверки, он утверждается.

Операция **Заявка утверждена** является одним из возможных завершений схемы рабочего процесса.

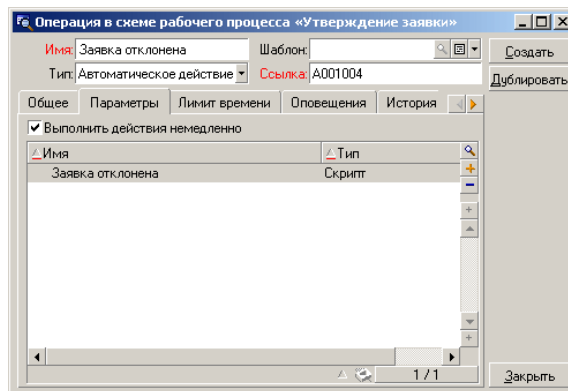
Данная операция необходима для изменения сведений о заявке, чтобы показать, что заявка утверждена.

Данная операция, таким образом, является операцией типа Автоматическое действие: действие, которое должно выполняться - действие **Заявка утверждена**, созданное в предыдущем шаге.



Создание операции **Заявка отклонена**

Операция **Заявка отклонена** подобна операции **Заявка утверждена**. В данном случае необходимо изменить сведения о заявке, чтобы показать, что заявка отклонена. Действие, которое должно выполняться - действие **Заявка отклонена**, созданное на промежуточном шаге.



Настройка событий, создаваемых одновременно с операциями

Когда в течение предыдущих шагов были созданы операции, были также созданы следующие события.

- Операция **Функциональное утверждение**:
 - Событие **Утверждение**
 - Событие **Отказ**
- Операция **Финансовая проверка**:
 - Событие **Утверждение**
 - Событие **Отказ**
- Операция **Заявка утверждена**:
 - Событие **Выполнено**
- Операция **Заявка отклонена**:
 - Событие **Выполнено**

Если выбрать все события друг за другом и выполнить следующие действия, то не придется зависеть от Asset Manager Automated Process Manager:

- 1 Щелкните правой кнопкой мыши имя события.
- 2 Выберите меню **Сведения о событии**.
- 3 Выберите вкладку **Общее**.
- 4 Выберите значение **Записать событие в журнал и обработать немедленно** для сохранения поля **Обработка** (имя SQL: seProcessingMode).
- 5 Нажмите кнопку **Изменить**.

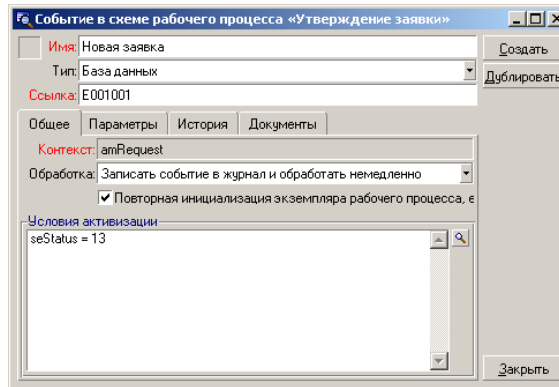
Создание начального события

События, которыми инициируется экземпляр рабочего процесса, связаны с операцией **Запуск**.

Чтобы определить событие **Запуск**, щелкните правой кнопкой мыши пустую область событий в операции **Запуск** и выберите Сведения о событии.

- 1 В данном случае экземпляр рабочего процесса инициируется, когда для поля **Статус заявки** (имя SQL: seStatus) в сведениях о заявке устанавливается значение Ожидается утверждение.

Поэтому начальное событие - событие типа База данных, и его параметры описываются на вкладке Общее, как показано на экране ниже:



- 2 Установите для поля **Обработка** (имя SQL: seProcessingMode) на вкладке **Общее** события значение Записать событие в журнал и обработать немедленно.
- 3 На вкладке **Параметры** события:
 - 1 Установите флажок **Обновление** (имя SQL: bUpdate).
 - 2 Укажите поле **seStatus** в поле **Отслеживаемые поля** (имя SQL: MonitFields).

Создание переходов

После создания операций их необходимо связать переходами.

Для создания перехода:

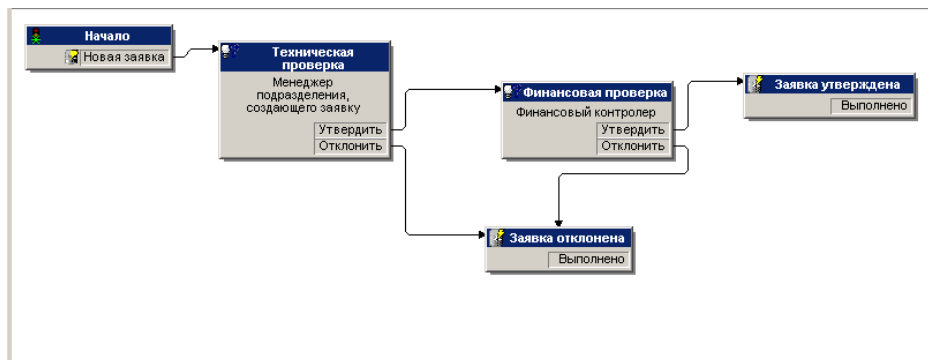
- 1 Щелкните начальное событие перехода.
- 2 Удерживая кнопку мыши нажатой, перетащите его в целевую операцию.

В данном случае необходимо создать следующие переходы:

- от начального события к операции **Техническая проверка**;
- от события **На утверждение** операции **Техническая проверка** к операции **Утверждение финансов**;
- от события **На утверждение** операции **Финансовая проверка** к операции **Заявка утверждена**;
- от событий **Отказ** операций **Техническая проверка** и **Финансовая проверка** к операции **Заявка отклонена**.

Получается следующая схема рабочего процесса:

Рисунок 9.4. Рабочий процесс - схема проверки заявки



Пример активации экземпляра рабочего процесса

Далее необходимо проверить, что схема рабочего процесса функционирует правильно.

Для этого необходимо выполнить следующее:

- 1 Заполнение таблицы «Сотрудники и подразделения» [стр. 175]
- 2 Создайте заявку на поставку, которую необходимо утвердить [стр. 176]
- 3 Контроль экземпляра [стр. 176]

Заполнение таблицы «Сотрудники и подразделения»

Перед созданием заявки на поставку, которую необходимо утвердить, важно определить запрашивающего и соответствующего ответственного за подразделения в таблице подразделений и сотрудников. У ответственного должны быть соответствующие права для выполнения следующих операций:

- 1 Создайте запрашивающего - **Jerome Carpenter**, относящегося к **подразделению IS**.
- 2 Присвойте имя пользователя (имя входа) **Carpenter**, пароль и профиль пользователя, позволяющие ему вводить заявку на поставку (вкладка **Профиль** в сведениях о соответствующей записи). Можно использовать профиль **Запрашивающий** в демонстрационной базе данных.
- 3 Присвойте значение **AM:Carpenter** полю **Электронная почта** (имя SQL: EMail).

- 4 Ответственный **подразделения IS - Philip Chavez**.
- 5 Чтобы упростить последующие операции, предоставьте административные права для базы данных сотруднику Philip Chavez: отобразите вкладку **Профиль** в сведениях о соответствующей записи и установите флажок **Права администрирования** (имя SQL: bAdminRight). Укажите Имя входа (имя SQL: UserLogin) и пароль для Philip Chavez.
- 6 Присвойте значение **AM:Carpenter** полю **Электронная почта** (имя SQL: EMail).
- 7 Выберите сотрудника **Gerald Colombo**.
- 8 Присвойте значение **AM:Colombo** полю **Электронная почта** (имя SQL: EMail) и значение **Colombo** полю **Имя входа** (имя SQL: UserLogin).

Создайте заявку на поставку, которую необходимо утвердить

Следующий шаг заключается в создании заявки на поставку, которую необходимо утвердить.

- 1 Подключитесь к демонстрационной базе данных с помощью имени входа Jerome Carpenter.
- 2 Щелкните пункт меню **Закупки/Заявки на поставку** в Asset Manager.
- 3 Нажмите кнопку **Создать**.
- 4 Выберите параметр **Создание новой заявки на поставку**.
- 5 Нажмите кнопку **ОК**.
- 6 В поле **Запрашивающий** (имя SQL: Requester) на вкладке **Общее** сведений о заявке выберите запись **Carpenter**.
- 7 Установите для поля **Статус заявки** (имя SQL: serStatus) в сведениях о заявке Готовится.
- 8 Подтвердите заявку на поставку: происходит начальное событие и Asset Manager регистрирует их в таблице элементарных событий рабочих процессов (имя SQL: amWfOccurEvent).

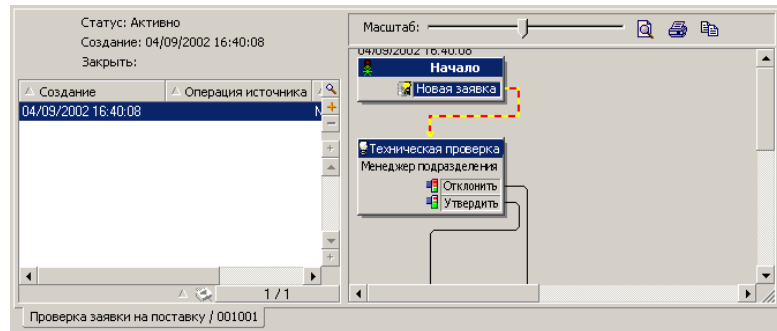
Контроль экземпляра

Чтобы проверить правильность функционирования экземпляра рабочего процесса, запустите Asset Manager и подключитесь к демонстрационной базе данных с помощью имени входа Philip Chavez, ответственного подразделения для Jerome Carpenter.

Просмотр экземпляра рабочего процесса

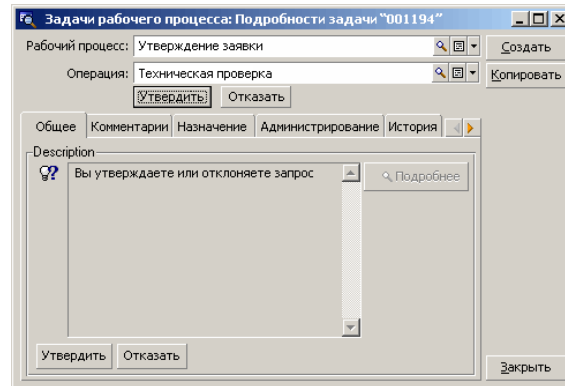
Отобразите сведения о созданной заявке на поставку: на вкладке **Рабочий процесс** перечисляются текущие экземпляры рабочих процессов для данной заявки. Каждый экземпляр описывается на отдельной подзакладке.

- Левая часть подзакладки перечисляет произошедшие события.
- В правой части показывается статус экземпляра. В данном случае должна мигать задача **Техническая проверка**.



Просмотр задачи для выполнения

- 1 Выберите пункт меню **Инструменты/Выполняемые задачи**: ответственный подразделения Philip Chavez может таким образом просматривать сведения о назначенной задаче:



На вкладке **Общее** отображается вопрос, определенный на вкладке **Вопрос** операции **Техническая проверка**.

На вкладке **Назначение** указывается, кто является ответственным за задачу, и соответствующий срок. Такая дата рассчитывается с помощью информации, содержащейся на вкладке **Лимит времени** в сведениях об операции **Техническая проверка**, и даты создания задачи (то есть даты активации перехода Сервером Asset Manager).

Можно щелкнуть **Сведения**, чтобы получить доступ к сведениям о заявке, ставшей причиной данной задачи.

- 2 Чтобы выполнить задачу, просто щелкните кнопку **Утверждение** или кнопку **Отказ**. Также можно ввести комментарий к данному решению на вкладке **Общее**.

Контекст рабочего процесса

Каждый экземпляр рабочего процесса имеет свой собственный отдельный контекст.

Определение контекста экземпляра рабочего процесса

При определении схемы рабочего процесса определяются:

- контекст рабочего процесса по умолчанию;
- контекст для всех событий, переходов и операций рабочих процессов (на экране сведений о переходе или на вкладке Общее сведений о событиях и операциях). Данный контекст связывается с контекстом по умолчанию экземпляра рабочего процесса.

В обоих случаях контекст представляет собой таблицу.

Объект, на который ссылается экземпляр рабочего процесса

Если запись удовлетворяет условиям активации, определенным в схеме рабочего процесса, инициируется экземпляр рабочего процесса. Данная запись становится ссылочным объектом на уровне начального события.

При выполнении экземпляра рабочего процесса ссылочный объект может изменять, в соответствии с контекстом, определенным на уровне операции, события и уровни перехода рабочего процесса.

Пример: когда утверждается заявка на поставку, инициируется экземпляр рабочего процесса. Им создается заказ на поставку, соответствующий данной заявке. Если утверждается заявка R1, она становится ссылочным объектом начального события. Ссылочный объект затем становится заказом на поставку PO1, то есть данный заказ создается из заявки на поставку.

Ограничение числа выполняющихся для данного объекта рабочих процессов

Параметр «Единственный активный экземпляр рабочего процесса для объекта» (имя SQL: bUniqueActive)

Asset Manager позволяет ограничивать число одновременно выполняющихся экземпляров рабочего процесса для данного объекта с помощью параметра Единственный активный экземпляр рабочего процесса для объекта на вкладке Общее сведений о рабочем процессе.

Если исходящее событие операции «Запуск» стало причиной появления второго экземпляра рабочего процесса для какого-либо объекта, результат определяется параметрами Единственный активный экземпляр рабочего процесса для объекта и Повторная инициализация экземпляра рабочего процесса, если один экземпляр уже выполняется (имя SQL: bReinitialize) (вкладка Общее сведений о событии):

В следующей таблице приводится краткая сводка различных возможных случаев:

		Параметр Один активный экземпляр рабочего процесса для объекта на вкладке Общее схемы рабочего процесса.	
		Проверено	Не проверено
Параметр Повторная инициализация экземпляра рабочего процесса, если один экземпляр уже выполняется на вкладке Общее исходящего события операции Запуск .	Проверено	Если один экземпляр рабочего процесса для объекта уже выполняется, он будет остановлен и запущен новый экземпляр рабочего процесса.	
	Не проверено	Если один экземпляр рабочего процесса для объекта уже выполняется, событие будет проигнорировано (новый экземпляр рабочего процесса запущен не будет).	Создан новый экземпляр рабочего процесса.

Пример приложения:

В случае схемы рабочего процесса, предназначенной для утверждения заявок на поставку, могут оказаться полезными следующие действия:

- Установите флажок Единственный активный экземпляр рабочего процесса для объекта, чтобы данная заявка на поставку обрабатывалась в единственном процессе утверждения.
- Установите флажок Повторная инициализация экземпляра рабочего процесса, если один экземпляр уже выполняется на уровне начального

события, чтобы перезапустить экземпляр, если изменился состав заявки на поставку.

Роли в рабочем процессе

Задачи, возникающие в результате определенных операций рабочих потоков, должны выполняться назначенным.



ЗАМЕЧАНИЕ:

Назначенные операций привлекаются только для операций типов Вопрос или Действие пользователя. Операции типов Автоматическое действие или Тест/скрипт назначенных не имеют.

Назначенные операции выбираются в таблице ролей в рабочем процессе (имя SQL: amWfOrgRole). Используйте ссылку **Администрирование/ Рабочие процессы/ Роли в рабочем процессе** на навигационной панели для доступа к таблице ролей в рабочем процессе.

Тип роли в рабочем процессе

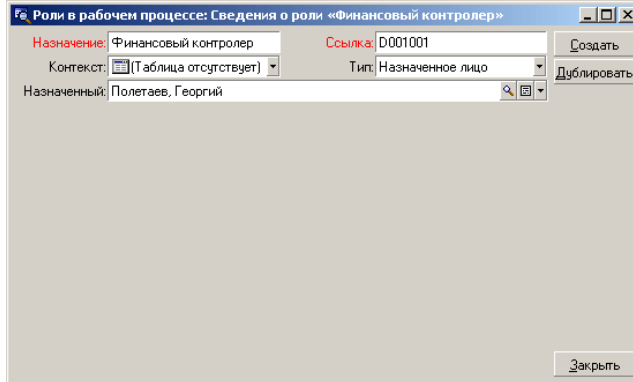
Есть несколько возможных типов ролей в рабочем процессе (поле Тип (имя SQL: seType) в сведениях о роли в рабочем процессе):

- Назначенное лицо.
- Рассчитанное лицо.
- Группа.
- Вычисляемая группа

Назначенное лицо

В данном случае назначенный выбирается непосредственно в таблице подразделений и сотрудников.

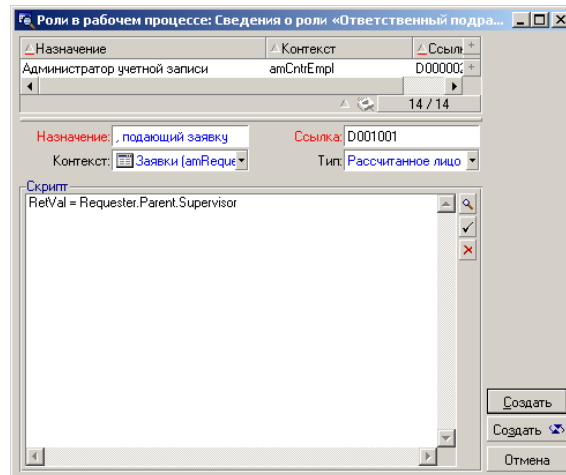
Пример:



Рассчитанное лицо

В данном случае назначенный присутствует в таблице подразделений и сотрудников, но рассчитывается с помощью скрипта.

Пример:



Группа

В данном случае назначенный выбирается в таблице групп сотрудников (amEmplGroup).

Вычисляемая группа

В данном случае назначенный рассчитывается с помощью скрипта из таблицы групп сотрудников (имя SQL: amEmplGroup).

Определение назначенного для операции

Поле Назначено (имя SQL: Assignee) во вкладке Вопрос (операция типа Вопрос) или во вкладке Действие (операция типа Действие пользователя) позволяет определять назначенных для операций типа Вопрос или Действие пользователя.

Операции рабочих процессов

Операции можно разделить на две группы:

- требующие взаимодействия с пользователем: операции типа Вопрос и Действие пользователя (поле Тип (имя SQL: LeType) в верхней части окна сведений об операции);
- выполняющиеся автоматически: операции типа Автоматическое действие и Тест/скрипт.

Значением поля Тип операции определяется, какие вкладки отображаются в сведениях об операции.

В данном разделе описываются следующие операции:

- операции типа «Вопрос»;
- операции типа «Действие пользователя»;
- операции типа «Автоматическое действие»;
- операции типа «Тест/скрипт»;
- Начать операцию
- Шаблоны операций
- Инициирование операций

Операции типа «Вопрос»

Данные операции требуют участия пользователя, указанного в поле Назначено (имя SQL: Assignee).

Для определения операций типа Вопрос указываются:

- вопрос или инструкции;
- возможные ответы.

Примеры:

- В ходе процесса утверждения поставки ответственному подразделению подается заявка от сотрудника подразделения.
- Операцию типа Вопрос можно также использовать как контрольную точку для подтверждения выполнения задачи. В таком случае был бы, например, единственный возможный ответ

Установите для поля Тип (имя SQL: seType) значение Вопрос. Отобразится вкладка Вопрос.

Укажите:

- 1 Запись в таблице ролей в рабочем процессе, соответствующая назначенному. Этот назначенный может быть уведомлен с помощью действия Asset Manager. Для этого достаточно лишь заполнить поле Уведомить лицо (имя SQL: bNotifAssignee) во вкладке Вопрос соответствующим образом.



ЗАМЕЧАНИЕ:

Действие, уведомляющее назначенного, инициируется, как только будет создана выполняемая задача, т.е. сразу после активации перехода, инициирующего операцию.

Назначенный использует пункт меню Инструменты/Выполняемые задачи для доступа к сведениям о задачах, подлежащих выполнению.

- 2 Текст вопроса или инструкций для выполнения.
- 3 Возможные ответы. Каждый ответ описывается на подзакладке. Он идентифицируется по своему описанию и имени SQL. Чтобы добавить, скопировать или удалить ответ, щелкните правой кнопкой мыши область меток подзакладки и выберите в контекстном меню Добавить связанную запись, Копировать связанную запись или Удалить ссылку.



ЗАМЕЧАНИЕ:

Каждым ответом автоматически создается исходящее событие операции.

Операция типа «Действие пользователя»

Данные операции требуют участия пользователя, называемого «назначенный». Назначенный показывается в поле Назначено (имя SQL: Assignee).

Для его определения указываются:

- инструкции для выполнения;
- мастер для выполнения.

Установите для поля Тип значение Действие пользователя, чтобы отобразить вкладку Действие.

Укажите:

- инструкции для выполнения;
- мастер для выполнения.
- Запись в таблице ролей в рабочем процессе, соответствующая назначенному. Этот назначенный может быть уведомлен с помощью действия Asset Manager. Для этого достаточно лишь заполнить поле Уведомить лицо (имя SQL: bNotifAssignee) во вкладке Действие соответствующим образом.



ЗАМЕЧАНИЕ:

Действие, уведомляющее назначенного, инициируется, как только будет создана выполняемая задача, т.е. сразу после активации перехода, инициирующего операцию.

Назначенный использует пункт меню Инструменты/Выполняемые задачи для доступа к сведениям о задачах, подлежащих выполнению.



ЗАМЕЧАНИЕ:

Событие выполнено создает автоматически как исходящее событие операции.

Пример: при управлении поставками можно использовать мастер, чтобы помочь пользователю принять поставку полностью или как часть от строк заказа, по которым ожидается поставка.

Операция типа «Автоматическое действие»

Данные операции выполняются автоматически Asset Manager или Asset Manager Automated Process Manager.

Описание

Операциями типа Автоматическое действие перечисляются действия, которые необходимо выполнить.

Пример: при выполнении операции «Перемещение активов» операцией типа Автоматическое действие автоматически изменяется местоположение всех активов, чьи родительские активы оказались перемещенными.

Установите для поля Тип значение Автоматическое действие, чтобы отобразить вкладку Действия.

Укажите здесь список выполняемых действий.



ЗАМЕЧАНИЕ:

Событие выполнено **создает автоматически как исходящее событие операции.**

Выполнение

Механизм рабочего процесса, который активизирует переход, инициирующий операцию, немедленно выполняет действия операции. В зависимости от выбранных опций, эти действия будут обработаны Asset Manager Automated Process Manager или агентом в Asset Manager.

- Если выбран параметр Выполнить действия немедленно (имя SQL: bExecImmediatelly), механизм рабочего процесса, который активизирует переход, инициирующий операцию, немедленно выполняет действия операции.
- В противном случае задачи будут выполнены Asset Manager Automated Process Manager во время следующего цикла проверки.

Операции типа «Тест/скрипт»

Данные операции автоматически выполняются Asset Manager или Asset Manager Automated Process Manager.

Описание

Данные операции определяются скриптом и возможными результатами.

Пример: в области управления запасом и заявками на поставку можно использовать операцию типа «Тест/скрипт», чтобы проверить, что позиции, на которые есть ссылки в строках заказа на поставку, доступны в запасе и не зарезервированы. В таком случае данной операцией может инициироваться операция типа Вопрос, которая спрашивает запрашивающего о необходимости резервирования позиций в запасе.

Установите для поля Тип значение Тест/скрипт, чтобы отобразить вкладку Тест.

Укажите:

- тестовый скрипт для выполнения.
- возможные результаты. Каждый результат описывается на отдельной подзакладке. Он идентифицируется по своему описанию и имени SQL. Чтобы добавить, скопировать или удалить возможный результат, щелкните правой кнопкой мыши область меток подзакладки и выберите в контекстном меню Добавить связанную запись, Копировать связанную запись или Удалить ссылку.



ЗАМЕЧАНИЕ:

Внимание: имена SQL каждого результата должны соответствовать возвращаемым значениям тестового скрипта.



ЗАМЕЧАНИЕ:

Каждым результатом автоматически создается исходящее событие операции.

Выполнение

Механизм рабочего процесса, который активизирует переход, инициирующий операцию, немедленно выполняет действия операции. В зависимости от выбранных опций, эти действия будут обработаны Asset Manager Automated Process Manager или агентом в Asset Manager.

- Если установить флажок «Опция немедленного выполнения действий» (имя SQL: bExecImmediately), механизм рабочего процесса, активирующий переход, который инициирует операцию, автоматически выполняет действия операции: в соответствии с режимом обработки, выбранным для события, которое инициирует переход, Asset Manager Automated Process Manager или агент Asset Manager выполняет действия.
- В противном случае задачи будут выполнены Asset Manager Automated Process Manager во время следующего цикла проверки.

Начальная операция

Операция Запуск - начальная точка схемы рабочего процесса.

Она является обязательной и автоматически создается при создании схемы рабочего потока. Нельзя изменять сведения о ней.

Ею не определяется работа, которая должна быть выполнена.

Исходящими событиями операции Запуск инициируется экземпляр рабочего процесса.

Шаблоны операций

Шаблоны операций упрощают создание операций схем рабочих потоков.

Они хранятся в таблице операций (имя SQL: amWfActivity).

Для доступа к списку шаблонов операций используется пункт меню Инструменты/Рабочие процессы/Шаблоны операций.

 ЗАМЕЧАНИЕ:

Предостережение: чтобы информация, содержащаяся в сведениях о шаблоне операции (тип операции и т.п.), автоматически копировалась на уровень операций, ссылающихся на данный шаблон (поле Шаблон (имя SQL: Template) в сведениях об операции), необходимо, чтобы администратор Asset Manager определил соответствующие значения по умолчанию полей и ссылок в сведениях об операции.

Инициирование операций

Чтобы инициировать операцию, необходимо заполнить поле Условие ввода (имя SQL: selnCond) на вкладке Общее сведений об операции. Данное условие относится к переходам, которыми иницируется операция.

- Если есть только один переход, которым может иницироваться операция, должен активироваться только данный переход (в Asset Manager или Asset Manager Automated Process Manager), чтобы иницировалась данная операция.
- Если есть несколько переходов, которыми может иницироваться операция:
 - Если условие ввода операции - AND, для иницирования операции должны быть активизированы все переходы.
 - Если условие ввода операции - OR, для иницирования операции необходимо активизировать только один из переходов.

 ЗАМЕЧАНИЕ:

Если условия ввода являются сложными (комбинациями условий AND и OR), для их выполнения можно создать последовательность промежуточных операций типа Тест/скрипт.

Задачи

В данном разделе объясняется, как создаются и выполняются задачи рабочих процессов.

Создание задач

Когда активизируется переход, иницирующий операцию, механизмом рабочего процесса, активизировавшим данный переход, автоматически создается задача, подлежащая выполнению.

В соответствии с параметром, выбранным в поле Записать задачу в журнал (имя SQL: bLogWorkItem) на вкладке Общее операции, данная задача заносится в таблицу задач рабочих процессов (имя SQL: WkElem).

Параметр Записать задачу в журнал проверяется автоматически.

- Для операций типа Вопрос или Действие пользователя.
- Для операций типов Автоматическое действие или Тест/скрипт, для которых не выбран параметр Выполнить действия немедленно (имя SQL: bExecImmediately).



ВНИМАНИЕ:

Если задача не записывается в журнал, невозможно создать оповещения рабочих процессов, связанные с данной задачей: не отображаются вкладки Лимит времени и Оповещения в сведениях об операции, если не выбран параметр Записать задачу в журнал.

Задача выполняется по-разному, в зависимости от того, обязательно ли участие пользователя или нет.

Операция типа «Автоматическое действие» или «Тест/скрипт»

Если появление задачи стало результатом операций типов Автоматическое действие или Тест/скрипт, для которых выбран параметр Выполнить действия немедленно (имя SQL: bExecImmediately), задача выполняется немедленно механизмом рабочего потока, активировавшим переход, которым создана задача. Это может быть Asset Manager Automated Process Manager или агент Asset Manager.

С другой стороны, Asset Manager Automated Process Manager через регулярные интервалы времени проверяет, есть ли необходимость в выполнении задач рабочего потока, и выполняет их, если необходимо.

Частота, с которой Asset Manager Automated Process Manager отслеживает функции рабочего процесса, определяется опциями Asset Manager Automated Process Manager.

Отображение списка всех задач рабочего процесса

Для отображения списка всех задач рабочего процесса откройте таблицу Задачи рабочего процесса (amWfWorkItem) (ссылка **Администрирование/ Рабочие процессы/ Задачи рабочего процесса** на навигационной панели).

Отображаемый список содержит все задачи рабочего процесса, независимо от их статуса (в ожидании, завершена, закрыта), для всех экземпляров рабочего процесса.

Администратор может также обращаться к списку выполняемых задач для конкретной операции из сведений о схеме рабочего процесса. Для этого:

- 1 Щелкните правой кнопкой мыши по операции.
- 2 Выберите Задачи... в контекстном меню.

Выполнение задачи пользователя

Назначенное лицо получает доступ к своим задачам по ссылке **Управление портфелем/Мои задачи рабочего процесса** на навигационной панели.

При более подробном рассмотрении, пользователь видит:

- Задачи, назначенные пользователю и подлежащие выполнению пользователем.
- задачи, которые назначены для группы, к которой он относится, но не назначены конкретному лицу.

Можно щелкнуть **Сведения**, чтобы получить доступ к сведениям об объекте, на который ссылается задача.

Чтобы выполнить задачу, подлежащую выполнению, отобразите вкладку Общее данной задачи:

- Если операция, причиной которой стала данная задача, представляет собой задачу типа Вопрос, на вкладке Общее отображаются текст вопроса или инструкции для выполнения. Возможные результаты имеют соответствующие кнопки. Щелкните соответствующую кнопку. Если необходимо, можно также ввести комментарий к принятому решению.
- Если операция представляет собой операцию типа Действие пользователя, просто щелкните кнопку Мастер, чтобы запустить мастер, который необходимо выполнить.

Кроме того, назначенное лицо может получить доступ к задачам, порученные ему/ей, по ссылке **Управление портфелем/ Мои порученные задачи** на навигационной панели. ► [Поручение задачи](#) [стр. 190]

Назначение задач пользователей

Информация о назначении задач пользователей показывается на вкладке Назначение в окне подробностей о задаче.

При наличии необходимых прав можно изменить назначение задачи пользователя:

- значение поля Назначение (имя SQL: seAssignment);
- назначенного задачи.

Поручение задачи

Поручения позволяют пользователям передавать назначения задач от одного пользователя (Поручающий сотрудник) другому пользователю (Сотрудник, получивший поручение); пользователи могут делать это в течение определенного или неопределенного периода времени.

Создание поручения

Чтобы создать поручение, используйте мастер Создать/Модифицировать поручения... (sysCoreDelegation), доступ к которому можно получить из сведений о сотрудниках, а также щелкнув Поручить (Windows-клиент) или выбрав Поручить из раскрывающегося меню Действия... (веб-клиент).

► Руководство Общие таблицы, глава Подразделения и сотрудники, раздел Создать поручение.



ЗАМЕЧАНИЕ:

Если для поручения не была указана дата окончания (dtEndDeleg), поручение является постоянным.



ВАЖНО:

Для активации поручения выберите опцию Активировать (bActive) в сведениях о поручении.

Это позволяет временно деактивировать поручение, не удаляя его или не изменяя дату его окончания.



ВНИМАНИЕ:

Если поручение создано, профили или права на проверку не передаются. Это значит, что сотрудник Б, получающий поручение, должен иметь такие же права на проверку, как и сотрудник А, для того, чтобы принять или отклонить задачи, порученные ему/ей.

Windows-клиент: просматривать существующие поручения можно в сведениях о сотруднике. Щелкните вкладку **Поручения**, а затем подвкладки

Предварительный просмотр предоставленных поручений и **Предварительный просмотр полученных поручений**.

Дополнительную информацию см. в руководстве Интерфейс пользователя, глава Средство просмотра планировщика (Windows-клиент).

Порученные задачи: обзор и просмотр

Когда создается и активируется поручение от сотрудника А сотруднику Б, все задачи рабочего процесса, назначенные сотруднику А или группе, членом которой является сотрудник А, отсылаются сотруднику А и сотруднику Б.

Соответственно:

- Сотрудник А все еще может видеть все назначенные задачи рабочего процесса по ссылке **Управление портфелем/ Мои задачи рабочего процесса** на навигационной панели.
- Сотрудник Б видит все назначенные задачи рабочего процесса по ссылке **Управление портфелем/ Мои задачи рабочего процесса** на навигационной панели, а также все задачи рабочего процесса, переданные ему/ей посредством поручения по ссылке **Управление портфелем/ Мои порученные задачи рабочего процесса** на навигационной панели.

В период поручения сотрудник Б может просматривать, проверять или отклонять все задачи рабочего процесса, назначенные сотруднику А, а также задачи рабочего процесса, назначенные группе проверки, членом которой является сотрудник А.

Задачи рабочего процесса, которые сотрудник Б может просматривать по ссылке **Управление портфелем/ Мои порученные задачи рабочего процесса** на навигационной панели, все еще отображаются назначенными сотруднику А (или группе проверки, членом которой является сотрудник А).

Администрирование задачи рабочего процесса

Информация об администрировании задачи рабочего потока содержится на вкладке Администрирование в окне подробностей о задаче.

Данная информация доступна только пользователям с административными правами.

События

События, связанные с операциями. Ими инициируются переходы к другим операциям.

Есть три возможных типа систем для события на уровне операции. Тип системы события определяется полем Тип системы (имя SQL: seType) в сведениях о событии:

- событие Системы;

- событие Пользователя;
- событие Оповещения.

Событие системы

События Системы автоматически определяются в Asset Manager при создании или определении операций.

Они соответствуют различным возможным итогам (результатам) работы, выполненной операцией:

- ответы на операцию типа Вопрос;
- результаты операции типа Тест/скрипт;
- событие выполнено в случае операции типа Действие пользователя или Автоматическое действие.

Пример: если операцией создается вопрос, возможными ответами на который являются «Да» и «Нет», на уровне операции создаются два системных события с названиями «Да» и «Нет».

Событие оповещения

События оповещения для операции создаются, когда определяются оповещения операций, которыми иницируются события.

Такое оповещение определяется на вкладке Оповещения сведений об операции. Событие носит то же самое имя, что и оповещение.

Событие пользователя

События пользователей не зависят от задач, выполняемых в операции. Они создаются вручную с помощью графического редактора рабочих потоков (команда Добавить событие контекстного меню).



ЗАМЕЧАНИЕ:

События, связанные с операцией Запуск - события пользователей.

Есть два типа событий Пользователя (поле Тип (имя SQL: seMonitoringType) в верхней части окна сведений о событии):

- База данных
- Периодическое

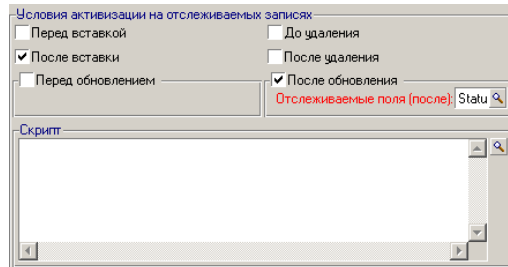
Событие типа «База данных»

События типа База данных позволяют активизировать экземпляры рабочих процессов для конкретных записей.

Событие типа База данных происходит:

- при выполнении общих условий активации, указанных на вкладке Общее;
- при проверке определенных параметров инициирования на уровне отслеживаемых записей.

Рисунок 9.5. Вкладка параметров действия типа «База данных»



Параметры, которыми иницируется событие типа База данных, описаны на вкладке Параметры сведений о событии. Указывается следующая информация.

- Записи, которые должны отслеживаться (такие записи могут быть записями в таблице, указанной в контексте, или связанными записями). Если записи, которые должны отслеживаться, это записи, связанные с таблицей, указанной в контексте, укажите соответствующую ссылку в поле Ссылка/контекст (имя SQL: LinkToMonitTable).
- Условия активации событий относятся к отслеживаемым записям. Для указания условий активации можно использовать:
 - параметр Перед вставкой (имя SQL: bPreInsert), позволяет активизировать переход до вставки новой записи;



ВНИМАНИЕ:

Эту опцию нельзя использовать, если схема рабочего процесса создает запись в таблице AND выполняет действие в записи, связанной с данной записью.

Например, если схема рабочего процесса создает позицию портфеля и модифицирует поле актива, автоматически созданное позицией портфеля.

Вместо этого используйте опцию После вставки.

- параметр После вставки (имя SQL: bInsert), позволяет активизировать переход после вставки новой записи;



ВНИМАНИЕ:

Эту опцию нельзя использовать, если схеме рабочего процесса необходимо выполнить действие типа скрипта для заполнения поля или ссылки при помощи функции set().

Вместо этого используйте опцию До вставки.

- параметр До удаления (имя SQL: bDelete), позволяет активизировать переход до удаления записей;
- параметр После удаления (имя SQL: bPostDelete), позволяет активизировать переход после удаления записей;
- параметр **Перед обновлением** (имя SQL: bPreUpdate), позволяет указать поля, для которых будет выполняться переход, до обновления данных полей. Данные поля определяются в поле **Отслеживаемые поля (до)**;
- параметр После обновления (имя SQL: bUpdate), позволяет указать поля, которые, в случае изменения, должны учитываться в поле Отслеживаемые поля (после) (имя SQL: MonitFields). Для указания нескольких имен полей они отделяются с помощью запятых. Если оставить данное поле пустым, измененные поля не будут учитываться;



ВНИМАНИЕ:

Невозможно, чтобы условия активации события вызвали разрушение объекта, на который ссылается контекст.

- скрипт, записанный в области Скрипт (имя SQL: memScript). Если написать скрипт и установить один или несколько флажков Вставка, Обновление и Удаление, данный скрипт будет ограничивать условия активации.

Пример: если событие должно инициироваться при изменении цены существующей заявки, необходимо заполнить вкладку Параметры следующим образом.

Когда происходит событие типа База данных, оно учитывается компьютером клиента Asset Manager, на котором оно произошло. Способ его обработки зависит от того, какой параметр был выбран в поле Обработка (имя SQL: seProcessingMode) на вкладке Общее сведений о событии.

Дополнительную информацию можно найти в разделе «Обработка событий» главы «Рабочий процесс» данного руководства.

Ссылка «Старая»

Ссылка «Старая» указывает на текущую запись, отслеживаемую событием, до изменения одного из ее компонентов. При использовании в скрипте в форме [Old.<имя SQL поля>] ею возвращается значение поля записи до изменения. Таким образом, можно выполнить тесты с помощью выражения:

```
if [Old.IUserId] <> IUserId Then....
```

Событие типа «Периодическое»

События типа Периодическое имеют отношение к выбору записей в указанной таблице. Они позволяют инициировать экземпляр рабочего процесса на периодической основе для каждой записи выбора.

Пример: каждый месяц обновляются значения остатков активов с видом «ПК».

События типа Периодическое происходят при выполнении условий активации, указанных на вкладке Общее.

В данном случае такое событие инициируется Asset Manager Automated Process Manager.

Частота, с которой Asset Manager Automated Process Manager инициирует события типа Периодическое, определяется в планировщике во вкладке Параметры сведений о событии.

Способ, которым обрабатывается данное событие, описан в разделе «Обработка событий» главы «Рабочий процесс» данного руководства.

Общие условия активации

Условия активации для всех типов событий можно определить на вкладке Общее.

Условие AQL (имя SQL: AQLCond)

С помощью поля Условие AQL указывается выбор записей для схемы рабочего процесса.

Повторная инициализация экземпляра рабочего процесса, если один экземпляр уже выполняется (имя SQL: bReinitialize)



ЗАМЕЧАНИЕ:

Параметр Повторная инициализация экземпляра рабочего процесса, если один экземпляр уже выполняется **показывается в сведениях только тех событий, которые являются результатом операции «Запуск».**

Параметром Повторная инициализация экземпляра рабочего процесса, если один экземпляр уже выполняется определяется, что произойдет, если

исходящее событие операции Запуск затронет объект базы данных, который уже используется каким-либо экземпляром схемы рабочего процесса.

Что происходит, зависит не только от данного параметра, но также от параметра Единственный активный экземпляр рабочего процесса для объекта (имя SQL: bUniqueActive) на вкладке Общее схемы рабочего процесса.

В следующей таблице приводится краткая сводка различных возможных случаев:

Таблица 9.1. Ограничение числа выполняющихся для данного объекта рабочих процессов - различные возможные случаи

		Параметр Один активный экземпляр рабочего процесса для объекта на вкладке Общее схемы рабочего процесса.	
		Проверено	Не проверено
Параметр Повторная инициализация экземпляра рабочего процесса, если один экземпляр уже выполняется на вкладке Общее исходящего события операции Запуск .	Проверено	Если один экземпляр рабочего процесса для объекта уже выполняется, он будет остановлен и запущен новый экземпляр рабочего процесса.	
	Не проверено	Если один экземпляр рабочего процесса для объекта уже выполняется, событие будет проигнорировано (новый экземпляр рабочего процесса запущен не будет).	Создан новый экземпляр рабочего процесса.

Обработка событий

- После выполнения общих условий активации способ обработки событий зависит от:
- типа событий (поле Тип (имя SQL: seMonitoringType) в верхней части окна сведений о событии);
 - параметра, выбранного в поле Обработка (имя SQL: seProcessingMode) на вкладке Общее сведений о событии.
- В следующей таблице собраны различные способы, которыми может быть обработано событие.

Таблица 9.2. Различные способы, которыми может быть обработано событие

	Записать событие в журнал и обработать на сервере	Записать событие в журнал и обработать немедленно	Обработать событие немедленно без записи в журнал
Событие типа Периодическое .	Asset Manager Automated Process Manager инициирует событие, если выполнены условия активации. Частота инициирования определяется на вкладке Параметры сведений о событии. Как только оно происходит, Сервер Asset Manager записывает событие в журнал в таблице с именем SQL amWfOccurEvent. Переход активируется Asset Manager Automated Process Manager позже (частота, с которой Asset Manager Automated Process Manager отслеживает переходы, которые подлежат активации, определяется опциями в Asset Manager Automated Process Manager).	Asset Manager Automated Process Manager инициирует событие, если выполнены условия активации. Частота инициирования определяется на вкладке Параметры сведений о событии. Как только оно происходит, Сервер Asset Manager записывает событие в журнал в таблице с именем SQL amWfOccurEvent. Переход активируется немедленно Asset Manager Automated Process Manager.	Asset Manager Automated Process Manager инициирует событие, если выполнены условия активации. Частота инициирования определяется на вкладке Параметры сведений о событии. Как только происходит событие, оно не записывается в журнал в таблицу с именем SQL amWfOccurEvent, но переход активируется немедленно Asset Manager Automated Process Manager.

	Записать событие в журнал и обработать на сервере	Записать событие в журнал и обработать немедленно	Обработать событие немедленно без записи в журнал
Событие типа База данных или событие системы, инициированное Asset Manager (в результате операции типа Вопрос или Действие пользователя , в результате операции типа Автоматическое действие или Тест/скрипт , выполняемой Asset Manager)	Как только происходит событие, оно записывается в журнал в таблицу с именем SQL amWfOccurEvent клиентским компьютером Asset Manager. Переход активируется Asset Manager Automated Process Manager позже (часто, с которой Asset Manager Automated Process Manager отслеживает переходы, которые подлежат активации, определяется опциями в Asset Manager Automated Process Manager).	Как только происходит событие, оно записывается в журнал в таблицу с именем SQL amWfOccurEvent клиентским компьютером Asset Manager. Переход активизируется немедленно клиентским компьютером Asset Manager.	Как только происходит событие, оно не записывается в журнал в таблицу с именем SQL amWfOccurEvent , но переход активизируется немедленно клиентским компьютером Asset Manager.
Событие системы, инициируемое Asset Manager Automated Process Manager (результат операции типа Тест / скрипт или Автоматическое действие , выполняемой Asset Manager Automated Process Manager) или событие оповещения об операции	Как только происходит событие, оно записывается в журнал в таблицу с именем SQL amWfOccurEvent Сервером Asset Manager Переход активируется Asset Manager Automated Process Manager позже (часто, с которой Asset Manager Automated Process Manager отслеживает переходы, которые подлежат активации, определяется опциями в Asset Manager Automated Process Manager).	Как только происходит событие, оно записывается в журнал в таблицу с именем SQL amWfOccurEvent Сервером Asset Manager. Переход активируется немедленно Asset Manager Automated Process Manager.	Как только происходит событие, оно не регистрируется в таблице с именем SQL amWfOccurEvent , но переход немедленно активируется Asset Manager Automated Process Manager.

Таблица 9.3. Различные способы, которыми может быть обработано событие

	Записать событие в журнал и обработать на сервере	Записать событие в журнал и обработать не-медленно
Событие типа Периодическое .	Asset Manager Automated Process Manager инициирует событие, если выполнены условия активации. Частота инициирования определяется на вкладке Параметры сведений о событии. Как только оно происходит, Сервер Asset Manager записывает событие в журнал в таблице с именем SQL amWfOccurEvent . Переход немедленно активируется Asset Manager Automated Process Manager.	Asset Manager Automated Process Manager инициирует событие, если выполнены условия активации. Частота инициирования определяется на вкладке Параметры сведений о событии. Когда это происходит, Asset Manager Automated Process Manager не регистрирует событие в таблице с именем SQL amWfOccurEvent . Но переход немедленно активируется Asset Manager Automated Process Manager.
Событие типа База данных или событие системы, инициированное Asset Manager (в результате операции типа Вопрос или Действие пользователя , в результате операции типа Автоматическое действие или Тест/скрипт , выполняемой Asset Manager)	Как только происходит событие, оно записывается в журнал в таблицу с именем SQL amWfOccurEvent клиентским компьютером Asset Manager. Переход активизируется немедленно клиентским компьютером Asset Manager.	Как только происходит событие, оно не записывается в журнал в таблицу с именем SQL amWfOccurEvent , но переход активизируется немедленно клиентским компьютером Asset Manager.
Событие системы, инициируемое Asset Manager Automated Process Manager (результат операции типа Тест / скрипт или Автоматическое действие , выполняемой Asset Manager Automated Process Manager) или событие оповещения об операции	Как только происходит событие, оно записывается в журнал в таблицу с именем SQL amWfOccurEvent Сервером Asset Manager. Переход немедленно активируется Asset Manager Automated Process Manager.	Когда происходит событие, оно не записывается в журнал в таблицу с именем SQL amWfOccurEvent , но переход немедленно активируется Asset Manager Automated Process Manager.

Такие различные режимы обработки позволяют точно указывать, как выполняется экземпляр рабочего процесса.

В соответствии с выбором, сделанным на уровне:

- типов событий;

- режимов обработки событий;
- операций

Можно определять и синхронную, и асинхронную схемы рабочих процессов или объединять оба подхода.

Применение: реализация синхронной схемы рабочего процесса

При создании схемы синхронного рабочего процесса необходимо определить:

- Событие типа База данных - Записать событие в журнал и обработать немедленно (поле Обработка (имя SQL: seProcessingMode) во вкладке Общее в сведениях о событии).
- Операции типов Автоматическое действие или Тест/скрипт, для которых выбран параметр Выполнить действия немедленно (имя SQL: bExecImmediately) и которые инициируются данными событиями.

Пример:

С помощью схемы рабочего потока, описанной на диаграмме ниже, как только какой-либо актив изменяет местоположение, его подактивы автоматически перемещаются в то же самое местоположение:

Рисунок 9.6. Пример схемы синхронного рабочего процесса



В данном случае, при изменении местоположения актива и при щелчке по кнопке **Изменить**:

- 1 Начинается транзакция базы данных.
- 2 Изменяется местоположение актива.
- 3 Запускается экземпляр рабочего процесса.
- 4 Активируется переход рабочего процесса.

5 Изменяется местоположение подактивов.

6 Затем проверяется вся транзакция.

Если во время одного из шагов происходит ошибка, местоположения и актива, и подактивов остаются неизменными.

Если шаги успешно выполняются, изменяются все местоположения.

С другой стороны, если такая же процедура реализуется с помощью асинхронной схемы рабочего потока, как описано ниже, и если происходит ошибка, местоположение актива можно изменить, не изменяя местоположения подактивов.

Рисунок 9.7. Пример схемы асинхронного рабочего процесса



Заключительное событие

Определение

Заключительным событием заканчивается экземпляр рабочего процесса, даже если остаются задачи, которые должны быть выполнены.

Пример:

Рисунок 9.8. Схема рабочего процесса с заключительным событием



Если создается экземпляр рабочего процесса, как показано выше, и:

- происходит исходящее событие операции 1, которым инициируется операция 2, создавая задачу, подлежащую выполнению;
- происходит заключительное событие операции 3;

то экземпляр рабочего процесса прекращается, даже если не выполнена задача, созданная операцией 2.

Указание того, что событие является заключительным

При создании схемы рабочего процесса с помощью графического редактора на вкладке Операции схемы рабочего процесса, указание, что событие является заключительным, выполняется следующим образом.

- 1 Щелкните правой кнопкой мыши по событию.
- 2 Выберите пункт Заключительное событие в контекстном меню.

Переходы рабочих процессов

Переходами связываются исходящие события одной операции с другими операциями.

Событие может связываться с несколькими переходами.

При необходимости можно указать условия активации перехода в поле Условие AQL (имя SQL: AQLCond) в сведениях о переходе.

Оповещения рабочих процессов и лимиты времени

Для каждой операции рабочего потока можно определить:

- предельный срок выполнения;

- оповещения, связанные с данным предельным сроком или датами, хранящимися в базе данных. Данными оповещениями запускаются действия.



ВНИМАНИЕ:

Если не активизирован параметр Записать задачу в журнал (имя SQL: bLogWorkItem) на вкладке Общее операции, нельзя определять лимиты времени или оповещения.

Лимит времени

Лимит времени для выполнения операции рабочего процесса определяется на вкладке Лимит времени сведений об операции.



ЗАМЕЧАНИЕ:

Вкладка Лимит времени в сведениях об операции отображается, только если проверен параметр Записать задачу в журнал (имя SQL: bLogWorkItem) на вкладке Общее сведений об операции.

Данный лимит времени определяется в связи с моментом времени, при котором инициируется операция.

Он связывается с календарем рабочих дней.

Можно указать период времени или выбрать один из трех предварительно определенных параметров:

- Конец следующего рабочего дня
- Конец рабочей недели
- Конец рабочего месяца



ВНИМАНИЕ:

Если указывается период времени, Asset Manager истолковывает его как выраженный в единицах рабочего времени и преобразовывает его в рабочие часы. Пример: если ввести «2 дня», такое значение будет рассматриваться как 48 рабочих часов.

Оповещения рабочих процессов

Можно связать оповещения с каждой операцией на вкладке Оповещения сведений об операции.



ЗАМЕЧАНИЕ:

Вкладка Оповещения в сведениях об операции отображается, только если проверен параметр Записать задачу в журнал (имя SQL: bLogWorkItem) на вкладке Общее сведений об операции.

Лимиты времени

Для определения лимитов времени, которыми инициируются оповещения, можно использовать:

- период времени после даты, хранящейся в базе данных (тип Время, прошедшее с начала задачи);
- период времени до даты, хранящейся в базе данных (тип Время до конца задачи);
- в процентах от лимита времени на выполнение операции (поле Лимит времени (имя SQL: tsResolDelay) на вкладке Лимит времени).



ЗАМЕЧАНИЕ:

Периоды времени, которыми определяется лимиты времени рабочих процессов, выражаются в рабочих днях.

После создания задачи создаются связанные оповещения рабочих процессов. Лимиты времени рабочих процессов отслеживаются Asset Manager Automated Process Manager. Частота отслеживания определяется параметрами Asset Manager Automated Process Manager.

Результат воздействия оповещений

Оповещениями инициируются:

- действия Asset Manager;
- или события. События, инициируемые оповещениями, являются событиями типа Оповещение. Их имена совпадают с именами оповещений, которыми они определяются.

Группы выполнения рабочих процессов

Группы выполнения рабочих процессов позволяют распределять по категориям схемы рабочих процессов. Группа выполнения, к которой относится данная схема рабочего процесса, указывается в поле Группа выполнения (имя SQL: GroupName) на вкладке Общее сведений о рабочем процессе.

Asset Manager Automated Process Manager отслеживает создание новых групп выполнения рабочих процессов.

Как только Asset Manager Automated Process Manager обнаруживает новую группу G выполнения рабочих процессов, он создает новый модуль мониторинга Выполнение правил рабочего процесса для группы выполнения G.

Такой механизм полезен по следующим причинам:

- он позволяет определять расписания проверки для каждой группы выполнения рабочего потока;
- разные группы выполнения рабочих процессов можно отслеживать разными экземплярами Asset Manager Automated Process Manager.

После обнаружения группы выполнения рабочих процессов Asset Manager Automated Process Manager отслеживаются и выполняются соответствующие правила рабочих процессов (оповещения отслеживания, обработка событий типа Периодическое, активация переходов, выполнение задач и так далее).

Приоритет

Для каждой группы необходимо определить приоритет. Данный приоритет соответствует порядку выполнения групп рабочих процессов и позволяет определять, какой рабочий поток будет запускаться в указанный период времени.

Уровень приоритета определяется в поле **Приоритет**. Уровень приоритета 1 более высокий, чем уровень приоритета 3.

Отслеживание рабочего процесса

Отображение экземпляров рабочего процесса в записи

Если какая-либо таблица в Asset Manager определяется как контекст начального объекта схемы рабочего процесса, в представлении сведений об этой таблице отображается вкладка Рабочий процесс.

На данной вкладке Рабочий процесс показывается статус выполняемых экземпляров рабочих процессов, использующих данную запись как начальный объект.

Windows-клиент: можно отобразить список экземпляров рабочего процесса в виде подвкладок или списка.

Веб-клиент: доступно только представление в виде списка.

Каждый экземпляр рабочего процесса указывает способ развития экземпляра:

- Windows-клиент:
В левой части указываются произошедшие события.
В правой части отображается рабочая схема. Выполняемые операции показываются как мигающие. Следующие шаги при этом затенены.
- Веб-клиент:
В верхней части указываются произошедшие события.
В нижней части отображается рабочая схема. Ячейки, представляющие операции, подлежащие выполнению, - голубого цвета, а другие ячейки - темно-синего цвета.

Просмотр информации на разных этапах экземпляра

- Чтобы просмотреть информацию о завершенных этапах или об этапах, которые еще выполняются в отношении экземпляра рабочего процесса (например, кому назначена задача), следует наводить курсор на разные ячейки схемы рабочего процесса (передача информации отображается непосредственно в момент выполнения этого действия).
- Текстовое описание рабочего процесса и завершенных событий можно отобразить, щелкнув на значок .
- Windows-клиент: посредством функции масштабирования можно изменять масштаб схемы. Эта функция недоступна в веб-клиенте.

Обновление схем и экземпляров рабочего процесса

Windows-клиент

Обновление схем и экземпляров рабочего процесса выполняется динамически: оно осуществляется автоматически при переходе схемы рабочего процесса с одного этапа на другой, или если выполняются модификации самой схемы.

Веб-клиент

Существует два типа информации:

- Информация, относящаяся к экземпляру рабочего процесса: завершенные события, статус развития (цветные стрелки), дата завершения каждой операции и т.д. Эта информация обновляется динамически: графическое представление экземпляра рабочего процесса мгновенно обновляется с учетом изменений (используйте клавишу **F5** для обновления экрана).
- Информация, относящаяся к схеме рабочего процесса: список и положение ячеек и ссылок и т.д. Эта информация управляется кэшем, хранящимся в памяти на сервере Asset Manager Web Tier, и обновляется периодически

(по умолчанию - каждые 24 часа). Каждый элемент в кэше управляется индивидуально:

- 1 Когда вы открываете схему рабочего процесса впервые, схема рабочего процесса размещается в кэш и дата ее сохранения записывается.
- 2 При следующем открытии схемы рабочего процесса Asset Manager Web Tier проверяет разницу между текущей датой и датой последнего сохранения схемы рабочего процесса в кэше:
 - Если разница во времени меньше предварительно определенного периода времени, через который должен обновляться кэш, изображение схемы переносится непосредственно из кэша. Это сокращает время, необходимое для отображения информации.
 - Однако если разница во времени больше предварительно определенного периода времени, через который должен обновляться кэш, информация о схеме рабочего процесса обновляется, и новая дата сохранения фиксируется в кэше.

Для изменения частоты обновления кэша, прочитайте руководство Tailoring, раздел Customizing Web clients, глава Modifying the Web client's default behavior, раздел To modify how often the cache managing the workflow schemes and instances is refreshed.

Удаление экземпляров завершенных рабочих процессов

Необходимость удаления завершенных рабочих процессов

Выполняющимися схемами рабочих процессов создаются экземпляры рабочих процессов (в таблице **Экземпляры рабочих процессов** (amWfInstance)).

Данные экземпляры рабочих процессов не ликвидируются автоматически, даже если их выполнение завершено (поле **Статус** (seStatus)).

Если используется большое число схем рабочих процессов, то это, очевидно, приведет к появлению большого числа экземпляров рабочих процессов.

Это без необходимости увеличивает размер базы данных и может ухудшить производительность Asset Manager.

Следовательно, рекомендуется регулярно удалять экземпляры рабочих процессов, выполнение которых завершилось.

Автоматизация удаления завершенных экземпляров рабочих процессов

Чтобы автоматизировать удаление завершенных экземпляров рабочих процессов из рабочей базы данных:

- 1 Добавьте поле в таблицу **Схемы рабочего процесса** (amWfScheme), чтобы определять срок, по истечении которого должны удаляться устаревшие экземпляры рабочих процессов.
- 2 Создайте действие, которым удаляются такие устаревшие экземпляры рабочих процессов.
- 3 Создайте схему рабочего процесса, которой автоматизируется выполнение данного действия.
- 4 Настройте Asset Manager Automated Process Manager для автоматического выполнения такой схемы рабочего процесса.

Добавление поля в таблицу Схемы рабочего процесса

Добавьте следующее поле в таблицу **Схемы рабочего процесса** (amWfScheme):

Параметр	Значение
Имя SQL	AutoCleaningDelay
Метка	Задержка удаления экземпляра
Описание	Срок, по истечении которого выполняется удаление устаревших экземпляров рабочих процессов.
Тип	Продолжительность (временной диапазон)
Создать индекс для этого поля	Не следует использовать этот параметр
Описание (вкладка Справка)	Определяется, сколько времени должно пройти до того, как можно будет удалить устаревшие экземпляры рабочих процессов.
Пример	-1: экземпляры рабочих процессов никогда не удаляются.Положительное или нулевое значение: экземпляры рабочих процессов могут удаляться после истечения крайнего срока.

Информацию о том, как добавить поле в существующую таблицу, можно найти в руководстве Tailoring, глава Customizing the database, раздел Creating new objects, подраздел Creating a field, link or index.

Заполнение поля Задержка удаления экземпляра

Заполните поле **Задержка удаления экземпляра** (AutoCleaningDelay) в каждой используемой схеме рабочего процесса.

Подключение к рабочей базе данных

- 1 Запустите Asset Manager.
- 2 Подключитесь к своей рабочей базе данных.

Создание действия для удаления устаревших экземпляров

- 1 Отобразите список действий с помощью меню

Инструменты/Действия/Изменить.

- 2 Нажмите кнопку **Создать**.
- 3 Заполните следующие поля:

Имя	Имя SQL	Значение
Имя	Name	Удалить завершенные экземпляры
Контекст	ContextTable	Схемы рабочих процессов (amWfScheme)
Тип	seActionType	Скрипт
Имя SQL	SQLName	DeleteFinishedWfInstances
Скрипт действия.	Скрипт	См. (*) ниже.

(*) Скрипт действия:

```
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 ADDSECONDSDT(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 Нажмите кнопку **Создать**.

Создание схемы рабочего процесса для автоматизации выполнения действия

- 1 Отобразите список схем рабочих процессов с помощью меню **Инструменты/Рабочие процессы/Схемы рабочего процесса**.
- 2 Нажмите кнопку **Создать**.
- 3 Заполните следующие поля:

Имя	Имя SQL	Значение
Имя	Name	Удалить завершенные рабочие процессы
Ссылка	Ссылка	ADM_CLEAN_WF_INSTC
Контекст объекта запуска	StartContextTable	Схемы рабочих процессов (amWfScheme)
Группа выполнения	GroupName	Укажите любое имя для автоматизации выполнения схемы рабочего процесса в Asset Manager Automated Process Manager (например, ADMIN).
Единственный активный экземпляр рабочего процесса для объекта	bUniqueActive	Выберите этот параметр.
Не сохранять экземпляры в базе данных	bTransient	Не следует использовать этот параметр

- 4 Нажмите кнопку **Создать**.
- 5 Выберите вкладку **Операции**.
- 6 Щелкните правой кнопкой мыши операцию Запуск и выберите меню **Добавить событие**.
- 7 Заполните следующие поля:

Метка	Имя SQL	Значение
Имя	Имя	Таймер
Тип	seMonitoringType	Периодическое
Повторная инициализация экземпляра рабочего процесса, если один экземпляр уже выполняется.	Периодическое	Не следует использовать этот параметр
Условие AQL	AQLCond	AutoCleaningDelay >= 0

- 8 Выберите и заполните вкладку **Параметры** в соответствии с потребностями.
- 9 Нажмите кнопку **Добавить**.
- 10 Щелкните правой кнопкой мыши и выберите меню **Добавить событие**.

11 Заполните следующие поля:

Метка	Имя SQL	Значение
Имя	Имя	Clean W/F instances
Тип	seType	Автоматическое действие
Записать задачу в журнал	bLogWorkItem	Выберите этот параметр.
Контекст	ContextTable	Схемы рабочих процессов (amWfScheme)
Условие ввода	seInCond	OR

12 Нажмите кнопку **Добавить**.

13 Выберите вкладку **Параметры**.

14 Заполните следующие поля:

Метка	Имя SQL	Значение
Выполнить действия немедленно	bExecImmediately	Выберите этот параметр.
Действия	Действия	Удалить завершенные рабочие процессы

15 Нажмите кнопку **Заккрыть**.

16 Дважды щелкните операцию Очистить экземпляры рабочих процессов.

17 Щелкните правой кнопкой мыши событие Выполнено и выберите меню **Сведения о событии**.

18 Заполните следующие поля:

Метка	Имя SQL	Значение
Обработка	seProcessingMode	Записать событие в журнал и обработать немедленно

19 Нажмите кнопку **Заккрыть**.

20 С помощью мыши создайте связь между операциями Запуск и Очистить экземпляры рабочих процессов.

21 Щелкните правой кнопкой мыши событие Выполнено и выберите меню **Заключительное событие**.

22 Нажмите кнопку **Изменить**.

Настройка Asset Manager Automated Process Manager для автоматического выполнения схемы рабочего процесса

1 Запуск Asset Manager Automated Process Manager.

- 2 Настройте модуль, который будет инициировать выполнение рабочего процесса Удаление завершенных рабочих процессов с помощью меню **Инструменты/Настройка модулей**.

Это модуль с именем Выполнение правил рабочих процессов для группы выполнения X, где X - значение поля **Группа выполнения** (GroupName), определенного на уровне схемы рабочего процесса.

- 3 Оставьте Asset Manager Automated Process Manager активным, если необходимо автоматическое выполнение данного рабочего процесса.

Техническая информация: модель данных

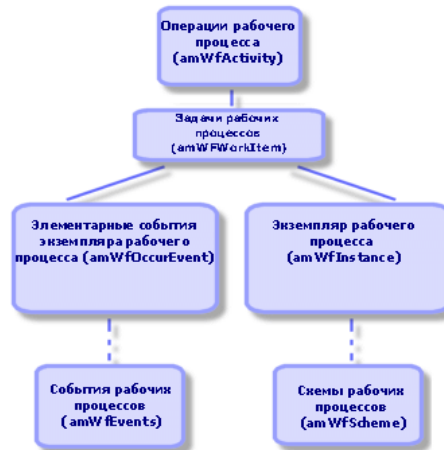
На следующей диаграмме представлены основные таблицы, необходимые для рабочего процесса, и связи между ними. Таблицы указываются их метками и именем SQL:

Рисунок 9.9. Рабочий процесс - основные таблицы, позволяющие определить схему рабочего процесса



На следующей диаграмме представлены основные таблицы, необходимые, когда выполняется экземпляр рабочего процесса:

Рисунок 9.10. Рабочий процесс - основные таблицы, необходимые, когда выполняется экземпляр рабочего процесса



10 Экспорт данных и создание представлений SQL

В данной главе объясняется, как экспортировать данные Asset Manager и управлять представлениями SQL базы данных.

Определения

Скрипты экспорта

Скрипты экспорта позволяют экспортировать данные или (повторно) создавать/удалять представления SQL с помощью Asset Manager Export Tool или приложения amexr.exe. Определенные таким образом скрипты экспорта можно сохранять для их повторного использования.

Скрипт экспорта выполняется либо:

- в «Режиме экспорта» для экспорта данных.
- либо в «Режиме представлений» для (повторного) создания или удаления представлений SQL в базе данных.
- Действия, которые необходимо выполнить для создания/удаления представлений SQL.

Скрипт экспорта выполняется либо:

- в режиме «Экспорт данных» для экспорта данных;
- либо в режиме «Создать/удалить представления SQL» для (повторного) создания или удаления представлений SQL в базе данных.

Запросы экспорта

Запросы экспорта определяются с помощью Asset Manager Export Tool.

Для определения запроса экспорта используются:

- имя;
- конечный файл экспорта (при использовании режима «Экспорт данных»);
- комментарий (который не экспортируется);
- начальная таблица;
- список столбцов, которые необходимо извлечь (поля, ссылки, признаки и вычисляемые поля из начальной таблицы) и соответствующие критерии сортировки;
- фильтр, содержащий оператор WHERE и определяющий условия извлечения;
- фильтр, содержащий оператор HAVING и определяющий условия извлечения;
- Текст запроса (соответствующий вкладкам Фильтр (оператор WHERE) и «Оператор HAVING»).
- вкладка предварительного просмотра.

Экспорт данных из базы данных Asset Manager

Можно экспортировать данные из базы данных Asset Manager в текстовые файлы:

- с помощью скрипта экспорта;
- с помощью пункта контекстного меню Сервис/Экспорт списка. Данный пункт меню отображается при отображении, по крайней мере, одного списка или списка вкладки Он позволяет экспортировать активный список.

Экспорт данных с помощью скрипта экспорта

- 1 Запустите Asset Manager Export Tool с помощью меню Запуск или из группы программ Asset Manager.
- 2 Определите скрипт экспорта, для которого установлен режим «Экспорт данных»:
 - 1 На вкладке Запросы запишите запросы, определяющие данные для экспорта.
 - 2 На вкладке Форматирование укажите формат текстовых файлов, в которые будут экспортироваться данные.
 - 3 С помощью пунктов меню Файл/Сохранить скрипт или Файл/Сохранить скрипт как сохраните скрипт.

- 3 Выполните скрипт экспорта:
 - или непосредственно в Asset Manager Export Tool с помощью <Execute script> ;
 - или запустив приложение amexpl.exe в DOS.



ЗАМЕЧАНИЕ:

Чтобы сохранить согласованность ограничений доступа, определенных в Asset Manager, Asset Manager Export Tool или приложение amexpl.exe могут запускаться только администратором (имя для входа Admin) или пользователем с административными правами.

Экспорт данных с помощью пункта контекстного меню Экспорт списка

Пункт контекстного меню Экспорт списка доступен для всех пользователей Asset Manager; он позволяет пользователям экспортировать данные, которые им разрешено просматривать.

- 1 Отобразите список, который необходимо экспортировать (основной список или список вкладки). Если на экране отображаются несколько списков, удостоверьтесь, что активен тот список, который необходимо экспортировать.
- 2 Выберите пункт контекстного меню Экспортировать список.
- 3 Заполните появляющееся окно, затем щелкните кнопку **Экспорт**.



ЗАМЕЧАНИЕ:

Дополнительную информацию о пункте меню **Файл/Экспорт** можно найти в руководстве Интерфейс пользователя, глава «Начало работы с Asset Manager», раздел «Списки записей», подраздел «Экспорт списка».

Управление представлениями SQL в базе данных Asset Manager

Asset Manager Export Tool позволяет создавать, вновь создавать или удалять представления SQL в базе данных Asset Manager. Затем внешними инструментами такие представления могут использоваться вместо текстовых файлов.



ЗАМЕЧАНИЕ:

Предостережение: представления SQL, которые скрипты экспорта позволяют создавать, изменять или удалять, отличаются от представлений в смысле терминологии Asset Manager. Представление SQL эквивалентно инструкции SQL CREATE VIEW.

Чтобы создать, вновь создать или удалить представления SQL в базе данных Asset Manager:

- 1 Запустите Asset Manager Export Tool.
- 2 Определите скрипт экспорта, для которого установлен режим «Создать/удалить представления SQL»:
 - 1 На вкладке Запросы запишите запросы, определяющие данные для извлечения.
 - 2 На вкладке Представления укажите действия, которые необходимо выполнить: создать, изменить или удалить представления, непосредственно исполнить получающийся в результате скрипт SQL или сохранить файл.
 - 3 Сохраните скрипт экспорта.
- 3 Выполните скрипт экспорта:
 - или непосредственно в Asset Manager Export Tool;
 - или запустив приложение amexpl.exe.

Рекомендации

Не рекомендуется использовать в качестве ключей согласования поля «Id» таблиц, если необходимо повторно импортировать данные, которые были ранее экспортированы. Фактически, соответствующие идентификационные номера не являются фиксированными и могут быть изменены. Используйте ключи, значения которых являются «неизменными», например, «Метка актива» для активов.

Определение скрипта экспорта

Чтобы экспортировать данные или создать представления SQL для базы данных, необходимо определить скрипты экспорта и экспортировать запросы, которые они создают. Для этого используется Asset Manager Export Tool.

Скрипт экспорта выполняется либо:

- в «Режиме экспорта» для экспорта данных;
- либо в «Режиме представлений» для (повторного) создания или удаления представлений SQL в базе данных.

Действия, которые необходимо выполнить для создания/удаления представлений SQL.

Скрипт экспорта выполняется либо:

- в режиме «Экспорт данных» для экспорта данных.
- Либо в режиме «Создать/удалить представления SQL» для (повторного) создания или удаления представлений SQL в базе данных.

В данном разделе объясняется создание скрипта действия:

- Методология
- Определение запросов экспорта
- Формат вывода скрипта экспорта
- Действия, затрагивающие представления SQL

Методология

Чтобы создать или изменить скрипт экспорта:

- 1 Запустите Asset Manager Export Tool.
- 2 Откройте соответствующую базу данных. Предостережение: можно подключаться только с именем входа Admin или с именем входа с административными правами.
- 3 Создайте новый скрипт с помощью пункта меню Файл/Создать скрипт или откройте скрипт, измененный с помощью пункта меню Файл/Открыть скрипт.
- 4 В верхней части экрана Asset Manager Export Tool, что необходимо: экспортировать данные (режим «Экспорт данных») или управлять представлениями SQL (режим «Представления») базы данных.
- 5 Запишите запросы скрипта экспорта на вкладке Запросы.
- 6 Если выбран экспорт данных, укажите формат вывода экспортируемых данных на вкладке Форматирование.
- 7 Если необходимо управлять представлениями SQL, укажите, что необходимо делать, на вкладке Представления.
- 8 Сохраните скрипт с помощью пунктов меню Файл/Сохранить скрипт или Файл/Сохранить скрипт как.

Определение запросов экспорта

Можно определять запросы скрипта экспорта на вкладке Запросы в Asset Manager Export Tool.

- Щелкните кнопку **Создать**, чтобы добавить запрос экспорта.

- Щелкните кнопку **Удалить**, чтобы удалить выбранный запрос экспорта.

Создание запроса в скрипте экспорта

- 1 Щелкните кнопку **Создать** на вкладке Запросы.
- 2 Определите имя запроса. Данное имя используется в журнале выполнения на вкладке Сообщения сведений о скрипте экспорта.
- 3 Можно ввести комментарии (которые не экспортируются).
- 4 Определите данные, которые необходимо экспортировать, в поле Запрос.
- 5 Если необходимо экспортировать данные, а не создавать, изменять или удалять представления, то в поле Файл укажите путь и имя текстового файла вывода, в который будут записываться экспортируемые данные, выбранные запросом. Таким образом, скриптом экспорта, содержащим несколько запросов экспорта, будут создаваться несколько текстовых файлов.



ЗАМЕЧАНИЕ:

Поле Файл не отображается, если выбран параметр Создать/удалить представления SQL.

Извлекаемые данные

Чтобы указать извлекаемые данные, заполните поле Запрос в сведениях о запросе скрипта экспорта. Запросы применяются относительно таблицы в базе данных Asset Manager.

Можно ввести запросы напрямую или щелкнуть кнопку , чтобы вызвать окно, которое поможет определить запрос:

Экспортируемые столбцы и порядок сортировки

На вкладке Столбцы и сортировка (сведения о запросе) определяются список полей, ссылок, признаков и вычисляемые поля для экспорта, а также соответствующие критерии сортировки.

Выберите по одному все поля, ссылки, признаки и вычисляемые поля, используемые для экспорта, из списка слева и щелкните стрелку, чтобы вставить их в список справа.

Для каждой строки в списке справа:

- Установите флажок Видимость, чтобы экспортировать столбец. Если флажок Видимость не установлен, столбец не экспортируется (он может использоваться, однако, для сортировки экспортируемых данных и так далее).

- Установите флажок Группировка по, чтобы группировать данные по полям, соответствующим столбцам. Это эквивалентно добавлению оператора GROUP BY <Имя поля> к запросу SQL.

Например:

```
SELECT Brand, Count(IModelId) FROM amModel GROUP BY Brand ORDER BY Brand
```



ВНИМАНИЕ:

Если установлен флажок Группировка по, к запросу прикрепляется оператор GROUP BY, но для того, чтобы запрос был действителен, необходимо также добавить соответствующие групповые функции в операторе SELECT.

Определите порядок сортировки для экспортируемых данных:

- 1 Чтобы определить сортировку по индексу, выберите индекс в поле Сортировать по индексу.
- 2 Или установите соответствующие флажки Сортировать в необходимом порядке сортировки.



ЗАМЕЧАНИЕ:

Можно установить флажок Выполнить принудительное индексирование, чтобы принудительно использовать индексы, указанные в запросе.

Дополнительную информацию можно найти в главе Запросы AQL, раздел Сортировка и индексы, данного руководства.

Если установить флажок Только уникальные записи, абсолютно идентичные строки экспортируются только раз. Это эквивалентно добавлению оператора DISTINCT к запросу SQL.

Пример без установки флажка Только уникальные записи:

```
SELECT Brand FROM amModel
```

Пример с установкой флажка Только уникальные записи:

```
SELECT DISTINCT Brand FROM amModel
```

Фильтры

Можно определить два типа фильтров для выбора извлекаемых данных:

- запрос AQL, использующий оператор WHERE, на вкладке Фильтр (оператор WHERE);
- запрос AQL, использующий оператор HAVING, на вкладке оператор HAVING.

Отображение запроса

Запрос AQL, определяемый на вкладках Столбцы и сортировка, Фильтр (оператор WHERE) и Оператор HAVING, отображается на вкладке Запросы.

Предварительный просмотр результатов запроса

Можно проверить запрос и просмотреть его в синтаксисе языка SQL на вкладке Предварительный просмотр.

Просто щелкните кнопку  для предварительного просмотра результатов запроса в виде списка записей. Обратите внимание, что Asset Manager отображает число записей, соответствующих запросу, справа внизу окна.

Формат вывода скрипта экспорта

Если выбрать функцию Экспорт данных, можно определить формат текстовых файлов вывода на вкладке Форматирование. Данный формат применяется для всех запросов экспорта.



ЗАМЕЧАНИЕ:

Вкладка Форматирование не отображается, если выбрано удаление, создание или повторное создание представлений.

Заголовок столбца

Выберите значение, если необходимо, чтобы первая строка файла экспорта включала:

- псевдоним столбцов, указанный в запросе экспорта;
- имена SQL полей или ссылки, соответствующие столбцам;
- описание полей или ссылки, соответствующие столбцам.

Разделитель столбцов

Такой разделитель вставляется между данными в каждом столбце.

Код текста

Между символами такого кода помещаются текстовые строки. Если используется символ ', любой символ ' при экспорте будет выводиться как ". И наоборот - для символа ".

Набор символов

Данный параметр позволяет выбирать набор символов или ANSI, OEM(DOS), UTF-8, UNICODE, или Latin 1.

Десятичный разделитель

Данный символ используется для отделения дробной десятичной части экспортируемых чисел.

Разделитель даты

Данный символ вставляется между днем, месяцем и годом экспортируемых дат.

Формат даты

Форматом даты определяется порядок, в котором экспортируются, дни (ДД), месяцы (ММ) и годы (ГГ).

Формат года

Определяется, как будет экспортироваться номер года, с 2 или 4 цифрами.

Разделитель времени

Данный символ вставляется между часами, минутами и секундами.

Отображать секунды

Указывается, необходимо ли показывать секунды в экспортируемых значениях времени.

Действия, затрагивающие представления SQL

Если необходимо удалить или (вновь) создать представления SQL, соответствующие запросам экспорта, для определения действий, подлежащих выполнению, можно использовать вкладку Представления.



ЗАМЕЧАНИЕ:

Вкладка Представления не отображается, если выбран параметр Экспорт данных.

Выберите в рамке «Действия» одно из действий для выполнения:

- создать или восстановить представления;
- удалить представления.

В разделе «Скрипт управления представлениями SQL» укажите, как необходимо обрабатывать запрос (вкладка Запросы, рамка Действия):

- Чтобы создавать или вновь создавать представления SQL непосредственно при выполнении скрипта экспорта, выберите параметр Выполнить SQL напрямую.

- Чтобы создать скрипт представления SQL для создания представления (инструкция CREATE VIEW) или удаления представления (инструкция DROP VIEW), выберите параметр Сохранить код SQL в файле, затем:
 - 1 Щелкните кнопку , чтобы указать имя и путь файла.
 - 2 Укажите разделитель инструкций SQL: «;» (для Oracle) или «GO» (для всех остальных СУБД).

Выполнение скрипта экспорта

Скрипты экспорта можно использовать для экспорта данных или управления представлениями SQL.

В данном разделе описываются два метода выполнения скрипта экспорта:

- выполнение скрипта экспорта в Asset Manager Export Tool;
- выполнение скрипта экспорта в DOS.

Выполнение скрипта экспорта в Asset Manager Export Tool

Чтобы выполнить скрипт экспорта в Asset Manager Export Tool:

- 1 Запустите Asset Manager Export Tool.
- 2 Определите скрипт экспорта и сохраните его.
- 3 Затем выполните данный скрипт:
 - или с помощью пункта меню Действия/Выполнение скрипта;
 - или нажав F8;
 - или щелкнув кнопку .

Информация о выполнении процесса экспорта отображается на вкладке Сообщения.

Если процесс экспорта завершается успешно, последнее отображаемое сообщение - «Скрипт успешно выполнен». Если происходит ошибка, отображается следующее сообщение: «Ошибка при выполнении скрипта».

Перед всеми сообщениями отображается соответствующий значок:

-  Общая информация.
-  Ошибка.
-  Экспорт успешный.
-  Внимание.

Выполнение скрипта экспорта в DOS

Как это работает

Для «интерактивного» выполнения программного обеспечения DOS необходимо сначала создать скрипт экспорта с помощью Asset Manager Export Tool.

Можно выполнить команду экспорта, вручную или автоматически (с помощью пакетного файла, например), с помощью программы amexp.exe в подпапке bin папки установки Asset Manager.

Синтаксис

```
amexpl [-verbose] [-?|h|H] -script:<скрипт> -cnx:<cnx> [-login:<имя входа>] [-password:<пароль>]
```

-verbose: отображение сообщений в процессе экспорта.

-, -h или -H: отображает сообщения справки.

-script: путь и имя файла экспорта для выполнения.

-cnx: имя подключения к базе данных Asset Manager (как оно отображается в элементе меню Файл/ Управление подключениями).

-login: имя для входа администратора базы данных (Admin или пользователя с административными правами).

-password: пароль для имени входа.

Строки между <> не могут включать пробелы.

Например:

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


11 Скрипты

В данной главе объясняется, как использовать скрипты.

Определение скрипта

Обзор

Слово скрипт обычно означает программу, написанную на языке высокого уровня. В Asset Manager в данное понятие включаются три типа скриптов:

- Процедурные скрипты, которые включают:
 - скрипты расчета, написанные на языке Basic, используемые для расчета значений полей, которыми определяются свойства объектов в базе данных Asset Manager и так далее;
 - скрипты на языке Basic, которыми выполняются задачи, особенно в действиях.



ЗАМЕЧАНИЕ:

Такие программы на языке Basic могут включать функции. Предметом рассмотрения данной главы является такой тип скриптов.

- Декларативные скрипты. Это скрипты импорта и экспорта, в которых используется свой собственный язык для написания скриптов, отличный

от языка Basic. Данный тип скриптов подробно документирован в руководстве Администрирование, глава Импорт данных, и в главе Экспорт данных и управление представлениями SQL данного руководства.

- «Смешанные» - декларативные и процедурные. Такой тип скриптов используется в мастерах в Asset Manager.

Информация об используемой версии языка Basic

Версия языка Basic, используемая в Asset Manager, представляет собой сокращенную версию, разработанную в Cypress и совместимую с версией Visual Basic for Applications™. Дополнительную информацию о данном языке, его структуре и синтаксисе можно найти в документации по Basic.

Поддерживаются только некоторые функции языка Visual Basic for Applications, например:

- не поддерживаются функции доступа к файлам;
- имеется ограниченная поддержка функций даты и времени;
- недоступны элементы управления языка Visual Basic for Applications.



ЗАМЕЧАНИЕ:

Чтобы получить справку по функции Basic или по ключевому слову в руководстве «Programmer's Reference», поместите курсор на какое-либо слово и нажмите клавишу F1: отобразится контекстная справка.

Нотация доступа к данным

Синтаксис Basic, используемый в Asset Manager, аналогичен стандартному синтаксису, кроме функций доступа к данным из текущей записи; для этого используется следующий формат:

[Ссылка.Ссылка.Поле]

Пример из таблицы моделей:

[Category.FullName]



ЗАМЕЧАНИЕ:

Для восстановления номера кода ссылки можно использовать следующий синтаксис:

[Ссылка.Ссылка]

Если необходимо сослаться на ссылку, можно использовать или имя SQL ссылки, или имя ее ключа.

Пример:

RetVal=[Контакт.Местоположение] или RetVal=[Контакт.ILocald]
--

В обоих примерах возвращается один и тот же результат, код ссылки.

Применение скриптов

Asset Manager позволяет связывать со скриптами на языке Basic следующие свойства:

- для конфигурирования значений полей по умолчанию (команда **Настройка объекта** в контекстном меню);
- для значения по умолчанию признака, связанного с таблицей;
- в вычисляемых полях типа Basic;
- для конфигурирования полей (команда **Настройка объекта** в контекстном меню или Asset Manager Application Designer):
 - Значение по умолчанию;
 - Обязательный вид;
 - История;
 - Только чтение;
- для параметров признака, связанного с таблицей:
 - Значение по умолчанию (имя SQL: DefValScript);
 - Доступно (имя SQL: seAvailable);
 - Принудительное отображение (имя SQL: seForceDisplay);
 - Обязательный (имя SQL: seMandatory);
 - Сохранять историю (имя SQL: seKeepHistory);
- для действий типа Скрипт:
 - Скрипт действия (имя SQL: Script) для действия типа Скрипт;
- в мастерах:
 - запуск и окончание скриптов мастеров;
 - скрипты для определения свойств узлов;
- в вычисляемых полях типа Basic.
- в рабочих процессах:
 - для операций рабочего потока типа Тест/скрипт;
 - для событий рабочих потоков Базовое;
 - для назначенных рабочих процессов Рассчитанное лицо в таблице amWfOrghole.

Введение в функции

Данный раздел включает информацию по следующим темам:

- Определение функции
- Встроенные функции и программируемые функции
- Типы функций и параметров

Определение функции

Функция является программой, которая выполняет операции и возвращает пользователю значение. Это значение называется «возвращаемым значением» или «кодом возврата».

Функции имеют следующую структуру:

```
Function <Имя функции> (<Параметр> As <Тип параметра>[, ..., <Параметр>
As <Тип параметра>]) As <Тип функции>

<Программа (скрипт), выполняемая функцией. Данная программа должна о
пределять возвращаемое значение.>

End Function

End Function
```

Такая структура применяется и для встроенных функций, и для программируемых функций.

Встроенные функции и программируемые функции

Встроенные функции и программируемые функции представляют собой две главные категории функций, доступные в Asset Manager.

Встроенные функции

Встроенные функции подобны позициям программного обеспечения, уже написанным для пользователя. Такими позициями программного обеспечения выполняются задачи всех типов (расчеты, преобразования данных, предоставленных пользователем) и возвращается результат. Пользователь просто вызывает функцию по ее имени и предоставляет любую информацию, необходимую для возврата результата. Такие элементы информации, предоставляемые пользователем, называются «параметрами».

Например, функцией AmConvertCurrency() сумма в валюте A преобразуется в сумму в валюте B, используя валютный курс, определенный на указанную дату. В данном примере:

- имя функции - **AmConvertCurrency**;
- параметры, которые должен предоставить пользователь:
 - валюта A;
 - валюта B;
 - сумма для преобразования;
 - дата, когда будет выполняться преобразование (для указания используемого валютного курса).

Данной функцией выполняется преобразование и возвращается значение, соответствующее результату преобразования.

Программируемые функции

Программируемые функции - позиции программного обеспечения, которые пользователь может написать самостоятельно. Пользователь должен в явной форме определить значение, которое должно возвращаться (также называемое «возвращаемым значением»), в переменной RetVal с помощью программируемой функции в следующем формате:

RetVal=<Expression>

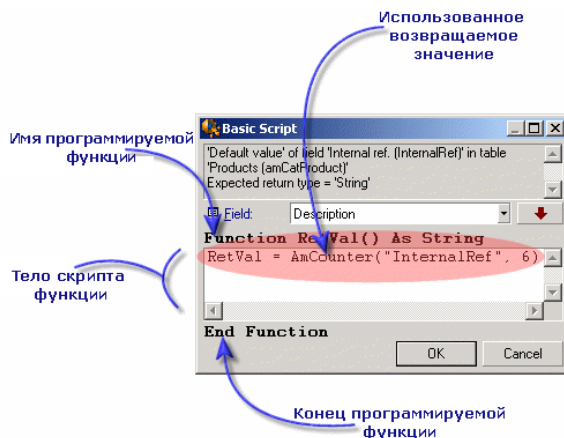


ЗАМЕЧАНИЕ:

Asset Manager отказывается компилировать скрипт функции, для которой не определено возвращаемое значение.

Программируемые функции доступны через составитель скриптов (по щелчку кнопки  в поле, для которого возможно написание скриптов). Составитель скриптов предназначен для того, чтобы помочь пользователям в создании позиций программного обеспечения, соответствующих тем или иным функциям. В построитель скриптов включается шаблон для написания программируемых функций:

Рисунок 11.1. Составитель скриптов



Описание программируемой функции доступно наверху окна составителя скриптов. В нем указывается объект, имеющий отношение к функции (например, значение по умолчанию поля Штрих-код (имя SQL: BarCode) в таблице активов), а также код ожидаемого типа возврата (в предыдущем примере: String).

Типы функций и параметров

Типы функций

Тип встроенной функции - тип значения, возвращаемого функцией. Необходимо особо обращать внимание на данное обстоятельство, потому что это может вызвать ошибки компиляции и выполнения в скриптах на языке Basic.

Например, нельзя использовать функцию, которой возвращается значение одного типа, при определении значения по умолчанию для поля другого типа. Например, пусть для любого поля типа «Дата» или «Дата и время» назначен следующий скрипт по умолчанию:

```
RetVal=AmLoginName()
```

Функцией **AmLoginName()** возвращается имя подключившегося пользователя в форме строки символов (тип «Строка»). Следовательно, возвращаемое значение имеет формат, несовместимый с форматом поля «Дата», и Asset Manager отобразит сообщение об ошибке при создании в следующий раз записи в той же таблице.

Типы параметров

Параметры, используемые во встроенных функциях, также имеют тип; необходимо учитывать такие типы, чтобы функции правильно выполнялись. В случае ошибки в типе параметра Asset Manager отображает сообщение об ошибке при выполнении функции.

Список типов

В следующей таблице приводится сводка различных типов, доступных для функции или параметра:

Таблица 11.1. Функции/параметры - типы

Тип	Описание
Integer	Целое число от -32 768 до +32 767.
Long	Целое число от -2 147 483 647 до +2 147 483 646.
Double	8-байтное число с плавающей точкой.
String	Текст, в котором допустимы все символы.
Дата	Дата или Дата+Время.
Variant	Простой тип, который может представлять любой тип.

Определение типа возврата программируемой функции

Перед изменением скрипта необходимо определить соответствующую функцию и ее тип. Такая информация отображается полужирным шрифтом во всех окнах «Скрипт на языке Basic» в следующем формате:

Function <Имя функции>() As <Тип функции>

- Три самых часто используемых типа функций - Boolean, Integer и String:
- функциями типа Boolean возвращаются значения TRUE или FALSE; любые другие значения вызывают ошибку при компиляции скрипта на языке Basic;
 - Функция целого числа выдает только значения, выраженные целыми числами (например, 0, 1, 8, 12).
 - функциями типа "String" возвращаются только строки символов (например, "Building21") в кавычках.

 **ЗАМЕЧАНИЕ:**

Если не учитывать тип функции, можно получить ошибки при компиляции программы на языке Basic. Следует всегда обращать внимание на тип используемой функции.

Имя и тип функции позволяют определить код возврата, который необходимо использовать в скрипте, в следующем формате:

RetVal=<Выражение, соответствующее типу функции>
--

Классификация функций Basic

В скриптах используются различные классы функций Basic:

- обычные функции Basic, соответствующие стандарту Visual Basic for Applications TM;
- общие функции, специфичные для Asset Manager, которые могут использоваться везде, где используются скрипты;
- специальные функции, которые можно использовать в определенных областях Asset Manager.

Первые шаги в написании скриптов

В данном разделе рассматривается функционирование скриптов и в него включен пример сценария:

- пример сценария;
- шаг 1 - создание признака «Руководство»;
- шаг 2 - открытие окна редактирования;
- шаг 3 - анализ и определение алгоритма;
- шаг 4 - написание скрипта Basic;
- шаг 5 - проверка скрипта Basic.

Пример сценария

Цель

Обеспечить доступность признака «Руководство» только для модели «Компьютер/материнская плата/» и ее потомков.

Метод

Прикрепление скрипта на языке Basic к параметру **Доступно** (имя SQL: seAvailable) признака «Руководство».

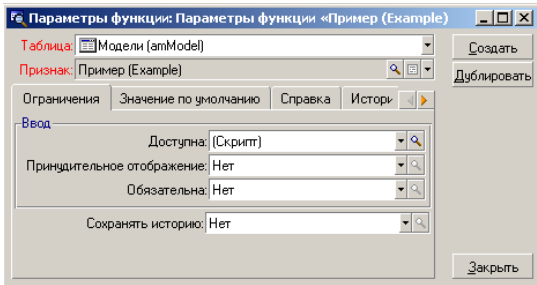
Шаг 1: создание признака «Руководство»

Выберите пункт меню **Администрирование/Признаки**. Нажмите кнопку **Создать** для создания нового признака. Заполните данный признак, как показано ниже:

Имя поля	Значение
Заголовок (Имя SQL: TextLabel)	«Руководство»
Имя SQL (имя SQL: SQLName)	«Руководство»
Тип ввода (SQL-имя: seDataType)	Числовые

Нажмите кнопку **Создать** для создания признака.

Перейдите на вкладку **Параметры** и нажмите кнопку  для изменения параметров признака «Руководство». Заполните вкладку **Ограничения**, как показано ниже.



Шаг 2: открытие окна редактирования

На вкладке **Ограничения** установите для параметра **Доступно** (имя SQL: seAvailable) значение «(Script)». Щелкните кнопку увеличения . Asset Manager открывает окно редактирования скрипта:

Шаг 3: анализ и определение алгоритма

Алгоритм должен выполнять следующие задачи:

- Установите для поля **Доступно** (имя SQL: seAvailable) значение Да, если модель - «/Компьютер/материнская плата/» или один из потомков.
- Установите для поля **Доступно** значение Нет во всех остальных случаях.

Поэтому наш алгоритм будет таким

Если полное имя модели начинается с «/Компьютер/материнская плата», то признак доступен, в противном случае признак недоступен

Поэтому значением поля **Полное имя** (имя SQL: FullName) в таблице моделей определяется значение поля **Доступно** признака. Только это поле указывается в нашем алгоритме.

Щелкните кнопку увеличения  около поля **Доступно**, чтобы начать редактирование скрипта на языке Basic. Раскрывающийся список в окне редактирования позволяет выбрать поле **Полное имя** (имя SQL: FullName) из таблицы моделей.

После того, как поле выбрано, перенесите его в окно редактирования, нажав кнопку .

Шаг 4: написание скрипта Basic

Теперь необходимо написать алгоритм из шага 1 на языке Basic в окне редактирования, предназначенном для данной задачи.

```
If Left([FullName], Len("/Computer/Motherboard/"))="/Computer/Motherboard/" Then RetVal=1 Else RetVal=0 End If
```



ЗАМЕЧАНИЕ:

В скриптах регистр не учитывается

Для подтверждения скрипта нажмите кнопку **ОК**.

Шаг 5: проверка скрипта Basic

Это шаг позволяет убедиться, что скрипт функционирует правильно.

- 1 Откройте таблицу моделей, выбрав пункт меню Портфель/Модели.
Нажмите кнопку **Создать** для создания новой модели.
- 2 Заполните только обязательные поля.
 - 1 **Имя**
 - 2 **Является подмоделью** (имя SQL: Parent) - «Компьютер/материнская плата».
 - 3 **Вид** (имя SQL: Nature).
 - 4 **Штрих-код** (имя SQL: BarCode).
- 3 Нажмите кнопку **Создать** для создания своей новой модели.

- 4 Выберите вкладку **Признаки** и щелкните кнопку , чтобы добавить признак. На экране выбора показывается имя признака, для которого редактировался скрипт.
 - 5 Измените значение поля **Является подмоделью** на «/Компьютер/» и утвердите данное изменение, щелкнув **Изменить**.
 - 6 Выберите в сведениях о модели вкладку **Признаки** и щелкните кнопку , чтобы добавить признак. На экране выбора больше не показывается имя признака, для которого редактировался скрипт.
- Скрипт правильно выполняет свою функцию.

Библиотеки скриптов

Asset Manager позволяет сохранять библиотеки скриптов, чтобы централизовать доступ к данным скриптам.

Библиотеку скриптов можно открыть с помощью пункта меню

Администрирование/Библиотеки скриптов.

Записанные библиотеки скриптов вызываются командой интерфейса API amEvalScript.

Дополнительные сведения об интерфейсе API amEvalScript можно найти в руководстве «Programmer's Reference», раздел «Alphabetic reference» («Алфавитный указатель»).

Концепции

В Asset Manager скриптом определяется функция.

Создание библиотеки скриптов, таким образом, означает определение набора функций.

Создание библиотеки скриптов

Чтобы создать библиотеку скриптов:

- 1 Откройте список библиотек скриптов.
- 2 Занесите в поле **Имя** имя своей библиотеки.
- 3 В поле **Скрипт** введите свой скрипт.
- 4 Проверьте свой скрипт, нажав на кнопку **Создать**.

Например, создайте библиотеку lib, введя следующий скрипт:

```
function FullName(strName As String, strFirstName As String) As String
FullName = strFirstName & ", " & strName
end function
```

Данной функцией возвращается строка, состоящая из имен и фамилий сотрудников.



ВНИМАНИЕ:

Создаваемые функции должны иметь разные имена в пределах создаваемого набора библиотек скриптов.

Вызов скрипта из библиотеки скриптов

Чтобы вызвать скрипт из библиотеки, необходимо определить несколько параметров: имя библиотеки, функцию, определяемую в скрипте, и параметры, связанные с данной функцией.

Например, создайте действие типа «Скрипт», callEvalScript, которое будет использовать предварительно созданную библиотеку.

- 1 Заполнит поле **Контекст** (имя SQL: ContextTable) данными таблицы **Сотрудники и подразделения** (имя SQL: amEmplDept).
- 2 Введите следующий скрипт на вкладке **Скрипт**:

```
Dim strFullName As String
strFullName = amEvalScript("biblio", "FullName", "", [Name], [FirstName])
amMsgBox (strFullName)
```

Данным скриптом вызывается функция FullName из библиотеки lib и отображаются в диалогом окне имя и фамилия сотрудника.

- 3 Проверьте, нажав на кнопку **Создать**.



ЗАМЕЧАНИЕ:

Контекстный параметр, обычно используемый с интерфейсом API amEvalScript, не используется в случае вызова библиотеки скриптов.

Советы и предупреждения

В данный раздел включены несколько советов, которые могут помочь при написании скриптов.

Меры предосторожности при использовании программируемых функций

Ниже указаны несколько мер предосторожности, которые необходимы иметь в виду при написании своих скриптов.

- Предназначение программируемых функций, например, тех, которыми определяется значение по умолчанию поля или ссылки, - установка возвращаемых значений функций. Поэтому настоятельно не рекомендуется выполнять другие операции в программируемых функциях. В самом лучшем случае может произойти значительное падение общей производительности, а в самом худшем случае - можно повредить свою базу данных.
- Программируемые функции широко используются в Asset Manager. По мере возможности, следует оптимизировать скрипты, чтобы сохранять общую производительность Asset Manager.

Формат констант «Дата+время» в скриптах

Даты, на которые выполняются ссылки в скриптах, выражаются в международном формате, независимо от параметров отображения, определенных пользователем:

yyyy/mm/dd hh:mm:ss

Пример:

```
RetVal="2001/07/12 13:05:00"
```

ЗАМЕЧАНИЕ:

Можно также в качестве разделителя даты использовать дефис («-»).

Дата в Basic

В языке Basic дата может выражаться в международном формате или как «число с двойной точностью». В этом случае целая часть числа представляет собой число дней, прошедших с полуночи 30 декабря 1899 г., а дробная часть представляет долю текущей даты (число секунд, прошедших с начала дня, разделенное на 86400).

Формат констант «Продолжительность» в скриптах

В скриптах значения длительности сохраняются и выражаются в секундах. Например, чтобы установить для поля типа «Продолжительность» значение по умолчанию 3 дня, используется следующий скрипт:

```
RetVal=259200
```

Аналогично функции, которые рассчитывают продолжительность, например, функция **AmWorkTimeSpanBetween()**, возвращают число секунд.



ЗАМЕЧАНИЕ:

При выполнении преобразований в Asset Manager продолжительность года принимается как 12 месяцев по 30 дней (таким образом, 1 год = 360 дней).

Доступ для считывания и для записи к системному подробному списку

Asset Manager управляет системными подробными списками, назначая целые значения для любых возможных значений в подробном списке.

Рассмотрим пример подробного списка, используемого для заполнения поля **Назначение** (имя SQL: seAssignment) в области **Стандартное назначение** вкладки **Общее** сведений об активе.

В приведенной ниже таблице приводится сводка значений, используемых в таком подробном списке.

Значение в подробном списке	Связанное целое
Используется	0
В запасе	1
Выбывший (или израсходованный)	2
Ожидает доставки	3

Таким образом, чтобы определить значение по умолчанию подробного списка, необходимо:

- 1 Указать целое число, подходящее для соответствующего значения.
- 2 Отредактировать следующую строку:

RetVal=<Целое число, связанное с соответствующим значением>

В данном примере, если необходимо установить для системного подробного списка, используемого в поле **Назначение**, значение по умолчанию Ожидает доставки, необходимо отредактировать строку следующим образом:

RetVal=3



ЗАМЕЧАНИЕ:

Не следует путать системный подробный список с определенным пользователем закрытым подробным списком.

 ЗАМЕЧАНИЕ:

Полный список значений системных подробных списков включен в файл `database.txt`, который можно найти в подпапке `doc\infos` папки установки Asset Manager. В двух столбцах, **DATA DISPLAY AND ENTRY TYPE** и **ADDITIONAL INFORMATION ON DATA DISPLAY AND ENTRY TYPE**, описываются тип подробного списка и значения, используемые в подробном списке, соответственно.

Виртуальная ссылка **CurrentUser**

Определение

CurrentUser можно рассматривать как ссылку, которая находится во всех таблицах и указывает на запись в таблице подразделений и сотрудников, соответствующую текущему пользователю.

- В формате **CurrentUser** она указывает на запись, соответствующую текущему пользователю, и возвращает строку описания из таблицы «Сотрудники и подразделения».
- В формате «**CurrentUser**.<имя SQL поля>» возвращается значение поля для текущего пользователя.

 ЗАМЕЧАНИЕ:

Такая виртуальная ссылка не отображается в списке полей и ссылок, поэтому она недоступна составителю скриптов непосредственно. Следует ввести это выражение вручную.

Эквиваленты

Функции **AmLoginName()** и **AmLoginId()**, которыми возвращаются имя и код текущего пользователя соответственно, можно рассматривать как функции, производные от **CurrentUser**. В действительности следующее является эквивалентным:

- `AmLoginName()=[CurrentUser.Name]`
- `AmLoginId()=[CurrentUser.IPersId]`

Ограничения

CurrentUser будет работать только в том случае, когда определен контекст (контекстом является таблица).

Если контекст отсутствует, следует использовать другую функцию.

Пример:

пусть необходимо создать неконтекстное действие для выполнения файла, путь которого зависит от пользователя, подключенного к базе данных Asset Manager.

Если бы действие было контекстным, можно было бы создать действие типа Исполняемое с помощью поля **Папка** (Folder) с установленным, например, значением c:\scripts\[CurrentUser.Name]\.

Однако если действие типа Исполняемое не имеет контекста, выражение [CurrentUser.Name] рассматривается как постоянный текст.

Поэтому необходимо найти другое решение, например, создание неконтекстного действия типа Скрипт следующим образом:

```
RetVal = amActionExec("program.exe","c:\scripts\" + amLoginName())
```

Виртуальная ссылка «Старая»

Данная ссылка используется в скриптах, относящихся к используемым в рабочих процессах действиям.

Определение

Ссылка «Старая» позволяет ссылаться на контекст рабочего процесса до изменения записи.

Ограничение

Данная ссылка может использоваться только в скриптах, относящихся к используемым в рабочих процессах действиям.

Синтаксис

Чтобы получить значение поля до обновления данного поля рабочим процессом, используется следующий синтаксис:

```
[Old.Field]
```

Комментарии в скрипте на языке Basic

Часто оказывается полезным вводить комментарии в скрипт на языке Basic, чтобы указать в явной форме, что выполняется, или позволить пользователю понимать и изменять скрипт. В Asset Manager обеспечивается возможность вставлять комментарии в тело скрипта с помощью символа апострофа ('). Все символы, следующие после символа одинарной прямой кавычки на той же строке игнорируются компилятором, который интерпретирует их как комментарий. Возможны две ситуации.

- Или комментарий имеет свою собственную строку в скрипте на языке Basic, как показано ниже.

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

- Или комментарий добавляется к концу строки, которая должна интерпретироваться компилятором Basic.

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

Инициирование сообщения об ошибке

Можно инициировать сообщение об ошибке преднамеренно с помощью функции Err.Raise. Используется следующий синтаксис:

```
Err.Raise (<Номер ошибки>, <Сообщение об ошибке>)
```



ЗАМЕЧАНИЕ:

Если создание или изменение записи оказывается недействительным, потому что затрагивается значение поля «Действительность» таблицы, рекомендуется инициировать сообщение об ошибке с помощью функции Err.Raise, чтобы предупредить пользователя. Без такого сообщения об ошибке пользователь необязательно сможет понять, почему нельзя создать или изменить запись.

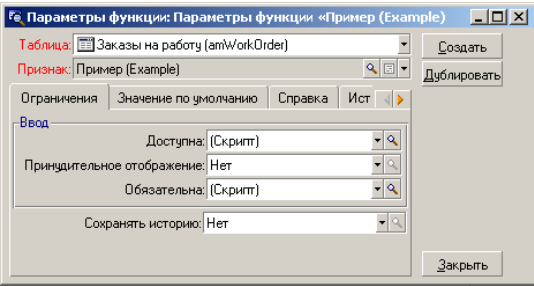
Первый пример

Этот раздел рассматривает гипотетическую проблему, которая может быть решена с помощью скрипта на языке Basic. Рекомендуем самостоятельно попытаться решить эту проблему, прежде чем обратиться к описываемому решению.

Описание проблемы

При закрытии заказов на работу необходимо заполнять признак с названием Example1, связанный с таблицей заказов на работу. Данный признак является дополнительным для заказов на работу, которые не закрыты. В остальной части примера предполагается, что признак уже создан, имеет произвольный тип ввода, связывается с таблицей заказов на работу, доступен и отображается

по умолчанию (принудительное отображение), как показано на записи экрана ниже:



Шаг 1: анализ и определение алгоритма

Алгоритм должен выполнять следующие задачи:

- устанавливать для поля **Обязательный** (имя SQL: seMandatory) значение Да, если закрывается квитанция.
- Установите для поля **Обязательный** (имя SQL: seMandatory) значение Нет во всех остальных случаях.

Поэтому наш алгоритм будет таким

Если закрывается заказ на работу, то заполнение признака является обязательным, иначе заполнение признака необязательно

Заказ на работу закрывается, если значение поля **Статус** (имя SQL: seStatus) также Закрытый.

Поэтому значением поля **Статус** (имя SQL: seStatus) в таблице заказов на работу определяется значение поля Обязательный (имя SQL: seMandatory) признака. Только это поле указывается в нашем алгоритме.

Раскрывающийся список в окне редактирования позволяет найти поле **Статус** в таблице заказов на работу.

После того, как поле выбрано, перенесите его в окно редактирования, нажав кнопку .

Данное поле заполняется с помощью системного подробного списка. Следовательно, имеется:

Значение в подробном списке	Связанное целое
Сообщено	0
Запланировано	1
Выполняется	2

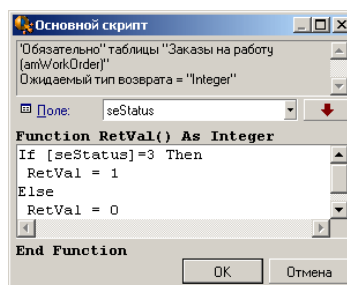
Значение в подробном списке	Связанное целое
Закрыто	3

Поэтому интересное значение в подробном списке:

- Закрытый, соответствующее значению «3».

Шаг 2: написание скрипта Basic

Теперь необходимо написать алгоритм из шага 1 на языке Basic.



Для подтверждения скрипта нажмите кнопку **ОК**.

Шаг 3: проверка скрипта Basic

Это шаг позволяет убедиться, что скрипт функционирует правильно.

- 1 Откройте таблицу заказов на работу, выбрав пункт меню **Портфель/Заказы на работу**. Выберите квитанцию со значением **Закрытый** (или создайте заказ на работу с таким статусом, если нет ни одного).
- 2 Выберите вкладку **Признаки**. В Asset Manager признак уже добавлен в соответствующий заказ на работу и данный признак является обязательным.
- 3 Теперь выберите заказ на работу со статусом, отличным от **Закрытый**. Перейдите на вкладку **Признаки** данного заказа на работу. Отображается признак Example1, но это необязательное поле.

Скрипт правильно выполняет свою функцию.

Второй пример

Этот раздел рассматривает гипотетическую проблему, которая может быть решена с помощью скрипта на языке Basic. Рекомендуем самостоятельно попытаться решить эту проблему, прежде чем обратиться к описываемому решению.

Описание проблемы

По умолчанию необходимо, чтобы в поле **Field1** (имя SQL: Field1) в сведениях о сотруднике показывались имя и фамилия сотрудника, если есть и то, и другое, или только фамилия, если имя пропущено.

Шаг 1: анализ и определение алгоритма

Алгоритм должен выполнять следующие задачи:

- отображать по умолчанию имя и фамилию сотрудника в поле **Field1** (имя SQL: Field1) в сведениях о сотруднике, если есть и имя, и фамилия;
- отображать по умолчанию только фамилию сотрудника в поле **Field1** (имя SQL: Field1) в сведениях о сотруднике, если нет имени.

Поэтому наш алгоритм будет таким:

Если имя сотрудника отсутствует, то значение по умолчанию поля Field1 - фамилия сотрудника, иначе значение по умолчанию поля Field1 - «Фамилия», «Имя»

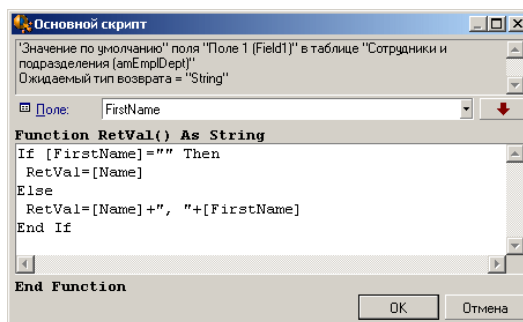
Поэтому значениями полей **Фамилия** (имя SQL: Name) и **Имя** (имя SQL: FirstName) в таблице сотрудников определяется значение по умолчанию поля **Field1** (имя SQL: Field1) в сведениях о сотруднике. Только эти два поля указываются в нашем алгоритме.

Щелкните правой кнопкой мыши поле **Field1** (имя SQL: Field1) в сведениях о сотруднике и выберите **Настройка объекта**.

Щелкните кнопку увеличения  около поля **По умолчанию**, чтобы приступить к редактированию скрипта на языке Basic.

Шаг 2: написание скрипта Basic

Теперь необходимо написать алгоритм из шага 1 на языке Basic.



Для подтверждения скрипта нажмите кнопку **OK**.

Шаг 3: проверка скрипта Basic

Это шаг позволяет убедиться, что скрипт функционирует правильно.

- 1 Откройте таблицу сотрудников и подразделений с помощью ссылки **Организация/Сотрудники и подразделения** на навигационной панели и создайте нового сотрудника.
- 2 Заполните поля **Фамилия** (имя SQL: Name) и **Имя** (имя SQL: FirstName) затем щелкните **Создать** для подтверждения. В Asset Manager отображаются имя и фамилия сотрудника в поле **Field1** (имя SQL: Field1). Скрипт правильно выполняет свою функцию.

12 Календари

Для доступа к списку календарей используется пункт меню «Инструменты/Календари».

Обзор календарей

Сведения о календаре

В сведениях о календаре показывается следующее.

- Общая информация, которая позволяет однозначно определять календарь:
 - Имя (имя SQL: Name) календаря;
 - Часовой пояс (имя SQL: TimeZone), к которому прикреплен календарь;
- описание обычных рабочих часов на вкладке Расписания;
- список исключений для данного расписания на вкладке Исключения;
- предварительный просмотр рабочих часов за указанный период, при котором учитываются правила, определенные на вкладках Расписание и Исключения.

Использование календарей

Объекты, с которыми связываются календари:

- операции рабочих процессов.

Они позволяют устанавливать время, когда должны инициироваться оповещения, определенные в схемах эскалации или в операциях рабочих процессов. Лимит времени, определяемый в данных оповещениях, фактически указывается в рабочих часах.



ВНИМАНИЕ:

Если изменяется календарь в базе данных, изменения учитываются на уровне полей, связанных с календарем, только после выхода из Asset Manager с последующим перезапуском.

Влияние календарей на некоторые области функциональности

Календари влияют на некоторые области функциональности Asset Manager. Изменение календаря вызывает как прямые, так и косвенные изменения некоторых записей в базе данных. Объекты, которыми используются календари:

- время выполнения задач рабочего процесса;
- оповещения, связанные с операциями рабочих процессов.

Методология, используемая для создания календаря

Для создания календаря используются следующие шаги:

- 1 Начните с определения календаря, указав его имя Имя (имя SQL: Name).
- 2 Если необходимо, свяжите календарь с часовым поясом, заполнив поле Часовой пояс (имя SQL: TimeZone).
- 3 Определите ежедневные рабочие часы на вкладке Расписания сведений о календаре.
- 4 Далее определите все исключения для этих рабочих часов на вкладке Исключения сведений о календаре.
- 5 Наконец, можно проверить функционирование календаря с помощью вкладки Предварительный просмотр.

Описание создания календаря

Календарь создается поэтапно:

- 1 Ввод общей информации

- 2 Заполнение вкладки «Расписание»
- 3 Заполнение вкладки «Исключения»
- 4 Проверка календаря

Ввод общей информации

Перед вводом фактических рабочих часов и исключений необходимо определить календарь, заполнив поле Имя (имя SQL: Name) на экране сведений.

Можно связать календарь с часовым поясом, заполнив поле Часовой пояс (имя SQL: TimeZone).

Заполнение вкладки «Расписание»

Вкладкой Расписания в сведениях о календаре определяются еженедельные расписания, связанные с данным календарем. Периодами рабочих часов, указанными на данной вкладке, определяются еженедельные расписания, связанные с данным календарем. Данными периодами описывается общее правило. Официальные праздники и им подобные дни представляют собой исключения и определяются на вкладке Исключения.

Рисунок 12.1. Календарь - вкладка «Расписания»

День недели	Периоды рабочего времени
Понедельник:	8:00-12:00;14:00-18:00
Вторник:	8:00-12:00;14:00-18:00
Среда:	8:00-12:00;14:00-18:00
Четверг:	8:00-12:00;14:00-18:00
Пятница:	8:00-12:00;14:00-18:00
Суббота:	
Воскресенье:	

Для каждого дня недели можно определить один или несколько периодов, представляющих собой рабочие часы. Для их определения можно использовать два метода.

- Графический - с помощью ползунков со шкалами, представляющих все дни недели.
 - 1 Щелкните элемент управления в начале периода расписания.

- 2 Расширьте выбор, перетащив курсор мыши к концу периода расписания. Asset Manager автоматически заполняет текстовое поле справа от градуированного элемента управления.
 - 3 Повторите при необходимости.
- «Вручную», с помощью текстового поля. Используйте следующий синтаксис для данного поля:

`<Первый час периода рабочего времени>-<Последний час периода рабочего времени>;< Первый час периода рабочего времени >-< Последний час периода рабочего времени >;...`

Для значений времени используется следующий формат:

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

Если дополнительный параметр [AM|PM] не указан, Asset Manager считает, что используется 24-часовой формат.

Asset Manager автоматически заполняет градуированный элемент управления - ползунок - слева от текстового поля.



ЗАМЕЧАНИЕ:

При использовании графического элемента управления значение устанавливается с точностью до получаса. При использовании текстового элемента управления - с точностью до минуты.

Заполнение вкладки «Исключения»

Вкладкой Исключения в сведениях о календаре определяются исключения в еженедельных периодах рабочего времени, определенных на вкладке Расписания.

Методология

Чтобы создать исключение:

- 1 Назначьте имя исключения.
- 2 Определите поле приложения исключения, заполнив поле Дни. Исключения могут определяться для указанных дня, месяца или года.
- 3 Также можно определить период рабочего времени в исключении с помощью поля Рабочие периоды. Данное поле позволяет определять более точные исключения, например: «В последнюю пятницу каждого месяца группа работает с 8:30 до 10:30 и с 17:30 до 18:30».

Вкладка «Исключения»



Данная вкладка делится на две части.

- Первая часть содержит список исключений и позволяет создавать, копировать, удалять, изменять и отменять изменения с помощью кнопок на панели инструментов:
 -  : Нажмите эту кнопку для создания нового исключения.
 -  : Нажмите эту кнопку для удаления исключения.
 -  : Нажмите эту кнопку для копирования исключения.



ЗАМЕЧАНИЕ:

Столбец Приоритет позволяет сортировать исключения по приоритету: им определяется, какое исключение имеет приоритет в случае конфликта. Asset Manager автоматически назначает исключению какой-либо приоритет (от P00 to P15). Чем меньше число, тем выше приоритет исключения. Таким образом, исключение с приоритетом P06 имеет больший приоритет, чем исключение с приоритетом P10.

- Вторая часть содержит сведения об исключении.
Значениями, принимаемыми полем Дни, определяется контекст применения исключения:

Значение поля Дни.	Контекст применения исключения
«Ежедневно»	Исключение применяется для всех дней года без исключения.
«День года:»	Исключение применяется для указанного дня или выбранных дней, определенных с помощью флажков День, Месяц и Год .
«Первый»	Правило применяется ко дню недели, определенному с помощью флажка День , и для месяцев и лет, определенных с помощью флажков Месяц и Год . Пример «Первая» пятница каждого месяца.
«Второй»	Правило применяется ко дню недели, определенному с помощью флажка День , и для месяцев и лет, определенных с помощью флажков Месяц и Год . Пример «Второй» понедельник сентября.

Значение поля Дни.	Контекст применения исключения
«Предпоследний»	Правило применяется ко дню недели, определенному с помощью флажка День , и для месяцев и лет, определенных с помощью флажков Месяц и Год . Пример «Предпоследняя» среда ноября.
«Последний»	Правило применяется ко дню недели, определенному с помощью флажка День , и для месяцев и лет, определенных с помощью флажков Месяц и Год . Пример «Последний» вторник каждого месяца в течение 2000 года.

Пример

Сотрудники компании Janus имеют следующие выходные дни:

- Первая пятница каждого месяца является выходным днем.
- В течение августа сотрудники компании Janus работают только по утрам с 8:30 до 12:30.

Правило №1: первая пятница каждого месяца является выходным днем.

- 1 Щелкните **Создать**, чтобы начать создание исключения.
- 2 Исключение применяется для первой пятницы каждого месяца для любого года. Поэтому флажки Месяц и Год не устанавливаются, что означает независимость исключения от месяца и года. Флажок День установлен, так как исключение применяется только по пятницам.
- 3 Установите для поля Дни значение «Первый».

Правило №2: в течение августа сотрудники компании Janus работают только по утрам с 8:30 до 12:30.

- 1 Нажмите кнопку **Создать**.
- 2 Данное исключение зависит только от месяца (применяется только для августа). Поэтому флажки День и Год не устанавливаются, флажок Месяц установлен и установлено соответствующее значение «Август».
- 3 Так как исключение применяется только для выбора дней (все дни на протяжении месяца августа), для поля Дни необходимо установить значение «День года:».

- 4 В этот период сотрудники работают с 8.30 утра до 12.30 дня. Для завершения ввода исключения нужно просто выбрать период с 8.30 утра до 12.30 дня в поле Рабочие периоды.

Проверка календаря

Вкладка Предварительный просмотр позволяет применять правила, определенные на вкладках Расписания и Исключения, для периода, выбранного с помощью полей Дата начала и Дата окончания в области Тест.

Рисунок 12.2. Календарь - вкладка «Предварительный просмотр»

Календарные планы Исключения Предварительный просмотр

Проверка

Дата начала: 12.07.2007 12:47:39 Дата окончания: 12.07.2007 12:47:39 Прошедшее время: 0с

Календарь, связанный с датой начала

Дата	Время
9 Июль 2007 г.	8:00-12:00;14:00-18:00
10 Июль 2007 г.	8:00-12:00;14:00-18:00
11 Июль 2007 г.	8:00-12:00;14:00-18:00
12 Июль 2007 г.	8:00-12:00;14:00-18:00
13 Июль 2007 г.	8:00-12:00;14:00-18:00
14 Июль 2007 г.	
15 Июль 2007 г.	

- Рамка Календарь, связанный с датой начала по умолчанию обеспечивает предварительный просмотр рабочих часов в течение недели, содержащей выбранную дату начала.
- В поле Прошедшее время указывается полное число рабочих часов в течение выбранного периода.



ЗАМЕЧАНИЕ:

Можно ввести значение продолжительности в поле **Прошедшее время** для принудительного перерасчета поля **Дата окончания** в соответствии с указанной **Датой начала**.

13 Часовые пояса

В данной главе объясняется, как использовать часовые пояса.



ЗАМЕЧАНИЕ:

Данная область функциональности доступна только для некоторых лицензионных соглашений Asset Manager.

Управление часовыми поясами

Так как клиентские компьютеры и сервер базы данных могут быть географически разнесены, Asset Manager управляет часовыми поясами и разностями во времени по отношению ко времени по Гринвичу. Asset Manager соблюдает следующие правила:

- все поля типа «Дата и время» отображаются на клиентском компьютере с соблюдением часового пояса данного клиентского компьютера;
- все поля типа «Дата и время» хранятся на сервере со ссылкой на определенный часовой пояс;
- при всех расчетах, включающих даты и времена, учитывается разница в часовых поясах.

Пример

Рассмотрим пример сервера, расположенного в Нью-Йорке, на котором есть данные, индексированные в соответствии с часовым поясом Парижа (Франция), и два клиентских компьютера, расположенные в Лондоне и в Париже. Начнем с определения часовых поясов каждого из этих клиентских компьютеров в соответствии со временем по Гринвичу (GMT).

- Часовой пояс сервера = GMT-5
- Часовой пояс клиента в Париже = GMT+1
- Часовой пояс клиента в Лондоне = GMT
- Часовой пояс данных = GMT+1

Таким образом, все значения типа «Дата и время» хранятся на сервере в формате GMT+1 и отображаются для клиента в Париже как GMT+1, а для клиента в Лондоне - как GMT. Например, при приеме заказа на работу на лондонском клиентском компьютере, если для срока разрешения заказа на работу установлено значение 15 мая 2000 года в 17:30, на других компьютерах это отображается следующим образом:

- на сервере: 15 мая 2000 г. в 12:30;
- у парижского клиента: 15 мая 2000 г. в 18:30;
- у лондонского клиента: 15 мая 2000 г. в 17:30.

Внедрение часовых поясов

Чтобы часовые пояса правильно обрабатывались в Asset Manager, необходимо выполнить следующие шаги:

- 1 Определите часовые пояса при создании базы данных в Asset Manager Application Designer с помощью параметра **Использовать часовые пояса**.
- 2 Создайте часовые пояса (например, импортировав информацию о часовых поясах).
- 3 Определите часовой пояс своего компьютера с помощью элемента меню Инструменты/Изменение часового пояса.
- 4 Определите календари в соответствии с часовыми поясами.

Создание часовых поясов

Функциональностью часовых поясов в Asset Manager, в отличие от Windows, учитываются изменения правил перехода на летнее время с годами. Это

позволяет отображать местное время в прошлом с большей точностью. Использование в Asset Manager информации о часовых поясах позволяет:

- отображать местные даты и время с учетом изменений в правилах перехода на дневное время;
- моделировать другое местоположение.

Чтобы избежать необходимости определения часовых поясов вручную, Asset Manager поставляется с файлом описания, содержащим основные часовые пояса. Данный файл можно импортировать с помощью следующей процедуры:

- 1 Выберите пункт меню **Файл/Импорт**. Asset Manager открывает окно выбора импорта.
- 2 Выберите «Выполнить скрипт», щелкнув кнопку . Asset Manager открывает экран обновления базы данных. Щелкните , чтобы выбрать скрипт для выполнения, в данном случае tz.scr в подпапке datasys папки установки Asset Manager.
- 3 Нажмите кнопку **Импорт**. Asset Manager выполняет импорт в соответствии со скриптом.

Управление часовыми поясами

В данном разделе более подробно рассматривается поле Летнее время (имя SQL: memDaylightInfo).

Формат поля «Летнее время»

Поле Дневное время (имя SQL: memDaylightInfo) имеет следующую структуру (одной строкой):

<code><Year>=<DaylightInfo> <Year> =<DaylightInfo> <Year>=<DaylightInfo> ...</code>

До конца данного раздела используются следующие соглашения:

- <Год>=<DaylightInfo> вместе называются «параметром»;
- <Год> и <DaylightInfo> по отдельности называются «аргументами».

В таблице ниже приведен обзор изменений летнего времени в соответствии со значениями аргументов <Год> и <DaylightInfo>.

	Аргумент <DaylightInfo> пуст	Аргумент <DaylightInfo> имеет значение
Аргумент <Year> - пуст. («<Year>=» не появляется)	Нет никаких изменений для летнего времени для часового пояса целиком.	Информация о летнем времени действительна для всех лет, кроме определяемых параметрами, имеющими аргумент <Год>.
Аргумент <Год> имеет значение	Неприменимо	Информация о летнем времени для данного часового пояса действительна для любого года из числа лет, указанных аргументом <Год>, вплоть до следующего аргумента <Год>.

Значения аргумента <Год>

Аргумент <Год>, которым указывается год, начиная с которого применима информация о летнем времени, определенная аргументом <DaylightInfo>, может принимать любое четырехзначное значение (например, 1990, 1997, 1998, 2012).

Значения аргумента <DaylightInfo>

Полный формат аргумента <DaylightInfo> является следующим (в одну строку):

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

Данный аргумент состоит из нескольких субаргументов, как показано ниже:

Субаргумент	Описание	Возможные значения
<StdShift>	Выражается в минутах, описывая разность во времени между стандартным временем в часовом поясе и временем рассматриваемого часового пояса. Например, для Парижа (часовой пояс GMT+1), если для субаргумента <StdShift> установлено значение 30 (минут), стандартное время в данном часовом поясе будет GMT+1 час 30 минут, а не GMT+1 час.	По умолчанию данный субаргумент имеет нулевое значение, но для него можно установить любое числовое значение. Пользователь должен проверить согласованность этого субаргумента.

Субаргумент	Описание	Возможные значения
<DltShift>	Выражается в минутах, описывая разность во времени между летним временем и временем рассматриваемого часового пояса.	По умолчанию для данного субаргумента устанавливается значение 60 (соответствующее разности по времени в 1 час между летним временем и эталонным временем «GMT +»), но может устанавливаться любое числовое значение. Пользователь должен проверить согласованность этого субаргумента.
<SDay>	День перехода с летнего времени на стандартное время.	"Monday" "Tuesday" "Wednesday" "Thursday" "Friday" "Saturday" "Sunday" Пусто (в таком случае, необходимо установить для субаргумента <SDayPos> значение между 1 и 31, чтобы указать день перехода со стандартного времени на летнее время)
<SMonth>	Месяц перехода с летнего времени на стандартное время.	"January" "February" "March" "April" и т.д. "November" "December"
<SDayPos>	Положение в месяце дня перехода с летнего времени на стандартное время.	"First" "Second" "Third" "Fourth" "Last" «Предпоследний» (перед последним) Значение между 1 и 31, если субаргумент <SDay> пусто
<SHour>	Время перехода с летнего времени на стандартное время (выраженное по летнему времени).	Любое значение, выраженное в 24-часовом формате (ЧЧ:ММ:СС).

Субаргумент	Описание	Возможные значения
<DDay>	День перехода со стандартного времени на летнее время.	"Monday" "Tuesday" "Wednesday" "Thursday" "Friday" "Saturday" "Sunday" Пусто (в таком случае, необходимо установить для субаргумента <DDayPos> значение между 1 и 31, чтобы указать день перехода с летнего времени на стандартное время).
<DMonth>	Месяц перехода со стандартного времени на летнее время.	"January" "February" "March" "April" и т.д. "November" "December"
<DDayPos>	Положение в месяце дня перехода со стандартного времени на летнее время.	"First" "Second" "Third" "Fourth" "Last" «Предпоследний» (перед последним) Значение между 1 и 31, если субаргумент <DDay> пуст
<DHour>	Время перехода со стандартного времени на летнее время (выраженное по стандартному времени).	Любое значение, выраженное в 24-часовом формате (ЧЧ:ММ:СС).

Пример

В качестве примера рассмотрим информацию о летнем времени для часового пояса «(GMT+01:00) Париж, Мадрид, Амстердам».

```
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
```

Идентификация последних параметров.

Первый параметр



- <Год> = 2000 означает, что следующие параметры <DaylightInfo> применимы, начиная с 2000 года.
- <StdShift> = 0 означает, что в этом часовом поясе нет различия между часовым поясом и стандартным временем.
- <DltShift> = 60 означает, что различие между стандартным и летним временем составляет 60 минут, т.е. 1 час. Таким образом, летнее время равно времени часового пояса плюс один час.
- <SDay> = Воскресенье означает, что переход на стандартное время осуществляется в воскресенье.
- <SMonth> = Октябрь означает, что переход на стандартное время осуществляется в октябре месяца.
- <SDayPos> = Последний указывает позицию дня в месяце. В этом случае переход на стандартное время осуществляется в последнее воскресенье октября месяца.
- <SHour> = 03:00:00 означает, что переход на стандартное время осуществляется в 03:00.
- <DDay> = Воскресенье означает, что переход на летнее время осуществляется в воскресенье.
- <DMonth> = Март означает, что переход на летнее время осуществляется в марте месяца.
- <DDayPos> = Последний указывает позицию дня в месяце. В этом случае переход на летнее время осуществляется в последнее воскресенье марта месяца.
- <DHour> = 02:00:00 означает, что переход на летнее время осуществляется в 02:00.

Второй параметр



- Так как нет аргумента <Год>, данным параметром указывается, что он действителен только для тех лет, которые не описаны в предыдущем параметре.
- <StdShift> = 0 означает, что в этом часовом поясе нет различия между часовым поясом и стандартным временем. Зимнее время, таким образом, совпадает со временем часового пояса.
- <DltShift> = 60 означает, что различие между стандартным и летним временем составляет 60 минут, т.е. 1 час. Таким образом, летнее время равно времени часового пояса плюс один час.
- <SDay> = Воскресенье означает, что переход на стандартное время осуществляется в воскресенье.
- <SMonth> = Сентябрь означает, что переход на стандартное время осуществляется в сентябре месяце.
- <SDayPos> = Последний указывает позицию дня в месяце. В этом случае переход на стандартное время осуществляется в последнее воскресенье сентября месяца.
- <SHour> = 03:00:00 означает, что переход на стандартное время осуществляется в 03:00.
- <DDay> = Воскресенье означает, что переход на летнее время осуществляется в воскресенье.
- <DMonth> = Март означает, что переход на летнее время осуществляется в марте месяце.
- <DDayPos> = Последний указывает позицию дня в месяце. В этом случае переход на летнее время осуществляется в последнее воскресенье марта месяца.
- <DHour> = 02:00:00 означает, что переход на летнее время осуществляется в 02:00.

В результате:

 ЗАМЕЧАНИЕ:

Начиная с 2000 года переход на стандартное время осуществляется в последнее воскресенье октября в 03:00:00 (часы переводятся назад на 02:00:00), а переход на летнее время осуществляется в последнее воскресенье марта в 02:00:00 (часы переводятся вперед на 03:00:00).

Для всех лет до 2000 года переход на стандартное время осуществляется в последнее воскресенье сентября в 03:00:00, а переход на летнее время осуществляется в последнее воскресенье марта в 02:00:00.

Управление часовыми поясами в Asset Manager Automated Process Manager

Asset Manager Automated Process Manager позволяет настраивать тесты для часовых поясов. Выберите пункт меню **Инструменты/Настройка модулей**.

Тесты, которые необходимо выполнить

На вкладке **Общее** экрана конфигурации можно конфигурировать типы тестов, которые необходимо выполнить.

- Проверьте часовой пояс сервера базы данных.
- Сравните локальное время и время сервера.

Эти два теста сравнивают время сервера базы данных со временем компьютера, на котором установлен Asset Manager Automated Process Manager. Разница во времени выражается как $[(n * 30 \text{ минут}) + m]$, где m находится в диапазоне от -15 до +15 минут.

В обоих случаях

Если разница во времени превышает 5 минут, Asset Manager Automated Process Manager предлагает обновить местное время на том компьютере, где он установлен.

В случае отказа от такого обновления (например, если оказывается, что необходимо изменить время на сервере) следует отказ в подключении. Повторное подключение будет возможно после того, как разница во времени станет меньше 5 минут (в результате изменения времени сервера базы данных и (или) компьютера, на котором установлен Asset Manager Automated Process Manager).

Конкретные аспекты параметра «Проверить часовой пояс сервера базы данных»

При необходимости информация о часовых поясах сервера в таблице параметров Asset Manager обновляется (если число ($n * 30$ минут) не соответствует часовому поясу сервера).



ЗАМЕЧАНИЕ:

Для этого необходимо, чтобы на компьютере, на котором выполняется Asset Manager Automated Process Manager, было установлено правильное время и правильные сведения о переходе на летнее время.

Конкретные аспекты параметра «Проверить местное время по сравнению со временем сервера»

Часовой пояс сервера, необходимый для внутренних операций Asset Manager, восстановлен.

Частота тестирования

Тест выполняется:

- 1 Сначала, когда Asset Manager Automated Process Manager подключается к базе данных.
- 2 Затем регулярно, в соответствии с расписанием, указанным на Сервере Asset Manager (**Инструменты/Настройка модулей**).

Последствия для различных операций

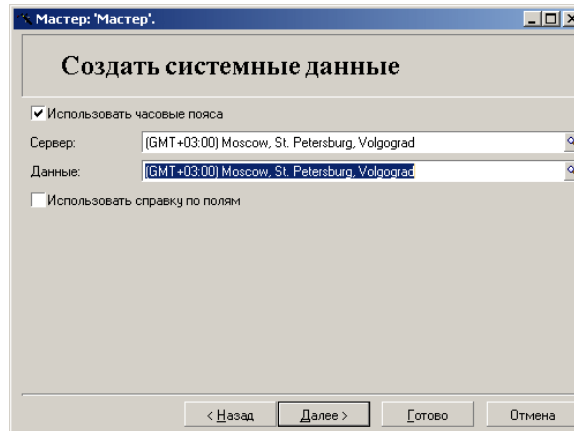
Способ, которым управляются часовые пояса, оказывает влияние на целый ряд операций:

- Создание базы данных
- Подключение к базе данных
- Импорт и экспорт

Создание базы данных

При создании базы данных Asset Manager позволяет определять параметры часовых поясов. Выберите пункт меню Действие/Создать базу данных в

Asset Manager Application Designer. Рамка Создание системных данных содержит параметры, относящиеся к часовым поясам.



Флажком **Использовать часовые пояса** определяется, учитываются ли при создании базы данных часовые пояса.

- Если данный флажок установлен, при создании базы данных учитываются часовые пояса.
- Если данный флажок снят, при создании базы данных часовые пояса не учитываются.

Полями **Сервер** и **Данные** определяются фактический часовой пояс сервера при создании базы данных и часовой пояс, используемый для хранящихся данных.

ЗАМЕЧАНИЕ:

Данный параметр доступен только при создании базы данных. Он позволяет определять эталонное время сервера и данных. Если изменить данные значения, значения типа «Дата и время», уже содержащиеся в базе данных, теряют смысл.

Подключение к базе данных

При подключении к базе данных Asset Manager ищет в файле `am.ini` запись `LocalTimeZone`, которой определяется часовой пояс клиентского компьютера.

Местоположение файла: ► Руководство по установке и обновлению Asset Manager, глава **Файлы .ini и .cfg**.

Если нельзя найти такую информацию, Asset Manager использует системный часовой пояс (определенный в Windows).

Далее Asset Manager ищет базу данных для часового пояса, соответствующую записи LocalTimeZone в файле am.ini, или часовой пояс, определенный в Windows.

В таблице ниже приводится сводка различных возможных ситуаций:

Существование записи LocalTimeZone в файле am.ini	Наличие соответствующего часового пояса в таблице часовых поясов	Информация, сохраненная в файле am.ini для записи LocalTimeZone
Да	Да	Часовой пояс базы данных
	Нет	Неизменен
Нет	Да	Часовой пояс базы данных
	Нет	Системный часовой пояс

Установка времени на клиентском компьютере

При подключении к удаленной базе данных Asset Manager проверяет правильность времени, указываемого часами на уровне клиента, по отношению к часам сервера.

Расхождение в показаниях часов - разность синхронизации, которое не следует путать с разностью по времени, которая представляет собой транспозицию по отношению к разности часовых поясов. Asset Manager рассчитывает часовой пояс клиентских часов и определяет расхождение в показаниях часов между двумя компьютерами. Такой расчет выполняется следующим образом:

$$\text{Расхождение} = \text{Модуль}((\text{Разность в минутах между временами двух рассматриваемых компьютеров})/30)$$

 ЗАМЕЧАНИЕ:

Модуль представляет собой остаток от деления.

Например, для следующих компьютеров:

- компьютер А находится в часовом поясе GMT и его часы показывают 18:02;
- компьютер В находится в часовом поясе GMT+1 и его часы показывают 18:19 (то есть 17:19 для компьютера А, 43 минуты разности с компьютером А))

$$\text{Расхождение} = \text{Модуль} (43/30) = 13 \text{ минут}$$

Если такая разница превышает пять минут (фиксированное значение), Asset Manager предлагает откорректировать показания часов на уровне клиента.

В случае отказа пользователя происходит сбой подключения.

Asset Manager выполняет такую проверку периодически, а также при изменении времени на клиентском компьютере. По умолчанию такая проверка выполняется каждые 60 минут, но может настраиваться изменением параметра `g_ITimeZoneCheckInMns` в файле `am.ini`, в разделе `[option]`.

Местоположение файла: ► Руководство по установке и обновлению Asset Manager, глава Файлы `.ini` и `.cfg`.

```
[option]
g_ITimeZoneCheckInMns = 30
```

Для периодичности проверки расхождения в показаниях часов устанавливается значение 30 минут.

Такая периодичность может также настраиваться с помощью параметра Проверка часового пояса сервера базы данных в Asset Manager.



ЗАМЕЧАНИЕ:

Такая проверка функционирует только в том случае, если соответствующая база данных была создана с учетом часовых поясов.

Импорт и экспорт

Для данных двух функций преобразование выполняется, предполагая, что все поля «Дата и время» соответствуют часовому поясу компьютера, выполняющего импорт или экспорт.

14 Вычисляемые поля

Доступ к экрану для создания вычисляемого поля возможен с помощью меню **Администрирование/Вычисляемые поля**.

Определение вычисляемого поля

Вычисляемое поле - поле, значение которого вычисляется в соответствии со значениями других полей и переменных по определяемой пользователем формуле. Имеются три типа вычисляемых полей:

- AQL
- Basic
- вычисляемое поле

Каждым из этих типов используется различный язык для формул расчета и оказывается влияние на возможности и ограничения, связанные с использованием поля. Например, в фильтрах могут использоваться только вычисляемые поля типа AQL.



ЗАМЕЧАНИЕ:

Вычисляемые поля - виртуальные поля только для чтения (в базе данных хранится только одна формула). Можно определять любое необходимое количество вычисляемых полей и назначать им права пользователя.

Полезность вычисляемых полей

Вычисляемые поля позволяют определять дополнительную информацию и рассчитывать синтетическую информацию на основе записей в базе данных Asset Manager. Таким образом, они подобны «классическим» полям, за исключением нескольких отличий:

- в отличие от «классических» полей, значения вычисляемых полей не хранятся в базе данных Asset Manager;
- значение вычисляемого поля не заполняется пользователем, а дается формулой;
- Нельзя связывать вычисляемое поле с одной записью указанного поля. Подобно всем остальным «классическим» полям в базе данных, вычисляемое поле связывается со всеми записями в таблице и имеет значение (может быть нулевым) для каждой записи в данной таблице.
- Вычисляемые поля не показываются на экране сведений о записи. Они могут отображаться только в списке.
- Вычисляемые поля могут использоваться только при расчете значений по умолчанию стандартных полей, если их тип - Вычисляемая строка или Скрипт на языке Basic.

Создание вычисляемых полей

Перед созданием таких полей следует ознакомиться с конкретными деталями, свойственными каждому типу вычисляемого поля.

Данный раздел включает информацию по следующим темам:

- Введение
- Методы создания

Введение

Каждый тип вычисляемых полей имеет разные свойства, которыми определяется, как он используется.

В следующей таблице собраны основные отличия между тремя типами:

Таблица 14.1. Типы вычисляемых полей

Тип поля	Свойства полей данного типа			Поле рас- считывает	Характеристики языка, используемого форму- лой расчета	
	Может отображать- ся	Может сор- тироваться	Может ис- пользовать- ся в филь- трах		Преимуще- ства	Недостатки
AQL	Да	Да	Да	Сервер ба- зы данных	Эффекти- вен Встроен- ный редак- тор	Ограничен- ный по воз- можностям язык. Поля данно- го типа не могут ис- пользовать- ся в значе- ниях по умолчанию
Вычисляе- мые строки	Да	Да	Нет	Клиент	Простой	Не очень эффекти- вен (только простое объедине- ние строк и значений полей и строк)
Basic	Да	Нет	Нет	Клиент	Много воз- можностей Гибок	Поля данно- го типа мо- гут только отображать- ся



ЗАМЕЧАНИЕ:

Данная таблица ясно показывает, что поля типа AQL имеют гораздо более широкий диапазон применения, чем два других типа вычисляемых полей.

Для запроса AQL можно использовать все три свойства (может отображаться, сортироваться, использоваться в фильтрах):

Свойство	Соответствующие аргументы AQL
Может отображаться	SELECT Оператор
Может сортироваться	SELECT ORDER BY GROUP BY Операторы
Может использоваться в фильтрах	SELECT ORDER BY GROUP BY WHERE HAVING Операторы

Дополнительную информацию запросах AQL можно найти в главе Запросы AQL данного руководства.

Расчет сервером/клиентским компьютером

В случае поля типа AQL сервер базы данных выполняет необходимые расчеты и отправляет результат на клиентский компьютер. По данной причине отсутствует влияние на быстродействие клиентского компьютера и уменьшается сетевой трафик. С другой стороны, запросы SQL, отправляемые базе данных, являются более сложными.

Методы создания

В данной главе подробно описывается метод, используемый для создания вычисляемого поля.

Анализ потребностей

Тип поля следует выбирать на основе двух следующих критериев.

- Свойства типа поля: возможность отображения, сортировки, использования в фильтрах или в значениях по умолчанию.
- «Стоимость» такого решения - сложность используемой формулы по сравнению с возможностью использования. Три типа вычисляемых полей могут классифицироваться по степени нарастания «стоимости» следующим образом:
 - вычисляемое поле
 - AQL
 - Basic



ЗАМЕЧАНИЕ:

Всюду, где возможно, рекомендуется использовать наименее «затратное» решение.

Например:

- если поле служит только для информационных целей, будет достаточно вычисляемого поля типа Basic;
- если необходимо сортировать записи в соответствии со значением поля, необходимо использовать вычисляемые поля типа AQL или «вычисляемая строка»;
- если необходимо фильтровать записи в соответствии со значением поля, необходимо использовать вычисляемые поля типа AQL.

После определения своих потребностей можно переходить к следующей фазе.

Открытие экрана создания

Выберите пункт меню Администрирование/Вычисляемые поля. Asset Manager отображает экран создания вычисляемого поля:

Название: Basic

Имя SQL: Basic cf_Basic

Описание: Basic

Таблица: Ресурсы (amAsset)

Тип поля: Основной скрипт

Тип результатов: Текст

☐ Только для программирования.

Синтаксис BASIC

```
If Left ([AssetTag], 2) = "PC" Then
  RetVal = "PC"
Else
  RetVal = "Other"
End If
```

Идентификация вычисляемого поля

Сначала заполните верхнюю часть данного экрана, чтобы уникальным образом идентифицировать вычисляемое поле:

- Поле Метка (имя SQL: Label) содержит метку вычисляемого поля, используемую для заголовков столбцов в списках;
- Поле Имя SQL (имя SQL: SQLName) содержит имя SQL вычисляемого поля. Данное поле с префиксом cf_ используется, например, при ссылке на данное поле в скриптах на языке Basic, запросах или фильтрах;



ЗАМЕЧАНИЕ:

Имя SQL поля после его создания не должно изменяться. Перестанут быть действительными любые ссылки на данное поле, в которых используется предыдущее имя SQL.

- Поле Описание (имя SQL: Description) содержит краткое описание поля, используемого в списках, показывающих поля (например, в фильтрах или на экране конфигурации списка).

Определение контекста использования поля

Таблица (имя SQL: TableName) и Тип поля (имя SQL: seType) позволяют определять контекст использования вычисляемого поля:

- Поле Таблица (имя SQL: TableName) позволяет связывать вычисляемое поле с таблицей. Данное поле будет доступно только для этой таблицы;
- Поле Тип результата (имя SQL: seType) позволяет указывать тип вычисляемого поля. В соответствии с данным типом будут отличаться свойства поля (возможность отображения, сортировки или использования в фильтрах);
- поле Тип результата позволяет указывать тип результирующего значения вычисляемого поля. Данный тип используется в целях форматирования и отображения. Таким образом, вычисляемое поле с типом результата «Дата» отображается таким же образом, что и все остальные поля типа «Дата» в базе данных.

Ввод формулы вычисления для поля

Теперь необходимо записать формулу вычисления поля. Можно или ввести ее непосредственно в текстовое поле в нижней части экрана (заметьте, что метка данного поля изменяется в соответствии с присваиваемым типом), или щелкните значок увеличения , или нажмите клавишу F4, чтобы получить доступ к соответствующему экрану редактора.



ЗАМЕЧАНИЕ:

Используемый язык отличается в соответствии с используемым типом поля.

Дополнительную информацию о языках, доступных для написания формул вычисления, можно найти в следующей документации:

- глава Скрипты данного руководства. Используемая функция - RetVal();
- глава Запросы AQL данного руководства для языка AQL;

- руководство Администрирование, глава Стандартные файлы описания базы данных, раздел Описание таблиц, подраздел Строки описаний таблицы для вычисляемых строк.

Определение прав пользователя для вычисляемого поля

Выберите пункт меню Администрирование/Права пользователя. Asset Manager отображает экран создания прав пользователя.



ЗАМЕЧАНИЕ:

Вычисляемые поля доступны только для доступа к статусу.

- 1 Введите краткое описание права пользователя в поле Описание (имя SQL: Description) и, дополнительно, комментарий в поле Комментарий (имя SQL: Comment).
- 2 Разверните древовидную структуру таблицы, связанной с вычисляемым полем. Ветвь, указываемая значком  (Вычисляемое поле), включает полный список вычисляемых полей для рассматриваемой таблицы.
- 3 Затем выберите поле, для которого необходимо редактировать право пользователя. Флажок Чтение в области Поля, ссылки и признаки позволяет определить права на считывания для данного поля. Если установлен данный флажок, просмотр данного вычисляемого поля возможен только для профилей с данным правом пользователя. Если флажок не установлен, то доступ к данному полю (только для чтения) будут иметь все пользователи.

Использование вычисляемых полей

Использование вычисляемого поля зависит от его типа. Необходимо удостовериться, что тип поля совместим с его предполагаемым использованием. В списках, показывающих поля (создания фильтра, конфигурирования списка и так далее), Asset Manager упрощает задачу, показывая только те поля, которые можно использовать.

Использование вычисляемых полей в конфигурациях списков

Можно отобразить значение поля, рассчитанное для всех записей в таблице, с помощью команды Настройка списка в контекстном меню.



ВНИМАНИЕ:

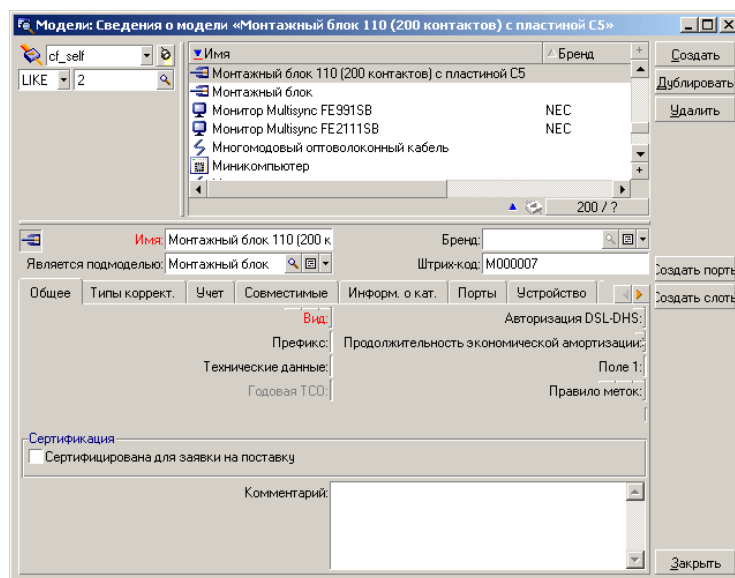
Это не срабатывает, если в качестве СУБД используется Microsoft SQL Server (вычисляемое поле не отображается).

Фильтрация записей таблицы

Asset Manager позволяет фильтровать записи в таблице в соответствии со значением вычисляемого поля типа AQL. Для этого выберите в контекстном меню команду Простой фильтр и перейдите к ветви Вычисляемые поля. Asset Manager показывает поля только типа AQL.

Ссылки на вычисляемые поля

Для ссылок на вычисляемые поля используются имена SQL с префиксом cf_. На снимке экрана, приведенном ниже, иллюстрируется использование имени SQL вычисляемого поля в фильтре:



Кроме того, вычисляемые поля могут использоваться различными модулями или функциями Asset Manager:

- Asset Manager Web
- Asset Manager API

- Отчеты

Дополнительную информацию о данных модулях или областях функциональности можно найти в соответствующей документации.

15 Мастера

Asset Manager позволяет создавать свои собственные мастера, а также приспосабливать существующие мастера для собственного использования. Мастера хранятся в виде текстовых полей (поле Скрипт мастера (имя SQL: WizardScript), вкладка Мастер сведений о действии типа Помощник). Создание мастера заключается в вводе его кода в данное поле непосредственно или с помощью графического редактора. Для этого необходимо знакомство со структурой мастеров и языком написания скриптов, используемым для описания их структуры.



ВНИМАНИЕ:

Если вы модифицируете действия в отношении скрипта или мастера, обязательно повторно отметьте веб-службы, если вы выполняли развертывание Asset Manager Web.

► **Руководство Tailoring, глава Customizing the database, раздел Development best practices/ Tag the Web service.**

Соглашения о нотации

Ниже приводится нотация, используемая для описания структуры мастеров:

Таблица 15.1. Используемые соглашения

[]	Квадратные скобки используются для ссылки на значение поля в базе данных (в случае контекстных мастеров) или одного из «специальных полей»: CurrentSelection и CurrentTable. Они также используются для указания дополнительных параметров.
< >	Угловыми скобками обозначаются значения свойств, описываемые простым языком. Данные угловые скобки не должны вводиться так, как они есть. Они сами и текст внутри них должны замещаться соответствующей информацией.
	Знак вертикальной черты используется для разделения возможных значений свойств. Он также используется для разделения заголовков и значений списков с несколькими столбцами.
{ }	В фигурные скобки заключаются определение узла или блок скрипта, занимающий несколько строк свойства. Они также используются для ссылки на значение свойства мастера.
'	В примерах кода на языке Basic апостроф обозначает строку комментариев, которая не интерпретируется в Asset Manager.
; или //	В мастерах точка с запятой или две косые черты обозначают строку комментариев, которая не интерпретируется в Asset Manager.

Определения

Ниже даны определения терминов, используемых при описании структуры мастеров.

Твип

Твип - единица размера и расстояния по умолчанию, используемая мастерами. Она не зависит от разрешения экрана. Она соответствует:

- 1440 твипов равняются одному дюйму;
- 567 твипов равняются одному сантиметру;
- при разрешении 96 точек на дюйм (стандартное разрешение Windows) 15 твипов эквивалентны одному пикселю.

Элемент управления

Элемент управления представляет собой графический элемент, который позволяет изменять элементы данных. Обычные элементы управления - это флажки, области изменения текста, кнопки, раскрывающиеся списки и так далее.

Узел

Узел соответствует уровню иерархии в древовидной структуре мастера. Подузел данного узла N представляет собой узел, который ниже на один уровень в древовидной структуре, прикрепленной к узлу N.



ЗАМЕЧАНИЕ:

В именах узлов не разрешается использовать только символы с акцентами. Длина имен узлов ограничена 22 символами.

Объект

Объект - общий термин, которым обозначается:

- весь мастер;
- страница мастера;
- элемент управления (флажок, область изменения текста, кнопка, раскрывающийся список и так далее) на странице;
- переменная;
- и т.п.

Родительский объект и дочерний объект

Если объект A содержит объект B:

- объект A называется родительским объектом объекта B;
- объект B называется дочерним объектом объекта A.

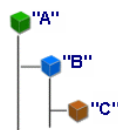


ВНИМАНИЕ:

При таком отношении имеется в виду состав, а не наследование.

Полное имя объекта

Полное имя объекта состоит из имен всех его родительских объектов и имени самого объекта. Каждый объект разделяется точкой (.). Возьмем в качестве примера следующую структуру:



Следовательно, полное имя объекта C:

<Имя объекта A>.<Имя объекта B>.<Имя объекта C>

Переменная

Переменная - постоянное пространство хранилища, содержащее данные, которые могут изменяться при выполнении мастера. Каждая переменная имеет имя, позволяющее уникально идентифицировать ее в мастере. Все переменные, используемые в мастере, являются глобальными. Это означает, что на них можно ссылаться в любом данном узле мастера с помощью их полных имен.

Мастерами в Asset Manager используются два типа переменных.

- «Переменные мастера», которыми определяются узлы типов LONG или STRING. Тип переменной определяется типом узла; переменная, определяемая в узле LONG, является длинным целым значением, переменная, определяемая в узле STRING, - строкой символов. Данные переменные являются глобальными по определению. То есть, например, на них можно ссылаться с помощью их полных имен в любом узле мастера. При необходимости, данные переменные автоматически пересчитываются в Asset Manager.
- Переменные Basic, используемые в скриптах на языке Basic в мастере. По умолчанию данные переменные являются локальными, но их можно сделать глобальными с помощью свойств COMMON и GLOBAL. Данные переменные не пересчитываются автоматически в Asset Manager.

Переход

Переход - процесс смены одной страницы мастера на другую. Для данной страницы могут определяться несколько переходов. Каждый из них имеет свои собственные, определяемые пользователем условия, от которых зависит их действительность и которые должны выполняться, чтобы инициировать переход.

- Когда пользователь щелкает кнопку **Далее**, выполняется первый действительный переход (то есть переход, для которого удовлетворяются условия). Если нет никаких действительных переходов, кнопка **Далее** отключена.
- Если для мастера есть обязательные свойства, которые не заполнены, кнопку **Далее** использовать нельзя.
- Если пользователь щелкает кнопку **Готово** до того, как пройти все шаги мастера полностью, для незавершенных шагов используются значения по умолчанию.

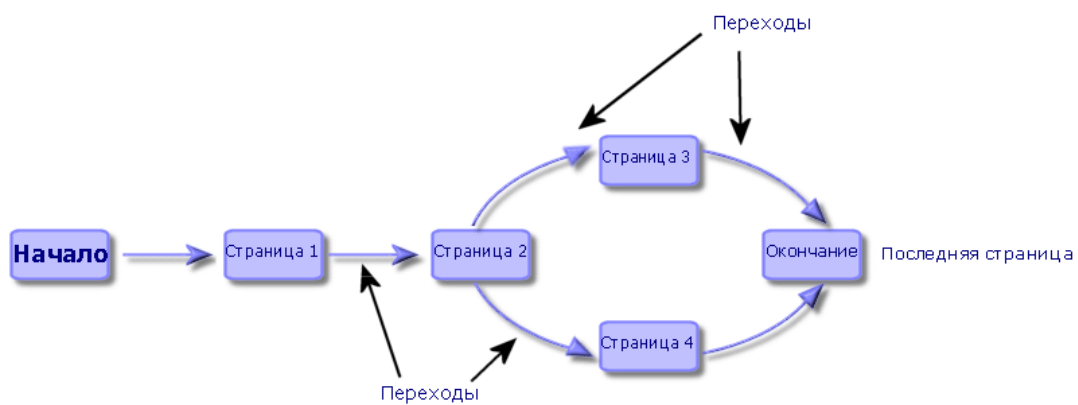
Структурный шаблон

Мастер Asset Manager представляет собой результат выполнения какого-либо мастера. По определению, структура мастера основывается на структурном скрипте. Т. е.:

- Скрипт мастера (и, следовательно, мастер) состоит из узлов.
- Каждый узел мастера имеет имя, один или несколько подузлов и набор свойств. Ниже перечислены типы узлов:
 - ROOT (Корневой узел). Данный узел является уникальным и заключает в себе все остальные.
 - START. Данный узел является уникальным и содержит скрипт, который выполняется при запуске мастера.
 - PAGE. Данным типом узла описывается страница мастера.
 - TRANSITION. Данным типом узла описывается переход между двумя узлами типа PAGE.
 - FINISH. Данный узел является уникальным и содержит скрипт, который выполняется при завершении мастера.
 - PARAMS. Данный узел является уникальным и содержит параметры, передаваемые в другой мастер. Несколько мастеров могут выполняться последовательно (с обменом параметрами или без него). Такие мастера называются соединенными.
 - LONG или STRING. Данным типом узла определяется соответствующий тип переменной.
- Значение свойства можно определить или через константу, или с помощью скрипта на языке Basic (в таком случае, значение является результатом оценки скрипта).

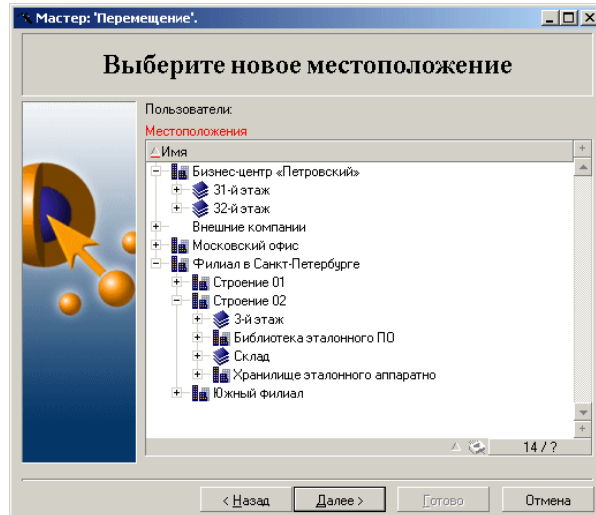
Мастера состоят из страниц, связанных переходами. Переход с одной страницы на другую определяется информацией, введенной сделавшим выбор пользователем. На диаграмме ниже проиллюстрирована структура мастера:

Рисунок 15.1. Мастера - структурные шаблоны



Шаблон страницы мастера

Страница мастера организована следующим образом:



Общее число пунктов

Программный код мастера (поле Скрипт мастера (имя SQL: WizardScript) на вкладке Мастер сведений о действии, использующем мастер) имеет вид структурированного текста, состоящего из блоков, разделяемых скобками ({ }). Данным текстом определяется структура мастера.

Каждый узел («Корень», «Страница» и так далее) в древовидной структуре мастера имеет неограниченное число подузлов и набор свойств.

Общая структура и синтаксис

Узлы имеют следующие структуру и синтаксис:

```
; This is a comment outside of the script
{ <Node type> <Node name>
  <Name of the property> = <Value of the property>
  ' This is a comment inside the script
```

```

...
{ <Name of the property> =
...
}
{ <Node type> <Node name>
<Name of the property> = <Value of the property>
...
{ <Name of the property> =
...
}
}

```

Для узла применяются следующие правила:

- Имена узлов являются необязательными. Если для узла не указано никакого имени, Asset Manager присваивает ему имя и номер автоматически.
- Имена узлов не могут включать пробелы.
- Если имя узла «=», это уже не узел, а многострочное свойство. Дополнительную информацию о многострочных свойствах можно найти в разделе «Синтаксис свойств» данной главы.
- Строки, начинающиеся с точки с запятой (;) за пределами скрипта, и строки, начинающиеся с апострофа (') внутри скрипта, интерпретируются как комментарии и игнорируются.

ЗАМЕЧАНИЕ:

Внимание: должен соблюдаться пробел между скобкой («{») и типом узла. Если это не так, Asset Manager откажется выполнять мастер.

Свойства узла

Значения свойств могут определяться с помощью констант или скриптов. Константы могут быть числовые, булевы или текстовые.

ЗАМЕЧАНИЕ:

Свойства, связанные с объектами, могут быть дополнительными или обязательными. Они могут быть логическими (дополняющими определение объекта), или физическими (имеющими влияние на визуальное восприятие объекта).

Декларативный шаблон

Свойство определяется в соответствии с декларативным режимом, которым определяются циклические ссылки ($A=\{B\}$, $B=\{A\}$):

`<Имя свойства>=<Скрипт>`

С данным определением связывается список зависимостей. Если имеется:

`A={B}+{C}`

Свойство A зависит от свойств B и C. Следовательно, список зависимостей A: B, C.

В результате свойство изменяется:

- если изменяется одно из свойств в его списке зависимостей;
- после действия пользователя, изменившего свойство или зависимое свойство.

Определение константы как значения свойства

Постоянное значение свойства определяется следующим синтаксисом:

- Свойство типа «Текст»:
 - `<Имя свойства> = "<Текст>"`
- Свойство типа Boolean:
 - `<Имя свойства> = TRUE`
 - `<Имя свойства> = FALSE`
 - `<Имя свойства>` (эквивалентно `<Имя свойства> = TRUE`)
- Свойство числового типа:
 - `<Имя свойства> = 42`
- `<Имя свойства> = {<Полное имя переменной Basic или свойство>}`



ЗАМЕЧАНИЕ:

Булево значение **TRUE** эквивалентно числовому значению, отличному от 0. Значение **FALSE** эквивалентно числовому значению 0.

Ссылка на свойство

Для ссылки на свойство объекта (то есть на содержимое данного свойства или объекта, и, в частности, на его значение) используется следующий синтаксис:

`{<Полное имя свойства>}`

Таким образом, если необходимо сослаться на свойство Prop страницы Page1, необходимо записать

```
{Page1.Prop}
```

В таком синтаксисе для полного имени должен учитываться регистр.

Определение скрипта как значения свойства

Понятие скрипта

Скрипт представляет собой однострочную или многострочную программу на языке Basic, которой возвращается значение в глобальную переменную RetVal. В случае однострочного скрипта данная переменная является неявной. В случае многострочного скрипта ее необходимо указывать.

Как и в случае всех скриптов на языке Basic, необходимо обращать внимание на тип возвращаемого значения. Он зависит от типа свойства, рассчитываемого с помощью скрипта.

Синтаксис однострочного скрипта

```
<Имя свойства>=<Скрипт>
```

Например:

```
Variable="The name is: " & {Name}
```

Предыдущий однострочный скрипт эквивалентен следующему многострочному скрипту:

```
{ Variable = RetVal="The name is: " & {Name} }
```

Синтаксис многострочного скрипта

```
{ <Имя свойства >=  
<Скрипт>  
}
```

Например:

```
{ LABEL =  
IF {Page1.Title}="Choose an employee" THEN  
RetVal="Employee"  
ELSE  
RetVal="Department"  
END IF  
}
```

Методы, применимые в свойствах

Метод позволяет восстанавливать значение, связанное со свойством или узлом, или даже выполнять функцию для данного свойства. В данном смысле его можно рассматривать как дополнительную функцию.

Метод имеет следующий синтаксис:

```
{node.node.node[.property][.method([arg1[, arg2[...]]])]}
```

где:

- узел: имя узла
- property: имя свойства
- метод: имя метода
- arg1, arg2, и т.д.: оператор или выражение Basic (не должны содержать символов скобок).

 **ЗАМЕЧАНИЕ:**

В данном примере между символами [и] заключаются дополнительные элементы.

Например, чтобы восстановить число строк из элемента управления LISTBOX на странице PAGE1, используется метод COUNT, связанный с данным типом элемента управления. Выполняется следующая команда:

```
{PAGE1.LISTBOX.VALUES.COUNT()}
```

Свойство типа «Таблица»

Свойства типа «Таблица» - свойства, значения которых определяются согласно следующему формату:

```
<Столбец|Столбец|Столбец|...>=<Код строки>, <Столбец|Столбец|Столбец|...>=<Код строки>, ...
```

Значения данных свойств можно просмотреть в форме таблицы:

		Столбец 1	Столбец 2	Столбец 3
Номер строки: 1	Код строки (напр.: 18)	Ячейка (1,1)	Ячейка (2.1)	Ячейка (3.1)
Номер строки: 2	Код строки (напр.: 29)	Ячейка (1.2)	Ячейка (2.2)	Ячейка (3.2)
Номер строки: 3	Код строки (напр.: 78)	Ячейка (1.3)	Ячейка (2.3)	Ячейка (3.3)

	Столбец 1	Столбец 2	Столбец 3
И Т.П.	И Т.П.	И Т.П.	И Т.П.



ЗАМЕЧАНИЕ:

Идентификатор - идентификатор текстового типа.

Пример

Рассмотрим свойство VALUES узла LISTBOX, имеющий своим значением результат запроса к таблице «Сотрудники и подразделения».

Рассматриваемым запросом возвращаются значения полей Фамилия (имя SQL: Name) и Имя (имя SQL: FirstName) для каждой записи в таблице.

Предположим, что свойство имеет следующее значение:

```
VALUES="Colombo|Gerard=32,Lubeck|Alexander=64,Daquin|William=24"
```

Данное значение можно просмотреть в форме таблицы:

		Имя	Имя
1	32	Colombo	Gerard
2	64	Lubeck	Alexander
3	24	Daquin	William

Использование глобальных переменных CurrentTable и CurrentSelection

Содержимое данных переменных можно восстановить с помощью следующего синтаксиса:

```
[CurrentTable]
[CurrentSelection]
```

В приведенной ниже таблице приведены признаки двух данных переменных:

Имя переменной	Описание переменной	Комментарий
CurrentTable	Содержит имя SQL активной таблицы во время запуска мастера. Если нет активных таблиц, она содержит пустую строку. Переменная типа String.	Это поле автоматически заполняется в Asset Manager. Пользователь не может принудительно присвоить значение.

Имя переменной	Описание переменной	Комментарий
CurrentSelection	Содержит список внутренних идентификаторов записей, выбранных во время запуска мастера, разделенных запятыми. Переменная типа String.	Данная переменная автоматически заполняется в Asset Manager>. Она содержит пустую строку, если ничего не было выбрано или если не были обновлены никакие таблицы. Пользователь не может принудительно присвоить значение.

Соединение мастеров

После выполнения мастер может использоваться для инициирования выполнения другого мастера и передавать параметры (переменные) в другой мастер. Это называется соединением мастеров.

Выполнение

Для того, чтобы мастер А инициировал запуск мастера В, его конечный узел не должен иметь свойство CHAIN. Данное свойство должно иметь значение имени SQL действия типа Помощник, которое должно выполняться, в данном случае В.

ПОДСКАЗКА:

По умолчанию установка взаимосвязи действий личного мастера доступна только для владельца (автора) действия. В связи с этим пользователи не могут вызвать (выполнить) «скрытый» мастер из доступного мастера.

Проверка для одной ссылки (Parent = 0), чтобы выбрать модели без родительских активов.

Если необходимо разрешить другим пользователям вызывать (выполнять) «скрытый» мастер из доступного мастера, воспользуйтесь ссылкой меню **Администрирование/ Параметры базы данных** и установите для параметра **Мастера/ ChainNotOwnedPrivWizards** значение Да.

Параметры

Параметры передаются в мастер В при помощи узла PARAMS мастера А. Эти параметры добавляются к параметрам узла PARAMS мастера Б. Если один и тот же параметр определяется для узла PARAMS мастера А и узла PARAMS мастера Б, параметр в мастере А преобладает над параметром в мастере В.

Функции Basic

Кроме общих функций Asset Manager (за исключением функции AmCounter), мастера принимают следующие дополнительные функции:

- AmComputeString()
- AmDecrementLogLevel()
- AmExecTransition()



ВНИМАНИЕ:

Этот интерфейс API нельзя использовать в веб-клиенте.

- AmLog()
- AmMsgBox()
- AmPagePath()
- AmProgress()
- AmRefreshProperties
- AmSetProperty
- AmUpdateDetail
- AmValueOf
- AmWizChain

Значения, возвращаемые функциями

При вызове функций Basic в скрипте мастера всегда необходимо назначать значение, возвращаемое функцией в переменную. В противном случае компилятором Basic будет возвращаться ошибка. Следующий пример не будет скомпилирован:

```
AmGetFieldLongValue(hRecord, "UserId", {lEmplDeptId})
```

Правильным скриптом будет следующий:

```
Dim IValue as Long  
IValue=AmGetFieldLongValue(hRecord, "IUserId", {IEmplDeptId})
```

Сцепление строк в скриптах на языке Basic

Скрипты мастеров могут вызывать скрипты на языке Basic.

В скриптах на языке Basic для сцепления строк используется оператор &;, а не оператор +.

Фактически, оператор + может интерпретироваться как оператор сложения, что будет вызывать ошибку при выполнении мастера.

Определение корневого узла

Корневым узлом описывается мастер в целом. Он состоит из блока общих свойств, которые могут применяться для всех мастеров, и последовательности подузлов, которой представляются объекты, содержащиеся в мастере.

Синтаксис корневого узла

Для корневого узла используется следующий синтаксис:

```
' Блок общих свойств корневого узла NAME=... IMAGE=... ... ' Определение п  
одузлов корневого узла { FINISH ... } { PAGE ... } { TRANSITION ... }
```

Свойства корневого узла

В приведенных ниже таблицах перечислены все логические и физические свойства, которые можно определить в корневом узле.

Таблица 15.2. Логические свойства корневого узла

Имя свойства=Значение	Описание свойства	Пример	Комментарий
NAME="<Имя мастера>"	Определяет название окна мастера. Свойство типа String.	NAME = "MoveWizard"	Необходимо определить значение для данного свойства. Длина имен мастеров ограничена 22 символами. Данное свойство используется для сериализации мастера: данные, относящиеся к этому мастеру, сохраняются под данным именем. Поэтому предпочтительнее, чтобы разные мастера имели разные имена.
TITLE="<Название окна>"	Определяет имя мастера. Свойство типа String.	TITLE="Move wizard"	Настоятельно рекомендуется определить значение для данного свойства.
GLOBAL=<Скрипт>	Позволяет (=TRUE) или не позволяет (=FALSE) сериализовать мастер. Если мастер сериализован, им сохраняются (в файле .ini) предварительные введенные значения для выполнения в следующий раз. Данный скрипт объявляется, но не выполняется. Для выполнения скрипта при запуске используется узел START. Свойство типа Boolean.	{GLOBAL=Dim Filter As String}	
COMMON	Хранит функции Basic, автоматически включаемые во все мастера. Данное свойство соответствует файлу gbbase.wiz (только для чтения), который вставляется в базу данных при ее создании.		

Имя свойства=Значение	Описание свойства	Пример	Комментарий
SERIALIZE=<TRUE FALSE>	Позволяет (=TRUE) или не позволяет (=FALSE) сериализовать мастер. Если мастер сериализован, им сохраняются (в файле .ini) предварительные введенные значения для выполнения в следующий раз. Свойством NAME определяется, в каком разделе файла .ini будут храниться значения. Свойство типа Boolean. Местоположение файлов .ini: ► Руководство по установке и обновлению Asset Manager, глава файлы .ini и .cfg.	SERIALIZE=TRUE	По умолчанию это свойство имеет значение «FALSE».
MODAL=<TRUE FALSE>	Определяет, является ли мастер модальным (=TRUE) или нет (=FALSE).		

Таблица 15.3. Физические свойства корневого узла

Имя свойства=Значение	Описание свойства	Пример	Комментарий
IMAGE="<Путь файла точечного рисунка>" "IMAGE16="<Путь файла точечного рисунка>"	Определяет графический файл точечного типа (.bmp) для отображения в мастере. Свойство типа String.	IMAGE="Config.bmp"	Если для данного свойства не определено никакого значения, то не будет отображаться никакого изображения. Путь файла графики зависит от папки Config в Asset Manager. Asset Manager сначала ищет изображение в базе данных. Если определить значение «IMAGE16», то это свойство будет использоваться вместе «IMAGE» при глубине цвета экрана равной 16.
WIDTH=<Ширина>	Определяет ширину по умолчанию ("<Ширина>") окна мастера. Этот параметр выражается в твипах. Свойство типа Long.	WIDTH=6000	
HEIGHT=<Высота>	Определяет высоту по умолчанию ("<Высота>") окна мастера. Этот параметр выражается в твипах. Свойство типа Long.	HEIGHT=5000	

Имя свойства=Значение	Описание свойства	Пример	Комментарий
MINWIDTH=<MinWidth>	Определяет минимальную ширину окна мастера. Значение, выраженное в твипах.		
MINHEIGHT=<MinHeight>	Определяет максимальную высоту окна мастера. Значение, выраженное в твипах.		
CTRLHEIGHT=<CtrlHeight>	Определяет высоту элемента управления, для которого установлен вертикальный размер (например, управляющего элемента TEXTBOX). Значение промежутка, выраженное в твипах.		
LABELSPACING=<labelSpacing>	Определяет промежуток между меткой элемента управления и самим элементом управления (когда название находится над элементом управления). Значение промежутка, выраженное в твипах.		
CTRLSPACING=<CtrlSpacing>	Определяет промежуток по вертикали между двумя элементами управления. Значение промежутка, выраженное в твипах.		

Имя свойства=Значение	Описание свойства	Пример	Комментарий
IMGBORDER=<Width>	Определяет промежуток по горизонтали между изображением мастера и его элементами управления. Значение, выраженное в твипах.		
NAVIGATION=<TRUE FALSE>	Отображает (=TRUE) или нет (=FALSE) в окне мастера навигационную панель, содержащую кнопки Далее и Отмена .		
CONFIRMCANCEL=<TRUE FALSE>	Отображает (=TRUE) или нет (=FALSE) сообщение подтверждения отмены.		
DEFAULTONNEXT=<TRUE FALSE>	Выбирает по умолчанию (=TRUE) кнопку Далее . Если DEFAULTONNEXT=FALSE , то по умолчанию выбирается кнопка Конец .		

Подузлы корневого узла

В таблице ниже перечислены подузлы, которые можно определять для корневого подузла. Каждый тип узла представляет собой объект.

Таблица 15.4. Подузлы корневого узла

Тип узла	Описание
PAGE	Описывает страницу мастера.

Тип узла	Описание
FINISH	Описывает последний переход с последней страницы мастера (к концу). Данный узел типа «Переход» не имеет свойства FROM и TO.
START	Содержит, например, скрипт, который должен выполняться при запуске мастера (с помощью свойства DO) и имя начальной страницы мастера (свойство TO).
PARAMS	Позволяет передавать параметры из одного мастера в другой (если заполнено свойство CHAIN подузла FINISH).
TIMER	Позволяет связать таймер со страницей мастера.

Определение узла «Страница»

Узлом «Страница» описывается страница мастера. Он состоит из блока свойств, применимых для данного узла и всех его подузлов, а также набора подузлов, которыми определяются объекты, определяемые на данной странице.

Синтаксис узла «Страница»

Для узла «Страница» используется следующий синтаксис:

```
' Declaration of the page { Page <Name of the page> ' Block properties of the page node IMAGE=... TITLE=... ' Definition of sub-nodes of the "Page" node { TRANSITION ... } { <Control type> <Control name> ... } ... }
```

Свойства узла «Страница»

В приведенных ниже таблицах перечислены все свойства, которые можно определить в узле «Страница».

Таблица 15.5. Логические свойства узла «Страница»

Имя свойства=Значение	Описание свойства	Пример	Комментарий
TITLE="<Название страницы>"	Определяет название страницы. Данное название показывается полужирным шрифтом в верхней части страницы. Свойство типа String.	TITLE="Move"	Если для данного свойства не определено никакого значения, им наследуется значение свойства TITLE корневого узла. В отличие от меток, данной строкой не поддерживается язык HTML.
ONENTER=<Скрипт>	Определяет скрипт на языке Basic, который выполняется при доступе к странице с помощью кнопок Далее или Назад . Свойство типа Boolean.	{ONENTER = AmMsgBox ("Hello") }	

Таблица 15.6. Физические свойства узла «Страница»

Имя свойства=Значение	Описание свойства	Пример	Комментарий
IMAGE="<Путь файла точечного рисунка>" IMAGE16="<Путь файла точечного рисунка>"	Определяет графический файл точечного типа (.bmp) для отображения на странице мастера. Свойство типа Boolean.	IMAGE =" C:\Images\Page1.bmp"	Если для данного свойства не определено никакого значения, им наследуется значение свойства IMAGE корневого узла. Если для данного свойства определено пустое значение, то не будет отображаться никакого изображения. Если определить значение «IMAGE16», то это свойство будет использоваться вместе «IMAGE» при глубине цвета экрана равной 16.

Подузлы узла «Страница»

Для узла «Страница» можно определить два типа подузлов:

Таблица 15.7. Подузлы узла «Страница»

Тип узла/Объект	Описание
<Тип элемента управления> <Имя элемента управления>	Определяет элемент управления, отображаемый на текущей странице.
TRANSITION <Имя перехода>	Описывает переход между текущей страницей и другой страницей мастера.
TIMER	Позволяет связать таймер со страницей мастера.

Определение узла перехода

Узлом перехода описывается переход между двумя страницами мастера. Она состоит исключительно из блока свойств.



ЗАМЕЧАНИЕ:

Переходы можно определять изнутри узла «Страница» (в таком случае, для них не требуется свойство FROM) или в корневом узле. Последний переход, приводящий к закрытию мастера, описан в узле FINISH (на уровне корневого узла) и не имеет свойств FROM и TO.

Синтаксис узла перехода

Для узла перехода используется следующий синтаксис:

' Объявление перехода { TRANSITION0 <Имя перехода> ' Блок свойств узла перехода FROM=... TO=... CONDITION=... }

Свойства узла перехода

В приведенной ниже таблице перечислены все свойства, которые можно определить в узле перехода:

Таблица 15.8. Логические свойства узла перехода

Имя свойства=Значение	Описание свойства	Пример	Комментарий
FROM="<Имя начальной страницы>"	Определяет имя начальной страницы перехода. Свойство типа String.	FROM="Page2"	Данное свойство обязательно, если переход определяется внутри корневого узла, и неприменимо, если оно определяется в узле «Страница», конечном узле или начальном узле.
TO="<Имя целевой страницы>"	Определяет имя целевой страницы перехода. Свойство типа String.	TO="Page3"	Данное свойство обязательно, если переход определяется внутри корневого узла или узла «Страница», и неприменимо, если оно определяется в конечном узле.
CONDITION=<Скрипт>	Определяет условие, которое должно выполняться для инициирования перехода. Свойство типа «скрипт», возвращающее булево значение.	CONDITION={Comment}="user"	Данное свойство недоступно в начальных узлах.
DO=<Скрипт>	Определяет скрипт, который должен выполняться во время перехода. Свойство типа Boolean.	{DO= Filter=""}	

Особенности узла перехода

Узел перехода не имеет подузлов.

Необходимость определения переходов в корневом узле

Отказ от использования переходов в узлах «Страница» позволяет создавать страницы, которые можно использовать в любых скриптах, и рационализировать написание скриптов.

Определение конечного узла

Конечным узлом описывается последний переход, который ведет к последней странице мастера. Это особый тип узла перехода, который не имеет свойства FROM и TO. Кроме данного исключения, синтаксис и свойства конечного узла идентичны синтаксису и свойствам узла перехода.

Свойство CHAIN, специфичное для конечного узла, позволяет инициировать выполнение другого мастера.

Таблица 15.9. Логические свойства конечного узла

Имя свойства=Значение	Описание свойства	Пример	Комментарий
CHAIN=<имя SQL мастера, который необходимо выполнить>	Определяет имя SQL мастера, который необходимо выполнить при завершении текущего мастера. Если данное свойство не заполнено, не будут выполняться никакие мастера. Свойство типа String	CHAIN = "Move"	
CONDITION=<Скрипт>	Определяет проверку условий кнопки Готово . Свойство типа Boolean.		
DO=<Скрипт>	Определяет скрипт, который должен быть выполнен в конце мастера. Свойство типа «скрипт», возвращающее булево значение.		



ЗАМЕЧАНИЕ:

Узел PARAMS позволяет передавать параметры следующему мастеру.

Таблица 15.10. Физические свойства конечного узла

Имя свойства=Значение	Описание свойства	Пример
SUMMARY=<TRUE FALSE>	Отображает (=TRUE) или нет (=FALSE) страницу сводки при запуске мастера. Заполнить данную страницу можно с помощью функций amLog и amProgress.	
SHOWPROGRESSBAR=<TRUE FALSE>	Отображает (=TRUE) или нет (=FALSE) индикатор выполнения на странице сводки.	
SHOWLOGLIST=<TRUE FALSE>	Отображает (=TRUE) или нет (=FALSE) журнал выполнения на странице сводки.	
LABEL="Title"	Отображает название страницы сводки.	
ISHTML=<TRUE FALSE>	Определяет вид текста заголовка. Свойство типа Boolean.	
TITLE="Title"	Определяет название страницы сводки. Название по умолчанию страницы сводки - такое же, что и корневого узла.	

Определение начального узла

Начальным узлом описывается, как запускается мастер. Это особый тип узла перехода, который не имеет свойства FROM или CONDITION. Кроме данного исключения, синтаксис и свойства начального узла идентичны синтаксису и свойствам узла перехода.

Таблица 15.11. Логические свойства начального узла

Имя свойства=Значение	Описание свойства	Пример
-----------------------	-------------------	--------

DO=<Скрипт>	Определяет скрипт, который должен быть выполнен при запуске. Свойство типа «скрипт», возвращающее булево значение.
TO="<Имя начальной страницы>"	Определяет имя первой страницы для отображения. Свойство типа String.



ЗАМЕЧАНИЕ:

Если данный узел не существует, мастер запускается с первой страницы.

Определение узла таймера

Узел таймера позволяет выполнять задачу через регулярные интервалы.

Таблица 15.12. Логическое свойство узла таймера

Имя свойства=Значение	Описание свойства	Пример
AUTO= <TRUE FALSE>	Определяет, запускается ли автоматически таймер при отображении страницы. Свойство типа Boolean.	
ENABLED= <TRUE FALSE>	Определяет, включен ли таймер (=TRUE) или нет (=FALSE). Данное свойство может использоваться для остановки или перезапуска таймера. Свойство типа Boolean.	
INTERVAL=time interval	Определяет интервал времени между двумя выполнениями таймера. Продолжительность выражена в секундах.	
TIMER="Script"	Определяется, что должно выполняться каждый раз, когда проходит интервал времени. Свойство типа «Скрипт».	

VALUE=tickcount	Число уже прошедших интервалов. Каждое свойство, зависящее от данного интервала, будет автоматически пересматриваться при каждом вхождении. Данное значение (tickcount) автоматически нарастает.
-----------------	---

Определение узлов «Длинный» и «Строка»

Узлами «Длинный» и «Строка» определяются переменные. На них можно ссылаться во всех узлах мастера. Имя переменной определяется именем узла.

Данные узлы имеют только одно, единственное свойство, тип которого зависит от узла: LONG для узла «Длинный» и STRING - для узла «Строка». Данное свойство, VALUE, позволяет определять значение переменной.

Таблица 15.13. Логические свойства узлов «Длинный» и «Строка»

Имя свойства=Значение	Описание свойства	Пример	Комментарий
VALUE=<значение>	Определяет значение переменной, чье имя - такое же, как у узла. Свойство типа Long для узла «Длинный» или String - для узла «Строка».	VALUE=12	



ЗАМЕЧАНИЕ:

Узлы «Длинный» и «Строка» могут определяться в любом узле мастера. У них нет подузлов.

Определение узла элемента управления

Элементы управления страницы для взаимодействия с пользователем. Для данной страницы можно определить столько элементов управления, сколько

необходимо. Asset Manager полностью управляет организацией элементов управления на странице. Нет необходимости самостоятельно указывать местоположение каждого из определяемых элементов управления. Узлы типа «Элемент управления» состоят исключительно из блока свойств, применимых для определяемого элемента управления.

Общий синтаксис узла элемента управления

Для узла типа «Элемент управления» используется следующий общий синтаксис:

" Объявление элемента управления { <Тип элемента управления> <Имя элемента управления> ' Свойства элемента управления ... }

Типы элементов управления и связанные свойства

Все элементы управления имеют общие свойства. Однако есть свойства, специфичные для конкретных элементов управления.

Общие свойства

В приведенной ниже таблице перечисляются дополнительные свойства, применимые для всех элементов управления.

Таблица 15.14. Логические свойства, общие для всех элементов управления

Имя свойства=Значение	Описание свойства	Пример	Комментарий
MANDATORY=<TRUE FALSE>	Делает обязательным для пользователя заполнение элемента управления для утверждения перехода. Данное свойство игнорируется, если элемент управления: ■ не показывается; ■ только для чтения; ■ отключен.	MANDATORY=TRUE	Данное свойство недоступно для элементов управления CHECKBOX и LABEL
VALUE="<Value, Value,...>"	Определяет значение по умолчанию создаваемого элемента управления. Если элемент управления отображается в мастере, это значение заменяется значением, выбранным пользователем. В зависимости от элемента управления, свойство VALUE может содержать несколько значений. Тип данного свойства зависит от типа элемента управления (булево, текстовое и так далее).	■ Для элемента управления CHECKBOX, <Value> может быть TRUE или FALSE. ■ Для элемента управления LISTBOX, где VALUE= 1,3,4; 1,3 и 4 - это значения, соответствующие строкам, выбранным в элементе управления.	Если свойство VALUE может содержать несколько значений, должно быть активировано свойство MULTISEL элемента управления (равное TRUE). В противном случае будет использоваться только значение, определенное в первом положении свойства VALUE.

Имя свойства=Значение	Описание свойства	Пример	Комментарий
PERMANENT=<TRUE FALSE>	При переходе от какой-либо страницы мастера к следующей элементы управления пропадают. Определяет, сохраняется и скрывается ли элемент управления при переходе от одной страницы к другой (=TRUE) или пропадает (=FALSE). Свойство типа Boolean.		
SERIALIZE=<TRUE FALSE>	Данное свойство позволяет сериализовать мастер на уровне элемента управления. Если сериализация корневого узла активирована (=TRUE), можно деактивизировать ее на уровне элемента управления.		По умолчанию, данное свойство имеет значение свойства SERIALIZE в корневом узле.
HELP = "help"	Данное свойство позволяет включать текст справки в формате HTML в элемент управления мастера. Данная справка вызывается нажатием сочетания клавиш SHIFT+F1.		

Таблица 15.15. Физические свойства, общие для всех элементов управления

Имя свойства=Значение	Описание свойства	Пример	Комментарий
VISIBLE=<TRUE FALSE>	Определяет, является ли элемент управления видимым (=TRUE) или нет (=FALSE). Свойство типа Boolean.	Label1.Visible=TRUE	

Имя свойства=Значение	Описание свойства	Пример	Комментарий
ENABLED = <TRUE FALSE>	Определяет, является ли элемент управления активным (=TRUE) или нет (=FALSE). Свойство типа Boolean. Свойство типа Boolean.	Choice1.Enabled=FALSE	
READONLY = <TRUE FALSE>	Определяет, только для чтения ли значение элемента управления (=TRUE) и, таким образом, не может изменяться пользователем, или оно изменяемо (=FALSE).	READONLY=TRUE	
LABEL = "<Текст метки>"	Определяет дополнительный текст, отображаемый выше или слева от элемента управления. Свойство типа String.	Choice1.Label="Select person"	Эта метка поддерживает HTML
LABELLEFT	Данное свойство позволяет поместить метку элемента управления слева от него. Использование данного свойства означает, что необходимо заполнить свойство XOFFSET. Свойство типа Boolean.		
XOFFSET	Определяет пространство, резервируемое для метки элемента управления, если она помещается слева от элемента управления (с помощью свойства LABELLEFT). Свойство типа Тип.		

Имя свойства=Значение	Описание свойства	Пример	Комментарий
ISHTML	Определяет вид текста метки. По умолчанию, вид текста - HTML. Свойство типа Boolean.		Эта метка поддерживает HTML
INDENT	Данное свойство позволяет смещать вправо элемент управления и его заголовков. Свойство типа Твип.		
YOFFSET	Определяет смещение до элемента управления и его заголовка. Свойство типа Твип.		
YOFFSET2	Определяет смещение после элемента управления и его заголовка. Свойство типа Твип.		

Элемент управления CHECKBOX

Элементом управления CHECKBOX определяется флажок.

Свойства

Кроме дополнительных свойств, общих для всех элементов управления, элементом управления CHECKBOX распознается следующее свойство:

Таблица 15.16. Свойство элемента управления CHECKBOX

Имя свойства=Значение	Описание свойства	Пример
CAPTION="<Текст>"	Определяет текст флажка. Свойство типа String. Данный текст не может быть в формате HTML и должен содержать одну единую строку. Свойство типа String.	TEXT="Identify by name"

Элемент управления COMBOBOX

Элементом управления COMBOBOX определяется однократный выбор в списке (подробном списке) предварительно определенных значений.

Свойства

Кроме дополнительных свойств, общих для всех элементов управления, элементом управления COMBOBOX распознается следующее свойство:

Таблица 15.17. Физические свойства элемента управления COMBOBOX

Имя свойства=Значение	Описание свойства	Пример	Комментарий
VALUES="<Имя=Значение, Имя=Значение, Имя=Значение,...>"	Определяет пары значений ("Имя"= "Значение") для элемента управления Combo. "Имя" - текст, отображаемый в элементе управления, "Значение" - значение, приписываемое, если "Имя" выбирается пользователем. Свойство типа String.	VALUES="Таблица активов=актив, Пользователь=пользователь"	Если пропустить «Значение», то Asset Manager присвоит его автоматически. Пример Выражение VALUES = "A,B,C" эквивалентно выражению VALUES = "A=1,B=2,C=3"

Элемент управления OPTIONBUTTONS

Элементом управления OPTIONBUTTONS определяется группа переключателей.

Свойства

Кроме дополнительных свойств, общих для всех элементов управления, элементом управления OPTIONBUTTONS распознаются следующие свойства:

Таблица 15.18. Физические свойства элемента управления OPTIONBUTTONS

Имя свойства=Значение	Описание свойства	Пример
VALUES="<Заголовок=Значение, Заголовок=Значение, Заголовок=Значение,...>"	Определяет пары значений ("Заголовок"= "Значение") для элемента управления CHOICE. "Имя" - текст переключателя, "Значение" - значение, приписываемое элементу управления, если данный переключатель выбирается пользователем. Свойство типа String.	VALUES="Таблица активов=актив, Пользователь=пользователь"
BORDER=<TRUE FALSE>	Указывает, изображается ли граница группы переключателей (=TRUE) или нет (=FALSE). Если группа переключателей заключается в рамку, текст встраивается в границу рамки. Данный текст не может быть в формате HTML или содержать несколько строк. Свойство типа Boolean.	BORDER= TRUE

Элемент управления LISTBOX

Элементом управления LISTBOX определяется список объектов, которые можно выбирать. Элементы управления LISTBOX могут быть элементами управления с несколькими столбцами.

Свойства

Кроме дополнительных свойств, общих для всех элементов управления, элементом управления LISTBOX распознаются следующие свойства:

Таблица 15.19. Физические свойства элемента управления LISTBOX

Имя свойства=Значение	Описание свойства	Пример	Комментарий
LISTHEIGHT = <Процент>	Определяет размер данного элемента управления LISTBOX по отношению к остальным элементам управления LISTBOX всего мастера. Свойство типа Long.	LISTHEIGHT=50	Если есть два элемента управления LISTBOX со значениями 10 и 20 соответственно для данного свойства, второй элемент управления будет в два раза выше, чем первый.
MULTISEL = <TRUE FALSE>	Указывает, поддерживает ли элемент управления множественный выбор (=TRUE) или нет (=FALSE). Свойство типа Boolean.	MULTISEL=TRUE	
COLTITLE = "<Столбец Столбец Столбец...>"	Определяет заголовки столбцов в списке. «Столбец» необходимо заменить на заголовков соответствующего столбца. Свойство типа String.	COLTITLE = "Фамилия Имя"	
COLWIDTH = "<Ширина Ширина Ширина...>"	Определяет размер столбцов пропорционально полному размеру элемента управления. Свойство типа String.	COLWIDTH = "50 50"	

Имя свойства=Значение	Описание свойства	Пример	Комментарий
VALUES = "<Текст Текст ...= Значение, Текст Текст ...= Значение,...>"	Определяет пары значений ("Текст Текст ..."= "Значение") для элемента управления LISTBOX. "Текст Текст .." - определяется текст, который должен отображаться в каждом из столбцов для строки элемента управления LISTBOX, "Значение" - значение, присписываемое элементу управления, если данная строка выбирается пользователем. Свойство типа String.	VALUES="Таблица активов=актив, , Пользователь=пользователь,"	Если пропустить «Значение», то Asset Manager присвоит его автоматически. Выражение VALUES="A,B,C" эквивалентно выражению VALUES="A=1,B=2,C=3" Пример Данное свойство можно заполнять непосредственно или с помощью функции AmdbGetList для записи, например: VALUES = AmDbGetList ("SELECT Name, FirstName FROM amEmplDept WHERE Name Like 'A%'", " ", ";", "=") Не следует путать свойства VALUES и VALUE.
EDITABLE="<0 1>"	Определяет, можно ли изменять текст в столбцах или нет. Свойство типа String.	EDITABLE="0 1"	
TABLE="<Имя таблицы>"	Определяет контекст применения заголовков столбцов, если они определены для свойства COLNAME.	TABLE="amEmplDept"	

Имя свойства=Значение	Описание свойства	Пример	Комментарий
COLNAME=<имя поля>	Определяет названия свойств столбцов в списке с именами SQL полей. Необходимо заполнить свойство TABLE; «Имя» замещается именем SQL поля для заголовка столбца. Если заголовок столбца заполнен с помощью свойства COLTITLE, оно имеет приоритет над свойством COLNAME, но при этом сохраняется тип поля (текст, дата и так далее).	COLNAME=NameField	ВНИМАНИЕ: Свойство COLNAME must должно быть заполнено, чтобы содержимое элемента управления LISTBOX правильно отображалось в веб-клиенте.
MULTISEL = <TRUE FALSE>	Определяет использование множественного выбора для списка. Свойство типа Boolean.	MULTISEL=1	

Таблица 15.20. Методы элемента управления LISTBOX

Имя свойства=Значение	Описание свойства	Пример	Комментарий
VALUES (i)	Возвращается содержимое строки.	a b c	
VALUES.COUNT()	Подсчитывает число строк в свойстве VALUES.	retval = {listbox1.values.count()}	
VALUES.CELL(h,v)	Возвращает содержимое ячейки, обозначенной ее координатами (по горизонтали, по вертикали).	VALUES.CELL(2,4)	
VALUES.COLUMN(i)	Возвращает содержимое (значение) столбца (i). Если значение i=0 или пусто, данной инструкцией возвращаются коды столбца.	VALUES.COLUMN(1)	

Имя свойства=Значение	Описание свойства	Пример	Комментарий
VALUES.LINE(i)	Возвращает содержимое (значение) строки (i). Если значение i=0 или пусто, данной инструкцией возвращаются коды строки.	VALUES.LINE(1)	
VALUES.SORT(iCol, bAsc)	Сортирует столбец (i) по возрастанию или нет (bAsc=1 или bAsc=0).	{ LISTBOX lb VALUES = "first,second,third" } { COMMANDBUTTON btn { CLICK = RetVal = {lb.Values.Sort(1)} } }	
VALUES (i,0)	Возвращает значение кода строки (i).		

Таблица 15.21. Обязательное логическое свойство элемента управления LISTBOX

Имя свойства=Значение	Описание свойства	Пример
TABLE=<Имя таблицы>	Имя таблицы используется для извлечения заголовков столбцов. Свойство типа String.	TABLE= amAsset
COLNAME=<Заголовок Заголовок ...>	Определяет заголовки столбцов с помощью имен SQL полей в таблице, определенной с помощью свойства TABLE. Данное свойство также позволяет определять используемый элемент управления изменением. Данный элемент управления - тот же самый, что и используемый в Asset Manager для заполнения поля. Asset Manager для определения заголовков столбцов будет сначала учитывать значения свойства COLTITLE (если существует). Свойство типа String.	COLNAME="Name FirstName"

Элемент управления LABEL

Элементом управления LABEL просто определяется метка. Данный элемент управления имеет следующее свойство:

Таблица 15.22. Физические свойства элемента управления LABEL

Имя свойства=Значение	Описание свойства	Пример	Комментарий
CAPTION=<Текст>	Содержит текст, отображаемый в метке.	CAPTION="Select a location"	

Элемент управления PROGRESSBAR

Элементом управления PROGRESSBAR определяется индикатор выполнения.

Свойства

Кроме дополнительных свойств, общих для всех элементов управления, элементом управления PROGRESSBAR распознаются следующие свойства:

Таблица 15.23. Физические свойства элемента управления PROGRESSBAR

Имя свойства=Значение	Описание свойства	Пример
MAXVALUE=<Максимальное значение>	Определяет максимальное значение, соответствующее 100% индикатора выполнения. Свойством VALUE указывается текущее значение элемента управления. Свойство типа Long.	MAXVALUE=200

Элемент управления COMMANDBUTTON

Элементом управления COMMANDBUTTON определяется кнопка действия.

Свойства

Кроме дополнительных свойств, общих для всех элементов управления, элементом управления COMMANDBUTTON распознаются следующие свойства:

Таблица 15.24. Физические свойства элемента управления COMMANDBUTTON

Имя свойства=Значение	Описание свойства	Пример
WIDTH=<Ширина>	Определяет в твипах ширину кнопки. Свойство типа Long.	WIDTH=250
HEIGHT=<Высота>	Определяет в твипах высоту кнопки. Свойство типа Long.	HEIGHT=125
CAPTION=<Текст>	Определяет текст (не в формате HTML), отображаемый в кнопке. Свойство типа String.	CAPTION="Start"
CLICK=<Скрипт Basic>	Определяет скрипт на языке Basic, который выполняется при щелчке пользователя по кнопке.	

Элемент управления DBLISTBOX

Элементом управления DBLISTBOX определяется список записей, которые можно выбирать из базы данных. Данный элемент управления может быть элементом управления с несколькими столбцами. Список, отображаемый в элементе управления, представляет собой результат частичного запроса AQL (используется только оператор WHERE) к базе данных Asset Manager.

ЗАМЕЧАНИЕ:

Свойством **VALUE** возвращается список идентификаторов (Id) выбранных строк. К значениям ячеек в списке нет доступа. Для этого необходимо или выполнить еще один запрос, или воспользоваться элементом управления типа LISTBOX.

Свойства

Кроме дополнительных свойств, общих для всех элементов управления, элементом управления DBLISTBOX распознаются следующие свойства:

Таблица 15.25. Физические свойства элемента управления DBLISTBOX

Имя свойства=Значение	Описание свойства	Пример	Комментарий
TABLE="<Имя SQL таблицы>"	Определяет таблицу, к которой относится запрос. Свойство типа String.	TABLE=amAsset	Данное свойство является обязательным.
COLNAME="<имя SQL поля или ссылки имя SQL поля или ссылки ...>"	Определяет элементы данных, извлекаемые из базы данных (идентифицируемые с помощью своих имен SQL). Свойство типа String.	COLNAME = "Name FirstName"	
COLWIDTH = "<Ширина Ширина Ширина ...>"	Определяет ширину столбцов в процентах от полного размера элемента управления DBLISTBOX. Свойство типа String.	COLWIDTH="40 60"	
LISTHEIGHT = <Процент>	Определяет размер данного элемента управления DBLISTBOX по отношению к остальным элементам управления DBLISTBOX всего мастера. Свойство типа Long.	LISTHEIGHT=50	Если есть два элемента управления DBLISTBOX со значениями 10 и 20 соответственно для данного свойства, второй элемент управления будет в два раза выше, чем первый.
TREE=<TRUE FALSE>	Отображает (=TRUE) или не отображает (=FALSE) данные в режиме дерева. Свойство типа Boolean.	TREE=TRUE	По умолчанию это свойство имеет значение «FALSE»
MULTISEL = <TRUE FALSE>	Указывает, поддерживает ли элемент управления множественный выбор (=TRUE) или нет (=FALSE). Свойство типа Boolean.	MULTISEL=TRUE	

Имя свойства=Значение	Описание свойства	Пример	Комментарий
DBLCLICK = <TRUE FALSE>	Если для данного свойства установлено значение TRUE, Asset Manager будет имитировать нажатие кнопки Далее текущей страницы при каждом двойном щелчке по строке.	DBLCLICK=FALSE	
FILTER= "<Условие>"	Определяет условие AQL WHERE для фильтрации записей, которые должны быть обработаны в запросе. Свойство типа String.	FILTER= "UserEmpId=Colombo, Gerard" "	
MAXSEL = <TRUE FALSE>	Определяет возможность (=TRUE) или невозможность (=FALSE) выбора более чем 99 объектов. По умолчанию выбор ограничен 99 объектами (=TRUE).		
SORT(iCol, bAsc)	Сортирует столбец (iCol) по возрастанию или нет (bAsc=1 или bAsc=0).	SORT(2, 0)	
[Value.].ISSELECTION()	Если выбранное содержит значения, отличные от 0 (иными словами, пользователь не сделал выбора в списке), этот метод возвращает ненулевое значение.	MANDATORY = not {dblistbox1.IsSelection()} (Переход невозможен, если пользователь ничего не выбрал в списке.)	

Имя свойства=Значение	Описание свойства	Пример	Комментарий
TABLE.LABEL(<i>(NameType)</i>)	Данным методом возвращается имя таблицы, для которой определен элемент управления. Имеются следующие типы имен: ■ 1 - Системное имя ■ 2 - Имя SQL ■ 3 - Метка по умолчанию ■ 4 - Описание (текст в справке) Свойство TABLE должно быть заполнено.	RetVal = {dblistbox1.table.label(2)}	

Элемент управления DBQUERYBOX

Элементом управления DBQUERYBOX определяется список записей, которые можно выбирать. Данный элемент управления может быть элементом управления с несколькими столбцами. Список, отображаемый в элементе управления, представляет собой результат полного запроса AQL к базе данных Asset Manager.

Свойства

Кроме дополнительных свойств, общих для всех элементов управления, элементом управления DBQUERYBOX распознаются следующие свойства:

Таблица 15.26. Физические свойства элемента управления DBQUERYBOX

Имя свойства=Значение	Описание свойства	Пример	Комментарий
QUERY="<Полный запрос AQL>"	Определяет запрос AQL, которым отображается информация, которая должна отображаться в элементе управления DBLIST. Свойство типа String.	QUERY="SELECT Name, FirstName FROM amEmplDept WHERE Location=Ariane Building"	
COLTITLE="<Столбец Столбец ...>"	Определяет заголовки столбцов в списке. Свойство типа String.	COLTITLE = "Name FirstName"	
COLWIDTH="<Столбец Столбец ...>"	Определяет размер столбцов в списке в процентах от полного размера элемента управления. Свойство типа String.	COLWIDTH = "50 50"	
LISTHEIGHT=<Процент>	Определяет размер данного элемента управления DBQUERYBOX пропорционально к остальным элементам управления DBQUERYBOX всего мастера. Свойство типа Long.	LISTHEIGHT=50	Если есть два элемента управления DBQUERYBOX со значениями 10 и 20 соответственно для данного свойства, второй элемент управления будет в два раза выше, чем первый.

Имя свойства=Значение	Описание свойства	Пример	Комментарий
TREE=<TRUE FALSE>	Отображает (=TRUE) или не отображает (=FALSE) данные в режиме дерева. Свойство типа Boolean.	TREE=TRUE	По умолчанию это свойство имеет значение «FALSE»
MAXSEL = <TRUE FALSE>	Определяет возможность (=TRUE) или невозможность (=FALSE) выбора более чем 99 объектов. По умолчанию выбор ограничен 99 объектами (=TRUE).		
MULTISEL=<TRUE FALSE>	Указывает, под-держивает ли элемент управления множественный выбор (=TRUE) или нет (=FALSE). Свойство типа Boolean.	MULTISEL=TRUE	
DBLCLICK=<TRUE FALSE>	Если для данного свойства установлено значение TRUE, Asset Manager будет имитировать нажатие кнопки Далее текущей страницы. Свойство типа Boolean.	DBLCLICK=FALSE	

Имя свойства=Значение	Описание свойства	Пример	Комментарий
[Value.]ISSELECTION()	Если выбранное содержит значения, отличные от 0 (иными словами, пользователь не сделал выбора в списке), этот метод возвращает ненулевое значение.		

Модификация максимального количества записей, выдаваемых запросом элемента управления

По умолчанию, максимальное количество выдаваемых записей установлено на 200.

Это предотвращает негативное влияние на работу веб-клиента вследствие отображения слишком большого количества записей.

Если было достигнуто максимальное количество записей, для пользователя отображается сообщение.

Для модификации значения по умолчанию:

- 1 Отредактируйте файл aamapi52.ini на каждом сервере Asset Manager Web Service.
- 2 В разделе [Option] модифицируйте или добавьте следующую строку:

/ACWeb/WizQueryBoxMaxLoad=1 <Maximum number of records>

Элемент управления DBEDIT

Элемент управления DBEDIT создает элемент управления, идентичный используемому для заполнения поля в базе данных Asset Manager. Данный элемент управления отличается в соответствии с каждым типом поля (даты, денежного и так далее).



ЗАМЕЧАНИЕ:

Кнопка увеличения  в данном элементе управления позволяет выбирать значения, которые фактически присутствуют в базе данных, но можно ввести другое значение.

Для данного элемента управления свойство VALUE имеет «Вариант» (он зависит от элемента управления).

Свойства

Кроме дополнительных свойств, общих для всех элементов управления, элементом управления DBEDIT распознаются следующие свойства:

Таблица 15.27. Физические свойства элемента управления DBEDIT в обычном режиме

Имя свойства=Значение	Описание свойства	Пример
TABLE=<Имя SQL таблицы>	Имя SQL начальной таблицы. Свойство типа String.	TABLE="amAsset"
FIELD=<Имя SQL поля>	Имя AQL поля, используемого для элемента управления. Свойство типа String.	FIELD="seAcquMethod"
TABLE.LABEL([iNameType])	Данным методом возвращается имя таблицы данных. Имеются следующие типы имен: ■ 1 - Системное имя ■ 2 - Имя SQL ■ 3 - Метка по умолчанию ■ 4 - Описание (текст в справке) Свойства TABLE и NAME должны быть уже заполнены.	► Физические свойства элемента управления DBQUERYBOX [стр. 325]
FIELD.LABEL([iNameType])	Данным методом возвращается имя указанного поля. Имеются следующие типы имен: ■ 1 - Системное имя ■ 2 - Имя SQL ■ 3 - Метка по умолчанию ■ 4 - Описание (текст в справке) Свойство TABLE должно быть заполнено.	

Элемент управления DBTABLE

Элементом управления DBTABLE создается элемент управления для ввода таблицы в базу данных Asset Manager.

Свойства

Данный элемент управления не имеет никаких дополнительных свойств.

Элемент управления DBPATH

Элементом управления DBPATH создается элемент управления для ввода пути базы данных Asset Manager.

Свойства

Кроме дополнительных свойств, общих для всех элементов управления, элемент управления DBPATH должен иметь следующее обязательное свойство:

Таблица 15.28. Обязательное логическое свойство элемента управления DBPATH

Имя свойства=Значение	Описание свойства	Пример
TABLE=<Имя SQL таблицы>	Имя таблицы, из которой необходимо выбрать путь. Свойство типа String.	TABLE= amAsset

Элемент управления LINKEDIT

Элементом управления LINKEDIT создается элемент управления для ввода ссылки в базу данных Asset Manager.

Свойства

Кроме дополнительных свойств, общих для всех элементов управления, элемент управления LINKEDIT имеет следующие свойства:

Таблица 15.29. Логическое свойство элемента управления LINKEDIT

Имя свойства=Значение	Описание свойства	Пример/Комментарий
TABLE=<Имя SQL таблицы>	Имя таблицы, из которой необходимо выбрать ссылку. Свойство типа String.	TABLE="amAsset"
FILTER=<оператор WHERE запроса AQL>	Определяет фильтр AQL. Свойство типа String.	Данное свойство не является обязательным.
LINK=<имя SQL ссылки>	Имя SQL ссылки из таблицы, определенной в свойстве TABLE. Дополнительное свойство	LINK="POrLine"

Имя свойства=Значение	Описание свойства	Пример/Комментарий
ZOOM=<TRUE FALSE>	Отображает (=TRUE) или нет (=FALSE) инструмент для увеличения. Это свойство применяется только в том случае, когда мастер не является модальным (свойство MODAL=FALSE у корневого узла).	
SRCCHOICE=<TRUE FALSE>	Отображает (=TRUE) или нет (=FALSE) значок  . Это свойство применяется только в том случае, когда мастер не является модальным (свойство MODAL=FALSE у корневого узла).	
TABLE.LABEL([iNameType])	Данным методом возвращается имя исходной таблицы ссылки. Имеются следующие типы имен: ■ 1 - Системное имя ■ 2 - Имя SQL ■ 3 - Метка по умолчанию ■ 4 - Описание (текст в справке) Свойство TABLE должно быть заполнено.	► Физические свойства элемента управления DBQUERYBOX [стр. 325]
LINK.LABEL([iNameType])	Данным методом возвращает-ся имя ссылки. Имеются следующие типы имен: ■ 1 - Системное имя ■ 2 - Имя SQL ■ 3 - Метка по умолчанию ■ 4 - Описание (текст в справке) Свойства TABLE и LINK должны быть уже заполнены.	

Элемент управления TEXTBOX

Элементом управления TEXTBOX создается элемент управления для ввода текста.

Свойства

Кроме дополнительных свойств, общих для всех элементов управления, элемент управления TEXTBOX может иметь следующее свойство:

Таблица 15.30. Физическое свойство элемента управления TEXTBOX

Имя свойства=Значение	Описание свойства	Пример
MULTILINE=<число>	Для данного свойства устанавливается значение 0, если элемент управления TEXTBOX однострочный, или числовое значение, выраженное в процентах от отображаемой высоты элемента управления, если элемент управления - многострочный.	MULTILINE=50
PASSWORD=<TRUE FALSE>	Данными свойствами скрывается (=TRUE) или отображается (=FALSE) введенный текст.	

Элемент управления CHART

Элемент управления CHART позволяет отображать графики. Это можно составить из нескольких последовательностей.

Свойства

Кроме дополнительных свойств, общих для всех элементов управления, элемент управления CHART может иметь следующие свойства:

Таблица 15.31. Логические свойства элемента управления CHART

Имя свойства=Значение	Описание свойства	Пример
SERIES	Определяет список имен линий, разделенных вертикальными чертами. Данный список не должен быть пустым, в противном случае на графике ничего не будет показываться.	SERIES="purchase price sale price"
VALUES	Определяет числовые значения линий на графике. Двумерная таблица	VALUES="1 2,1 4"

Имя свойства=Значение	Описание свойства	Пример
FORMAT	Определяет тип данных: ■ Длинное целое значение (long) ■ Double ■ Число (number) ■ Процент (percent)	
SERIE	Число для выбранной строки.	
INDEX	Свойство CHART должно быть в интерактивном режиме (=TRUE). Число для выбранного столбца. Свойство CHART должно быть в интерактивном режиме (=TRUE).	
CLICK	При щелчке на графике вызывается скрипт данного свойства.	

Таблица 15.32. Физические свойства элемента управления CHART

Имя свойства=Значение	Описание свойства	Пример
MODE	Определяет тип графика: ■ MODE=0: вертикальная гистограмма ■ MODE=1: горизонтальная гистограмма ■ MODE=2: сегменты ■ MODE=3: цикл (круговая диаграмма)	
LABELS	Определяет заголовки столбцов.	январь февраль
3D	Определяет, является ли график трехмерным (=TRUE) или нет (=FALSE).	
COLORS	Определяет список цветов для каждой линии. Объекты в списке разделяются вертикальной чертой. Значение RGB в десятичной форме.	255 16777215 16711680 Отображаются цвета голубой, черный и красный.
INTERACTIVE	Определяет, является ли график интерактивным (=TRUE) или нет (=FALSE) (другими словами, является ли он активным, когда по нему проходит курсор).	
POPUP	Отображает (=TRUE) или нет (=FALSE) контекстное меню.	
BACKGROUND	Определяет, есть ли фон у изображения (=TRUE) или нет (=FALSE).	

Имя свойства=Значение	Описание свойства	Пример
BACKIMAGE	Определяет путь изображения, используемого как фон для графика. Для отображения фонового изображения должно быть утверждено свойство BACKGROUND (=TRUE).	
STACKED	Определяет, с накоплением ли гистограмма графика (=TRUE) или нет (=FALSE).	
CHARTHEIGHT	Определяет размер элемента управления CHART по отношению к остальным элементам управления всего мастера.	
CAPTION	Отображает название.	
ELEVATION	Определяет угол возвышения в градусах для трехмерной круговой диаграммы.	
ROTATION	Определяет угол вращения для круговой диаграммы. Значение, выраженное в градусах.	
DISPLAYLABELS	Отображает (=TRUE) или нет (=FALSE) заголовки столбцов (LABELS).	
DISPLAYSLEGEND	Отображает (=TRUE) или нет (=FALSE) обозначения линий.	

Элемент управления FILEEDIT

Данным элементом управления отображается диалоговое окно, позволяющее сохранять или загружать файл или папку.



ВНИМАНИЕ:

Этот элемент управления не работает в веб-клиенте.

В веб-клиенте этот элемент управления появляется как элемент управления **EDIT, в который нельзя вводить значения.**

Обзор файлов нельзя запустить с помощью значка.

Таблица 15.33. Свойства элемента управления FILEEDIT

Имя свойства=Значение	Описание свойства	Пример
OPENMODE	Определяет тип диалогового окна. ■ OPENMODE=1: открывает файл. ■ OPENMODE=2: сохраняет файл. ■ OPENMODE=4: открывает папку. ■ OPENMODE=8: сохраняет папку.	
FILTERS	Определяет критерии отображения файлов, перечисленных в диалоговом окне.	(*.txt) *.txt (*.scn) *.scn
DEFEXT	Определяет расширение имени по умолчанию.	(*.*scn) *.scn

Элемент управления TICKEDIT

Данный мастер позволяет вставить таймер.

Таблица 15.34. Свойства элемента управления TICKEDIT

Имя свойства=Значение	Описание свойства	Пример
VALUE	Представление (в форме строки) параметров, определенных пользователем в таймере.	
LISTHEIGHT	Определяет размер элемента управления TICKEDIT по отношению к остальным элементам управления всего мастера. Свойство типа Long.	

Элемент управления CALENDAR

Данный мастер позволяет вставить календарь.

Элемент управления TIMESPANEDIT

Данный элемент контроля позволяет вставить область изменения временного диапазона.

Элемент управления NUMBOX

Данный элемент управления позволяет вставить элемент управления числового типа.

Таблица 15.35. Свойства элемента управления NUMBOX

Имя свойства=Значение	Описание свойства	Пример
MINVALUE	Определяет минимальное значение числа. По умолчанию - бесконечное значение.	
MAXVALUE	Определяет максимальное значение числа. По умолчанию - бесконечное значение.	
FORMAT	Определяет формат числа: ■ LONG в формате, определенном на панели управления используемой операционной системы. ■ RAWLONG ■ DOUBLE в формате, определенном на панели управления используемой операционной системы.	

Элемент управления COMBOEDIT

Данный элемент управления позволяет вставить элемент управления типа раскрывающегося списка.

Таблица 15.36. Свойства элемента управления COMBOEDIT

Имя свойства=Значение	Описание свойства	Пример
VALUES	Определяет пары значений. "Имя" - текст, отображаемый в элементе управления, "Значение" - значение, приписываемое, если "Имя" выбирается пользователем. Свойство типа String.	

Элемент управления DATETIMEEDIT

Элемент управления DATETIMEEDIT

Таблица 15.37. Свойства данного элемента управления

Имя свойства=Значение	Описание свойства	Пример
FORMAT	Определяет формат элемента управления: ■ Дата ■ Время ■ Дата и время Формат элемента управления зависит от определения системных параметров пользователем.	2002/02/07 13:37:19 2002/02/07 13:37:19

Использование графического редактора

Asset Manager позволяет создавать мастера с помощью встроенного графического редактора. Редактор предназначен для того, чтобы упростить и ускорить процесс создания мастера. Это не означает, что он заменяет язык написания скриптов, понимание которого является существенным для проектирования мастеров.

- Обзор интерфейса
- Создание нового узла
- Редактирование свойств узла
- Компиляция, выполнение и отладка мастера



ЗАМЕЧАНИЕ:

Чтобы использовать графический редактор, создаваемое или изменяемое действие должно быть действием типа «Мастер» (имя SQL: `seActionType`).

Обзор интерфейса

Для доступа к графическому редактору выберите пункт меню Инструменты/Действия/Изменить. Графический редактор показывается на вкладке Мастер сведений о действии. В его состав входят три компонента:

- панель инструментов, содержащая наиболее часто используемые функции;
- раздел Иерархия, которым показывается древовидная структура мастера;
- раздел, в котором перечисляются свойства узла, выбранного в иерархии.

Панель инструментов

Панель инструментов позволяет непосредственно активизировать команды изменения. Если оставить указатель мыши на элементе на короткий промежуток времени, отображается всплывающая подсказка, описывающая значок.

Команды редактирования

Доступны четыре команды редактирования:

-  переключение редактора между текстовым и графическим режимом;
-  перемещение узла на один узел вверх внутри его родительского узла;
-  перемещение узла на один узел вниз внутри его родительского узла;
-  удаление выбранного узла.

Команды выполнения и отладки

Данные команды позволяют компилировать, отлаживать и выполнять скрипт.

Рисунок 15.2. Кнопка выполнения и отладки



Инструмент поиска

Панель инструментов включает инструмент поиска, который может использоваться для поиска строки символов в древовидной структуре мастера (данный элемент управления непосредственно вызывается при нажатии сочетания клавиш Ctrl+F).

Щелкните данную зону и введите текст, который необходимо найти. В случае успеха Asset Manager перемещает выделение на найденное вхождение (клавиша F3 и сочетание клавиш Shift+F3 позволяют найти следующее и предыдущее вхождения соответственно).



ЗАМЕЧАНИЕ:

В текстовом режиме поиск является полнотекстовым. В графическом режиме поиск выполняется только по именам свойств.

Древовидная структура мастера

В левой части графического редактора показывается древовидная структура мастера.

Когда в древовидной структуре выбирается какой-либо узел, Asset Manager отображает в правой части экрана список свойств, связанных с данным узлом.

Список свойств, соответствующих выбранному узлу.

Правая часть экрана позволяет вводить значения для свойств узла.

Каждое свойство имеет фиксированное значение или скрипт. Используются следующие цветовые коды.

- Если свойством используется свое значение по умолчанию, его имя и значение отображаются серым цветом. Можно определить другое значение для данного свойства. Оно будет отображаться черным цветом.
- Если свойством используются значение или скрипт, определенные пользователем, его имя и значение отображаются черным цветом.
- Если свойство является обязательным, его имя и значение отображаются красным цветом.
- Измененные значения отображаются голубым цветом.

Создание нового узла

В данном разделе описываются операции, которые можно выполнять для узлов. Панель инструментов позволяет перемещать узел вверх или вниз или удалять его. Здесь будет описано создание нового узла.



ЗАМЕЧАНИЕ:

Можно также перемещать узел вверх или вниз или удалять его с помощью контекстного меню. В таком случае необходимо щелкнуть выбранный узел правой кнопкой мыши.

Чтобы создать узел, необходимо сначала выбрать его родительский узел. Например, чтобы создать новый узел «Страница», необходимо сначала щелкнуть корневой узел. После выбора родительского узла следует щелкнуть его правой кнопкой мыши, чтобы открыть контекстное меню. В пункте меню «Создать» сгруппированы вместе узлы, которые можно создать.

Asset Manager вставляет узел в древовидную структуру мастера.

Редактирование свойств узла

После создания узла можно назначить значения свойствам данного узла. Это можно сделать в правой части редактора.

Есть два способа определить значение свойства:

- ввод фиксированного значения;
- определение скрипта.



ЗАМЕЧАНИЕ:

Скрипт всегда имеет приоритет над фиксированным значением. Если назначить свойству и скрипт, и значение, Asset Manager будет игнорировать фиксированное значение и интерпретировать скрипт.

Назначение свойству фиксированного значения

Щелкните непосредственно столбец «Значение» соответствующего свойства. В соответствии с типом данных, принимаемых свойством (текстовые, булевы, число с удвоенной точностью и так далее), Asset Manager позволяет выбирать из списка возможных значений или заполнять область изменения текста.

Назначение скрипта свойству

Выберите свойство, для которого необходимо назначить скрипт. Сам скрипт вводится в поле Скрипт под списком свойств.



ЗАМЕЧАНИЕ:

При выборе в контекстном меню пункта Восстановить значение по умолчанию (после щелчка свойства правой кнопкой мыши), Asset Manager отменяет фиксированное значение или скрипт и возвращает свойству его значение по умолчанию. Данная операция возможна только для свойств, для которых были определены пользователем значение или скрипт (такие свойства отображаются черным цветом).

Компиляция, выполнение и отладка мастера

Можно запустить мастер, щелкнув кнопку  на панели инструментов редактора. Все обнаруженные при выполнении ошибки отображаются в окне истории ошибок (доступном из встроенного отладчика мастеров). Нажатием сочетания клавиш Shift+F9 можно прервать выполнение (если мастер модальный) и активизировать отладчик.

Таким образом можно упростить исправление ошибок в мастере.



ЗАМЕЧАНИЕ:

Кнопка выполнения недоступна, если мастер контекстный.

Пример создания мастера

Чтобы проиллюстрировать теоретическую часть программирования мастера, будет создан мастер «Перемещение». Связанный с действием типа «База данных», он упрощает процесс перемещения пользователя и связанных активов из одного местоположения в другое. Шаг за шагом описывается создание такого мастера. Рекомендуется создать такой мастер самостоятельно и рассматривать данный раздел как руководство в случае какой-либо проблемы.

- 1 Пример создания мастера
- 2 Шаг 1 - Анализ ваших потребностей
- 3 Шаг 2 - Определение того, как будет организован мастер

Пример создания мастера

Назначение данного мастера - перемещение активов из одного местоположения в другое. Для этого необходимо:

- 1 Указать активы для перемещения.
- 2 Выбрать новое местоположение этих активов.

Указание активов для перемещения

Есть три способа указания активов для перемещения:

- указание в соответствии с их пользователем. После выбора пользователя необходимо выбрать активы для перемещения;
- указание активов для перемещения непосредственным выбором их записей в таблице активов;
- указание активов для перемещения в соответствии с их местоположением. Сначала выбирается местоположение, затем перемещается актив из данного местоположения.



ЗАМЕЧАНИЕ:

Следовательно, необходимо создать страницу выбора пользователя, чтобы пользователь выбрал метод отбора активов для перемещения.

Выбор нового местоположения

Чтобы выбрать новое местоположение для активов, просто выберите запись в таблице местоположений.

Шаг 1 - Анализ ваших потребностей

В соответствии с потребностями, определенными в шаге 1, необходимо определить организацию мастера. Т. е.:

- 1 Число страниц.
- 2 Связи между различными страницами.
- 3 Содержимое каждой из этих страниц.

ЗАМЕЧАНИЕ:

Как уже упоминалось в шаге 1, необходимо создать страницу выбора. Данная страница будет первой страницей мастера. Она будет называться "AssetSelectionType".

Далее определяется, как организован скрипт, с помощью следующей диаграммы:

Рисунок 15.3. Мастер - пример организации



Далее с помощью данной блок-схемы можно определить переходы между страницами:

Страница	Может вести на страницы
AssetSelectionType	SelectAsset, SelectUser, SelectLocation
SelectAsset	SelectNewLocation
SelectUser	SelectAsset
SelectLocation	SelectAsset
SelectNewLocation	Нет

Далее определяется содержимое страниц. Это означает элементы управления, которые позволяют пользователю делать выбор:

Страница	Назначение данной страницы	Используемый элемент управления
AssetSelectionType	Позволяет пользователю выбирать между тремя возможностями	Элемент управления CHOICE
SelectAsset	Позволяет пользователю выбрать активы из списка записей в таблице активов.	Элемент управления ADBLIST
SelectUser	Позволяет пользователю выбрать из таблицы подразделений и сотрудников пользователя, активы которого необходимо переместить.	Элемент управления ADBLIST
SelectLocation	Позволяет пользователю выбрать текущее местоположение в таблице местоположений.	Элемент управления ADBLIST
SelectNewLocation	Позволяет пользователю выбрать новое местоположение в таблице местоположений.	Элемент управления ADBLIST

Шаг 2 - Определение того, как будет организован мастер

Данный шаг заключается в написании скрипта мастера. Для этого используются описания структуры каждого из узлов мастера. Ниже приведен исходный код мастера «Перемещение». Данный программный код представляет собой не единственный способ написания такого мастера. Есть много других способов написания мастера, выполняющего ту же самую задачу.

```

=====
==== NAME = "Move" TITLE = "Move user" VERSION = "699" ;=====
=====
===== ;Ask which user to move. By default, use selection in amEmplDept if
context is on this table ;=====
===== { PAGE pgUser TITLE = "C
hoose the persons who are moving" { DBLISTBOX Users COLNAME = "Name|Fi

```

```

rstName" COLWIDTH = "50|50" DBLCLICK = 1 LABEL = "Persons to move" MULTISEL = 1 TABLE = "amEmplDept" { VALUE = if [CurrentTable] = "amEmplDept" then RetVal = [CurrentSelection] else RetVal = "" end if } VISIBLE = 1 } { TRANSITION trPersonToNewLoc TO = "pgNewLoc" } }

;=====
;===== ;Ask for new location ;=====
;=====
{ PAGE pgNewLoc TITLE = "Choose the new location" { STRING UserName VALUE = AmDbGetString("SELECT FirstName + ' ' + Name FROM amEmplDept WHERE IEmplDeptId IN (" & {pgUser.Users} & ")") } { LABEL LABEL1 CAPTION = "Users(s): " & {UserName} } { DBLISTBOX NewLocId COLNAME = "Name" COLWIDTH = "100" DBLCLICK = 1 TABLE = "amLocation" VALUE = "-1" } { TRANSITION trNewLocToAssets TO = "pgRecap" } }

;=====
;===== ;Recapitulation ;=====
;===== { PAGE pgRecap TITLE = "Recapitulation" { LISTBOX Users COLTITLE = "Name" COLWIDTH = "100" LABEL = "Persons to move" MANDATORY = 0 MULTISEL = 1 READONLY = 1 VALUE = "" VALUES = AmDbGetList("SELECT FullName FROM amEmplDept WHERE FullName LIKE LikeParam(amEmplDept_2:FullName) + '%' AND amEmplDept_2:IEmplDeptId IN(" & {pgUser.Users} & ");", "|", ",", "=", ")"} }

;=====
;===== ;Finish ;=====
;===== { FINISH FINISH { DO = On Error Goto ErrHandler Dim IErr as long

dim hRecord as Long

dim iEmplCount as Integer iEmplCount = {pgRecap.Users.VALUES.Count()} dim iMax as Long iMax = iEmplCount

dim ILocald as long ILocald = {pgNewLoc.NewLocId}

IErr = amStartTransaction()

dim i as Integer For i = 1 To iEmplCount IErr = AmProgress((100 * i) / iMax) IErr = AmLog("Moving the employee" + {pgRecap.Users.VALUES(i,1)}) hRecord = AmGetRecordFromMainId("amEmplDept", {pgRecap.Users.VALUES(i,0)}) If hRecord <> 0 then IErr = AmSetFieldLongValue(hRecord, "ILocald", ILocald) IErr = AmUpdateRecord(hRecord) IErr = AmReleaseHandle(hRecord) End If Next i

```

```
lErr = amCommit()
```

```
RetVal = 0 Exit Function
```

```
ErrorHandler: On Error Goto 0 AmLog(AmLastError() & " - " & AmLastErrorMsg())  
AmLog("The transaction has been cancelled") RetVal = 1 Exit function } SUMMA  
RY = 1 }
```

Часто задаваемые вопросы

Назначение данной главы - дать ответы на вопросы, которые, скорее всего, возникнут при создании мастера.

Вопрос

Следующий код не работает:

```
{lbxMyListBox.Values.Count}
```

Ответ

Необходимо ввести в синтаксис метода открывающую и закрывающую круглые скобки. Вот исправленный код:

```
{lbxMyListBox.Values.Count()}
```

Вопрос

Следующий код не работает:

```
{lbxMyListBox.Line(lRow)}
```

Ответ

Метод LINE связан со свойством VALUES элемента управления LISTBOX. Вот исправленный код:

```
{lbxMyListBox.Values.Line(lRow)}
```


Вопрос

Следующий код не работает:

```
{lboxMyListBox.Values.Line({lboxTmp})}
```

Ответ

В методе нельзя использовать ссылочное свойство. В данном случае необходимо записать:

```
Dim lRow As Long  
lRow = {lboxTmp}  
{lboxMyListBox.Values.Line(lRow)}
```

Вопрос

Не работает следующий код, которым свойству присваивается фиксированное значение:

```
{Property} = 123
```

Ответ

Чтобы присвоить свойству какое-либо значение, необходимо использовать функцию AmSetProperty(), как показано ниже:

```
Dim irc as Integer irc= AmSetProperty("Property", 123)
```

ЗАМЕЧАНИЕ:

Необходимо обязательно восстановить возвращаемый код (в данном примере - irc), даже если его не нужно использовать.

Вопрос

При выполнении мастера, которым создается актив в базе данных, появляется следующее сообщение об ошибке:

```
12001 - У вас нет прав доступа для записи
```

Данное сообщение появляется, даже если пользователь подключается как администратор.

Ответ

Данное сообщение появляется при попытке получить доступ для записи за пределами узла FINISH.DO. Мастер выполняет следующее:

- 1 Собирает информацию через последовательность страниц (доступ для записи запрещен даже для администратора Asset Manager).
- 2 Выполняет скрипт, содержащийся в узле FINISH.DO (доступ для записи разрешается в соответствии с правами доступа пользователя).

Вопрос

Сообщения об ошибке, появляющиеся при выполнении мастера, иногда оказываются неполными.

Ответ

Нажмите сочетание клавиш SHIFT+F9, чтобы отобразить отладчик. Сообщения об ошибке в окне истории часто более подробны.

Вопрос

Когда на странице мастера используется управляющий элемент DBLISTBOX, ухудшается производительность. Это нормально?

Ответ

Данная проблема возникает, когда управляющий элемент DBLISTBOX используется вместе с фильтром. В данном случае, при каждом изменении выбора отправляется запрос к базе данных, чтобы проверить, что выбор соответствует фильтру. Такой дополнительный запрос не выполняется, если выбор устанавливается пользователем.

Вопрос

Как разрешить или запретить изменение некоторых столбцов в элементе управления LISTBOX.

Ответ

Необходимо использовать свойство EDITABLE данного элемента управления. Значение, назначаемое данному свойству, представляет собой строку, состоящую из нулей (0) и символов, разделенных знаком вертикальной черты

«|», который используется как разделитель столбцов. 0 определяет столбец как «неизменяемый», 1 - как «изменяемый». Если какое-либо значение пропущено, соответствующий столбец будет неизменяемым, в приведенном ниже примере изменяемыми являются столбцы 2 и 4.

```
EDITABLE = "|1||1"
```

Вопрос

Как сделать так, чтобы мастером открывался подробный экран?

Ответ

Необходимо использовать вызовы DDE (с помощью функции) внутри мастера. Мастер не должен быть модальным. Ниже приведен пример того, как открывать таблицу активов внутри мастера:

```
Dim irc as Long  
irc = AmActionDDE("am", "Asset Manager", "OpenTable(amAsset)")
```

Вопрос

В чем разница между свойствами COLNAME и COLTITLE элемента управления LISTBOX?

Ответ

Заголовки столбцов в элементе управления LISTBOX могут определяться автоматически или вручную.

- Свойство COLNAME, связанное со свойством TABLE, позволяет автоматически определять заголовки столбцов в элементе управления LISTBOX с помощью меток полей из базы данных.
- Заголовки столбцов определяются свойством COLTITLE, если оно заполнено. Если данное свойство не определено, заголовки столбцов будут определяться свойством COLNAME.

Таким образом, следующим примером:

```
...  
TABLE = "amEmplDept"  
COLNAME = "Name||FirstName"  
COLTITLE = "|A|B"  
...
```

отображаются следующие метки в столбцах элемента управления LISTBOX:
Name, A, B.

Свойством COLNAME также определяется тип управляющего элемента, используемого, когда значения столбца являются изменяемыми.

16 Новости

В данной главе объясняется, как рассылать новости и управлять ими с помощью Asset Manager.

Доступ к списку новостей возможен через меню **Инструменты/Новости**. Бегущая строка новостей активизируется или деактивизируется с помощью меню **Окна/Отобразить бегущую строку новостей** или кнопки  на панели инструментов.

Определение позиции новостей

Позиция новостей - часть информации, для которой необходима рассылка назначенной группе лиц в течение указанного периода времени.

Данные лица относятся к группам сотрудников.

В общем случае информация такого типа является актуальной в течение короткого периода времени.

Пример элемента новостей: «Сервер XXX будет отключен для обслуживания между 11.00 и 12.00 10 февраля 2009».

Обзор новостей

Создание позиции новостей

Пользователи с правами на таблицу «Новости» могут создавать новости с помощью меню **Инструменты/Новости**.

В сведениях о новостях будут показываться:

- Сообщение
- Список рассылки сообщения.
- Период действительности сообщения.

Чтение новостей

Бегущая строка новостей позволяет любому пользователю читать сообщения, рассылаемые группе, к которой он принадлежит.

Важность позиций новостей

Чтобы определить важность сообщения, необходимо заполнить поле **Важность** (имя SQL: seSeverity) в сведениях о новостях.

Каждый уровень важности связывается с каким-либо цветом, который выбирается в поле **Цвет** (имя SQL: IColor) и который затем появляется в бегущей строке новостей.

Сообщение для рассылки

Текст для рассылки содержится на вкладке **Сообщение** сведений о новостях. Сообщение может содержать до 255 символов.

Список рассылки новостей

Вкладка **Рассылка** сведений о новостях содержит список групп сотрудников, которые могут читать элементы новостей.

Параметр **Все группы сотрудников** (SQL: bAllGroups)

При выборе данного параметра сообщение будет видно для всех членов групп сотрудников.

В противном случае сообщение будут видеть только члены выбранных групп сотрудников (в списке на вкладке **Рассылка**).

Для добавления, удаления, просмотра или изменения групп сотрудников в списке рассылки используются кнопки ,  и .

Параметр **Включить подгруппы** (SQL: bChildGrps)

Сообщения могут рассылаться всем группам сотрудников или выбранным группам. Если выбран данный параметр (который выбирается по умолчанию), сообщение будет показываться для набора подгрупп выбранных групп. Это является следствием иерархического устройства групп сотрудников.

Отображение новостей

Для просмотра новостей:

- 1 Активизируйте бегущую строку новостей.
- 2 Для прокрутки сообщений используйте кнопки в бегущей строке новостей.

Можно определять:

- цвета отображения сообщений в зависимости от их важности;
- режим автоматического обновления бегущей строки новостей.

Активизация бегущей строки новостей

Активизировать бегущую строку новостей может любой пользователь Asset Manager. При этом пользователи будут видеть рассылку новостей для той группы сотрудников, к которой они принадлежат. Бегущую строку новостей можно активизировать/деактивизировать двумя способами:

- с помощью меню **Окно/Отобразить бегущую строку новостей**;
- с помощью значка  на панели инструментов.

Таблица 16.1. Кнопки бегущей строки новостей

	Нажмите эту кнопку, чтобы прочитать новые сообщения, не дожидаясь, пока бегущая строка новостей остановит их в своем следующем определенном интервале (через меню Изменить/Параметры).
	Нажмите эту кнопку для отображения предыдущего сообщения.
	Нажмите эту кнопку для отображения следующего сообщения.
	Нажмите эту кнопку, чтобы прервать или продолжить прокрутку новостей в бегущей строке.

Указатель

- *= (AQL), 51
- < (AQL), 51
- <> (AQL), 51
- =! (AQL), 51
- = (AQL), 51
- =* (AQL), 51
- =< (AQL), 51
- = ALL (AQL), 51
- = ANY (AQL), 51
- = SOME (AQL), 51
- > (AQL), 51
- >= (AQL), 51
- Автоматическое действие (операции рабочих процессов), 184
- Адрес - действия типа «Обмен сообщениями», 140
- База данных
 - (См. также Сохранение истории)
 - Подготовка, 75
 - Подключение, 74
 - Создать, 74
- Библиотеки скриптов, 237
- Важность позиций новостей, 350
- Внешние ключи - запросы, 41
- Внешние объединения
 - Операторы, 37
 - Число, 35
- Вопрос (операции рабочих потоков), 182
- Восстановить значение по умолчанию (меню), 339
- Времена - скрипты, 239
- Выполнить (кнопка), 146
- Выражения - запросы, 49
- Вычислить (кнопка), 145
- Вычисляемые поля, 271-279
 - (См. также Скрипты)
- База данных - хранящиеся значения, 271
- Запросы, 42
- Имя SQL - ограничения, 276
- Использование, 277
- Контекст использования, 276
- Определение, 271
- Полезность, 272
- Права пользователя, 277
- Создание, 272
- Списки, 277
- Ссылки, 278
- Типы, 272
 - Используемый язык, 276
 - Поле приложения, 273

- Рекомендации, 275
- Фильтры, 278
- Формулы вычислений, 276
- Группы выполнения рабочих процессов, 204 , 155
- Данные - экспорт (См. Экспорт)
- Даты
 - Запросы, 66
 - Скрипты, 239
- Действие (тип действия), 136 , 136
- Действие пользователя (операции рабочих процессов), 183
- Действия, 129-148
 - (См. также Мастера)
 - Вкладка «Обмен сообщениями», 139
 - Вкладка DDE, 138
 - Выполнение, 146
 - Исполняемый тип
 - Выполнение, 147
 - Кнопки - связывание, 148
 - Обмен сообщениями, 150
 - Определение, 130
 - Переменные, 145
 - Печать - контекст, 135
 - Примеры, 141
 - DDE, 142
 - Исполняемое, 141
 - Обмен сообщениями, 142
 - Скрипт, 143
 - Создание, 132
 - Тесты, 145
 - Типы, 132
 - Функциональная область, 131
- Декларативный шаблон - мастера, 289
- Длинный (узел мастера), 308
- Дочерние объекты, 283
- Единая регистрация (См. Crystal Reports Server)
- Задачи рабочих процессов, 187 , 155
- Задержки рабочего процесса, 203 , 202
- Заключительные события рабочего процесса, 201 , 201
- Записать задачу в журнал (параметр), 203
- Запросы, 23-68
 - AQL по сравнению с SQL, 25
 - CurrentUser, 38
 - DELETE, 58
 - DUPLICATE, 58
 - FROM, 52
 - GROUP BY, 54
 - HAVING, 55
 - INSERT, 56
 - NULL, 37
 - ORDER BY, 56
 - Self, 38
 - sLvl (поле), 40
 - UPDATE, 57
 - WHERE, 53
 - Администратор, 29
 - Внешние ключи, 41
 - Внешние объединения
 - Операторы, 37
 - Число, 35
 - Выражения, 31
 - Вычисляемые поля, 42
 - Записи первичного ключа со значением 0, 35
 - Иерархические таблицы, 40
 - Индексы, 43
 - Индексы - меры предосторожности, 45
 - Константы, 33
 - Необходимые навыки., 24
 - Обзор, 25
 - Объединения, 34
 - Подробные списки, 33
 - Полезность AQL, 24
 - ПолноеИмя (поле), 40
 - Поля, 30
 - Предварительный просмотр, 30
 - Представление, 23
 - Признаки, 42
 - Примеры, 64 , 64
 - alias, 68
 - FullName, 65
 - FullName и sLvl, 65
 - Выражения, 66
 - Двойные условия, 66
 - Иерархические таблицы, 65
 - Незаполненные поля, 66
 - Отсутствие ссылки, 67

- Переменные, 69
- Поля в таблице, связанной с основной таблицей, 65
- Признаки, 66
- Ссылки, 65 , 65
- Числа, даты и текстовые величины, 66
- Редактор, 26
 - Доступ, 27
 - Обзор, 26
- Рекомендации, 33
- Синтаксис, 45
 - UNION, 46
 - Подзапросы, 46
 - Простые запросы, 46
 - Соглашения, 45
- Системные подробные списки, 39 , 39
- Совместное использование, 29
- Создание, 29
- Сортировка, 43
 - Заказ, 44
 - Меры предосторожности, 45
 - Подробные списки, 44
 - Принудительное использование индексов, 43
- Строки описаний, 42
- Упрощенная нотация, 41
- Фильтры, 29
 - Прямой ввод, 30
- Функции (См. Функции AQL)
- Числовые поля, 38
- Число таблиц, 53
- Элементы, 47
 - Выражения, 49
 - Константы, 48
 - Операторы, 50
 - Переменные, 49
 - Поля и ссылки, 47
 - Список выбора, 52
- Запросы (меню), 27
- Запросы экспорта
 - Дизайн, 219
 - Определение, 216
- Запуск (операции рабочих потоков), 186
- Значение по умолчанию
 - Поля (См. Скрипты)
 - Признаки (См. Скрипты)
- Иерархические таблицы - запросы, 65
- Изменение URL-адресов серверов приложений... (мастер), 87
- Импорт часовых поясов, 269
- Индексы
 - (См. также Запросы)
- Инфопанели, 103-106
 - Индикаторы, 103
 - Навигация, 106
 - Обзор, 103
 - Пример, 105
 - Создание, 103
 - Составление отчетов, 103
 - Статистика и отчеты, 103
- Исключения (вкладка), 252
- Исполняемое (тип действия), 133
- Исполняемый
 - Действия
 - Выполнение, 147
- Использовать часовые пояса (параметр), 267 , 258
- История
 - База данных - импорт, 18
 - Удаление, 21
- Календари, 249-255
 - Вкладка «Расписания», 251
 - Влияние на другие функциональности, 250
 - Изменения - учет, 250
 - Контроль, 255
 - Обзор, 249
 - Общая информация, 251
 - Предварительный просмотр, 249
 - Создание, 250 , 250
 - Точность, 252
- Комментарии - скрипты, 242
- Конец (узел мастера), 305
- Константы
 - Запросы, 48
 - Узлы (в мастерах), 289
- Контекст рабочего процесса, 178 , 178
- Конфигурация полей
 - (См. также Скрипты)

- Корневой (узел мастера), 295
- Летнее время, 262
- Летнее время (поле), 259
- Лимиты времени рабочих процессов, 204
- Мастер (тип действия), 135
- Мастера, 281-348
 - { - синтаксис, 288
 - Basic
 - Сцепление, 295
 - Вопросы, 344
 - Выполнение, 339
 - Выполнение - ограничения, 339
 - Длинный (узел), 308
 - Конец (узел), 305
 - Корневой (узел)
 - Определение, 295
 - Подузлы, 300
 - Свойства, 295
 - Синтаксис, 295
 - Начало (узел), 306
 - Начало - страница, 307
 - Нотация - соглашения, 281
 - Обзор, 287
 - Отладка, 339
 - Параметры, 306
 - Переход (узел)
 - Определение, 303
 - Особенности, 304
 - Свойства, 304
 - Синтаксис, 303
 - Последовательности, 293
 - Редактор
 - Доступ, 336
 - Интерфейс, 336
 - Использование, 336
 - Панель инструментов, 337
 - Текст - поиск, 337
 - Свойства, 309
 - Скрипты - ограничения, 294
 - Создание - примеры, 340
 - Страница (узел)
 - Определение, 301
 - Подузлы, 303
 - Свойства, 301
 - Синтаксис, 301

- Страницы - шаблоны, 287
- Строка (узел), 308
- Структурные шаблоны, 285
- Таймер (узел), 307
- Узлы
 - CurrentSelection, 292
 - CurrentTable, 292
 - Декларативный шаблон, 289
 - Имена - ограничения, 283
 - Константы, 289
 - Методы, 291
 - Свойства, 338 , 289 , 288
 - Свойства - ограничения, 288
 - Скрипты, 290
 - Скрипты или фиксированные значения - приоритет, 339
 - Создание, 338
 - Структура и синтаксис, 287
 - Таблицы, 291
 - Числа, 289
- Функции Basic, 294
- Элемент управления (узел), 309 , 308
- Элементы управления, 309
 - CALENDAR, 334
 - CHART, 331
 - CHECKBOX, 313
 - COMBOBOX, 314
 - COMBOEDIT, 335
 - COMMANDBUTTON, 320
 - DATETIMEEDIT, 335
 - DBEDIT, 327
 - DBLISTBOX, 321
 - DBPATH, 329
 - DBQUERYBOX, 324
 - DBTABLE, 328
 - FILEEDIT, 333
 - LABEL, 320
 - LINKEDIT, 329
 - LISTBOX, 315
 - NUMBOX, 334
 - OPTIONBUTTONS, 314
 - PROGRESSBAR, 320
 - TEXTBOX, 330
 - TICKEDIT, 334
 - TIMESPANEDIT, 334

- Ячейки - значения, 321
- Методы - мастера, 291
- Модули
 - Отчеты, 101
- Назначенные операций рабочих процессов
 - Назначение, 182
 - Определение, 155
- Настроить список (меню), 43 , 28
- Начало (узел мастера), 306
- Новости
 - Обзор, 350
 - Определение, 349
 - Отображение, 351
 - Приоритет, 350
 - Создание, 350
 - Сообщение, 350
 - Списки рассылки, 350
 - Чтение, 350
- Обмен сообщениями, 149-151
 - Внутренняя система обмена сообщениями - адреса, 134
 - Обзор, 149
 - Обращение к сообщениям, 150
 - Отправка сообщений, 150
 - Подтверждение получения, 151
- Обмен сообщениями (вкладка), 139
- Обмен сообщениями (тип действия), 133
- Объединения, 34
- Объекты, 283
- Операторы - запросы, 50
- Операции рабочих процессов, 182
 - Графический редактор, 158
 - Определение, 155
- Оповещения рабочих процессов, 203 , 202
- Определения
 - Вычисляемые поля, 271
 - Группы выполнения рабочих процессов, 155
 - Действия, 130
 - Длинный (узел мастера), 308
 - Дочерние объекты, 283
 - Задачи рабочих процессов, 155
 - Заключительные события рабочего процесса, 201
 - Запросы экспорта, 216
- Конец (узел мастера), 305
- Корневой (узел мастера), 295
- Назначенные операций рабочих процессов, 155
- Начало (узел мастера), 306
- Новости, 349
- Объекты, 283
- Операции рабочих процессов, 155
- Переменные, 284
- Переход (узел мастера), 303
- Переходы, 284
- Переходы рабочих процессов, 155
- Полное имя объекта, 283
- Родительские объекты, 283
- Скрипты экспорта, 215
- События рабочих процессов, 155
- Страница (узел мастера), 301
- Строка (узел мастера), 308
- Таймер (узел мастера), 307
- Твипы, 282
- Узлы, 283
- Функции Basic, 230
- Экземпляры рабочих процессов, 154
- Элемент управления (узел мастера), 308
- Элементы управления, 282
- Отчеты, 71-102
 - CurrentUserId, 98
 - URL-адрес, 91
 - Доступность, 90
 - Идентификация, 101
 - Изменение, 97
 - Имена SQL (ограничения), 98
 - Импорт всех отчетов
 - Новая база данных, 93
 - Существующая база данных, 94
 - Кнопки - связывание, 102
 - Конфигурация, 74 , 73
 - Обзор, 71
 - Ограничения, 71
 - Отображение, 88
 - Отчеты со сведениями
 - Использование - пример, 97
 - Конфигурация, 98
 - Создание, 97
 - Печать, 99

- Поочередный импорт, 93
- Создать, 92
- Сохранение в Crystal Reports Server, 95
- Сохранение в базе данных
 - Asset Manager, 93
- Ссылки, 72
- Статистика, 90
- Типы, 72
- Установка, 74 , 73
 - Создать базу данных, 93
 - Существующая база данных, 94
- Хранение, 72
- Ошибки - скрипты, 243
- Переменные
 - Действия, 145
 - Запросы, 49
 - Определение, 284
- Переменные (AQL), 69
- Переход (узел мастера), 303
- Переходы мастеров, 284
- Переходы рабочих процессов, 202
 - Определение, 155
- Редактор, 160
- Печать (тип действия), 135
- Печать отчетов, 99
- Повторная инициализация экземпляра рабочего процесса, если один экземпляр уже выполняется (параметр), 195
- Подзапросы, 46
- Подробные списки, 13-352
 - Закрытые настраиваемые подробные списки, 14
 - Настраиваемые подробные списки, 14
 - Открытые настраиваемые подробные списки, 14
 - Системные подробные списки, 15 , 15
 - настраиваемые подробные списки, 14
- Подтверждение получения, 151 , 140
- ПолноеИмя (поле), 40
 - Запросы, 65
- Полное имя объекта, 283
- Поля
 - (См. также Сохранение истории)
 - Запросы, 65 , 47
 - Незаполненные поля, 66
- Права пользователя - вычисляемые поля, 277
- Представления SQL (См. Экспорт)
- Признаки
 - (См. также Сохранение истории)
- Запросы, 66 , 42
- Проверить местное время по сравнению со временем сервера (параметр), 269 , 266
- Проверить часовой пояс сервера базы данных (параметр), 266
- Программируемые функции, 239
- Продолжительность - скрипты, 239
- Рабочий процесс, 153-210
 - Графический редактор, 158
 - Группы выполнения, 204
 - Действие - инициирование, 183
- Задачи, 187
 - Автоматические действия, 188
 - Администратор, 191
 - Задачи пользователей, 189
 - Назначение, 189
 - Оповещения - ограничения, 188
 - Полный список, 188
 - Поручение, 190
 - Скрипт, 188
 - Создание, 187
 - Тестовые, 188
- Задержки, 204 , 202
- Имена SQL (ограничения), 186
- Инициирование - ограничения, 194
- Контекст, 178
- Лимиты времени, 204
- Обзор, 156
- Операции, 182
 - Автоматическое действие, 184
 - Вкладка «Задержка» - ограничения, 203
 - Вкладка «Оповещения» - ограничения, 204
 - Вопрос, 182
 - Действие пользователя, 183
 - Запуск, 186
 - Инициирование, 187
 - Назначенные - типы операций, 180
 - Тест/скрипт, 185

- Шаблоны, 186
- Оповещения, 202
- Определения, 154
- Отслеживание, 205
- Переходы, 202
- Пример, 160
 - Инициирование, 175
 - Настройка событий, 172
 - Начальное событие, 173
 - Операции, 168
 - Переходы, 174
 - Предпосылки, 162
- Продолжительность, 203
- Реализация, 157
- Роли, 180
- Синхронный рабочий процесс, 200
- Сложные условия ввода, 187
- События, 191
 - System, 192
 - Активация, 195
 - База данных, 192
 - Заключительное событие, 201
 - Начальная операция, 192
 - Обработка, 196
 - Оповещение, 192
 - Периодическое, 195
 - Пользователь, 192
- Таблицы, 212
- Шаблоны операций - ограничения, 187
- Экземпляры
 - Автоматическое удаление, 208
 - Удаление, 207 , 207
- Рабочий процесс - действие - скрипт, 242
- Расписания, 251
- Распределение ПО (тип действия), 135
- Редактор запросов (См. Запросы)
- Редактор рабочего процесса, 158
- Родительские объекты, 283
- Роли в рабочем процессе, 180
- Синтаксис запросов (См. Запросы)
- Системные подробные списки - значения, 40
- Системные подробные списки - скрипты, 240
- Скрипт (тип действия), 134
- Скрипты, 227-247
 - CurrentUser, 241
 - Библиотеки, 237
 - Времена, 239
 - Даты, 239
 - Комментарии, 242
 - Меры предосторожности, 238
 - Нотация - соглашения, 228
 - Первые шаги, 234
 - Поле приложения, 229
 - Примеры, 246 , 243
 - Продолжительность, 239
 - Редактирование, 227
 - Системные подробные списки, 240
 - Советы, 238
 - Сообщения об ошибках, 243
 - Ссылки - идентификатор, 228
 - Узлы (в мастерах), 290
 - Учет регистра, 236
 - Функции Basic
 - Введение, 230
 - Встроенные функции, 230
 - Имена - ограничения, 238
 - Классификация, 234
 - Программируемые функции, 231
 - Справка, 228
 - Типы - ограничения, 233
 - Типы параметров, 233
 - Типы функций, 232
- Скрипты экспорта
 - Выполнение, 224
 - Запись, 218
 - Определение, 215
- События рабочих процессов, 191
 - Определение, 155
 - Редактор, 159
- Сообщения (меню), 150
- Сообщения об ошибках - скрипты, 243
- Сортировка
 - (См. также Запросы)
- Сохранение истории, 17-22
 - n-ссылка, 20
 - n-ссылки, 20 , 19
 - Запись, 18
 - Обзор, 17

- Поля, 19
- Признаки, 20
 - Активация, 21
 - Добавление, 20
 - Изменение, 21
 - Создание, удаление или изменение строк истории, 21
 - Удаление, 21
- Создание строк истории, 17
- Ссылки «1», 19
- Списки
 - Вычисляемые поля, 277
 - Значения подробного списка, 14
- Списки выбора - запросы, 52
- Списки рассылки - новости, 350
- Список значений (См. Подробные списки)
- Ссылки
 - (См. также Сохранение истории)
 - Запросы, 65, 65, 47
 - Отсутствие ссылки, 67
- Ссылочный объект (поле), 139
- Ссылочный объект - рабочий процесс, 178
- Статистика, 107
 - График, 111
 - Навигация, 111
 - Настройка дисплея, 111
 - Оптимизация работы, 110
 - Внедрение способа, 111
 - Проблемы, 110
 - Устранение, 110
 - Отображение, 113
 - Примеры, 114, 113
 - Просмотр статистики, 112
 - На домашних страницах компаний, 113
 - На инфопанелях, 112
- Скрипт, 109
 - Примеры, 118
- Создать, 107
- Сохранение результатов, 110
- Статистика Crystal Reports, 90
- Страница (узел мастера), 301
- Строка (узел мастера), 308
- Строки истории (См. Сохранение истории)
- Строки описаний - запросы, 42
- Схемы рабочих процессов (См. Рабочий процесс)
- Схемы рабочих процессов (меню), 156
- Таблицы
 - (См. также Сохранение истории)
 - Рабочий процесс, 212
- Таймер (узел мастера), 307
- Твипы, 282
- Тест/скрипт (операции рабочих потоков), 185
- Тестирование действий, 145
- Типы действий, 132
- Узлы (в мастерах)
 - Определение, 283
 - Структура и синтаксис, 287
- Фильтры - вычисляемые поля, 278
- Функции AQL, 58
 - Агрегаты, 59
 - Avg, 59
 - Count, 59
 - Countdistinct, 59
 - Max, 59
 - Min, 59
 - Sum, 59
- Даты, 60
 - AddDays, 60
 - AddHours, 61
 - AddMinutes, 61
 - AddSeconds, 61
 - Day, 61
 - DayOfYear, 61
 - DaysDiff, 61
 - DbToLocalDate, 61
 - Getdate, 61
 - Hour, 61
 - HoursDiff, 61
 - LocalToDbDate, 61
 - Minute, 61
 - MinutesDiff, 61
 - Month, 61
 - NumberToTime, 62
 - Second, 62
 - SecondsDiff, 62
 - TextToTime, 62
 - WeekDay, 62

- Year, 62
- Строки, 59
 - Ascii, 59
 - Char, 59
 - CharIndex, 59
 - DateToText, 59
 - EmptyString, 59
 - Left, 60
 - Length, 59
 - LikeParam, 59
 - Lower, 60
 - Ltrim, 60
 - NullBlob, 60
 - NullDate, 61
 - NullMemo, 60
 - NullString, 60
 - NullTime, 61
 - NullTimeStamp, 61
 - NumberToText, 60
 - Right, 60
 - Rtrim, 60
 - Substring, 60
 - TimeStampToText, 60
 - TimeToText, 60
 - Upper, 60
- Тестовые, 64
 - IsNull, 64
- Числовые, 63
 - Abs, 63
 - Ceil, 63
 - DataLength, 63
 - Floor, 63
 - Length, 63
 - Mod, 63
 - NullNumeric, 63
 - NumberToNumber, 63
 - Round, 63
 - Sign, 63
 - TextToNumber, 63
 - Trunc, 63
- Функции Basic
 - (См. также Мастера)
 - (См. также Скрипты)
 - Определение, 230
- Функциональные области - действия, 131
- Часовые пояса
 - Asset Manager Automated Process Manager - ограничения, 266
 - Время на клиентском компьютере, 268
 - Доступность, 257
 - Импорт, 269
 - Летнее время, 265 , 259
 - Обзор, 265
 - Подключения, 267
 - Полезность, 257
 - Последствия для различных операций, 266
 - Создание, 258
 - Управление, 259
 - Экспорт, 269
- Часовые пояса - внедрение, 258
- Числа - запросы, 66
- Шаблоны операций рабочего процесса, 186
- Экземпляры рабочих процессов
 - Ограничения, 179
 - Определение, 154
 - Удаление, 207
- Экспорт, 215-225
 - Запросы, 222
 - Запросы экспорта, 219
 - Извлекаемые данные, 220
 - Методология, 219
 - Права доступа, 217
 - Предварительный просмотр, 222
 - Представления (вкладка) - отображение, 223
 - Представления SQL, 217
 - Создание, 223
 - Удаление, 223
- Процесс, 216
 - Контекстное меню, 217
 - Скрипты, 216
- Рекомендации, 218
- Скрипты экспорта
 - Выполнение, 224
 - Выполнение - DOS, 225
 - Выполнение - процесс, 224
 - Запись, 218
- Сортировка, 220
- Фильтры, 221

Формат вывода скрипта, 222
Форматирование (вкладка) -
отображение, 222
Часовые пояса, 269
Экспортируемые столбцы, 220
Экспорт списка (меню), 217
Элемент управления (узел мастера), 309 ,
308
Элементы управления, 282

, 84 , 79 , 78 , 77

A

Abs (AQL), 63
AddDays (AQL), 60
AddHours (AQL), 61
AddMinutes (AQL), 61
AddSeconds (AQL), 61
Advanced Query Language (См. Запросы)
alias (AQL), 68
am.ini, 267
amexpl.exe, 217
AND (AQL), 50
AQL (См. Запросы)
Ascii (AQL), 59
Asset Manager Automated Process Manager
- часовые пояса, 265
Avg (AQL), 59

B

Basic (См. Скрипты)
(См. также Мастера)
Информация, 228

C

CALENDAR (элемент управления мастера),
334
Ceil (AQL), 63
Char (AQL), 59
CharIndex, 59
CHART (элемент управления мастера), 331
CHECKBOX (элемент управления мастера),
313

COMBOBOX (элемент управления мастера),
314
COMBOEDIT (элемент управления мастера),
335
COMMANDBUTTON (элемент управления
мастера), 320
Count (AQL), 59
Countdistinct (AQL), 59
Crystal Reports (См. Отчеты)
Цель, 71
Crystal Reports (Designer)
Установка, 75
Автоматический режим, 76
Ручная установка, 76
Crystal Reports Server
Единая регистрация
Конфигурация, 84
Объявить сервер, 87
Отчеты
Импорт отчетов, 78
Пользователь Asset Manager, 79
Установка, 75
Автоматический режим, 76
Конфигурация, 77
Ручная установка, 76
Crystal Reports - конфигурация отчетов, 98
CurrentSelection (узлы мастера), 292
CurrentTable (узлы мастера), 292
CurrentUser (AQL), 38
CurrentUser (скрипты), 241
CurrentUserId (отчеты), 98

D

database.txt, 241 , 34 , 15
DataLength (AQL), 63
DATETIMEEDIT (элемент управления
мастера), 335
DateToText, 59
Day (AQL), 61
DayOfYear (AQL), 61
DaysDiff (AQL), 61
DBEDIT (элемент управления мастера), 327
DBLISTBOX (элемент управления мастера),
321

DBPATH (элемент управления мастера), 329
DBQUERYBOX (элемент управления мастера), 324
DBTABLE (элемент управления мастера), 328
DbToLocalDate (AQL), 61
DDE (вкладка), 138
DDE (тип действия), 133
DELETE (AQL), 58
DUPLICATE (AQL), 58

E

EmptyString, 59

F

FILEEDIT (элемент управления мастера), 333
Floor (AQL), 63
FROM (AQL), 52
FullName (AQL), 65

G

g_IZTimeZoneCheckInMns (am.ini), 268
gbbase.wiz, 296
gbbase.xml, 94
Getdate (AQL), 61
GROUP BY (AQL), 54

H

HAVING (AQL), 55
Hour (AQL), 61
HoursDiff (AQL), 61

I

INSERT (AQL), 56
IS NOT NULL (AQL), 51
IsNull (AQL), 64
IS NULL (AQL), 51

L

LABEL (элемент управления мастера), 320
Left (AQL), 60

Length, 59
Length (AQL), 63
LIKE (AQL), 51
LikeParam, 59
LINKEDIT (элемент управления мастера), 329
LISTBOX (элемент управления мастера), 315
LocalToDbDate (AQL), 61
Lower (AQL), 60
Ltrim (AQL), 60

M

MAPI (См. Обмен сообщениями)
Max (AQL), 59
Microsoft SQL Server
 Запросы
 Порядок сортировки, 45
Min (AQL), 59
Minute (AQL), 61
MinutesDiff (AQL), 61
Mod (AQL), 63
Month (AQL), 61

N

NOT LIKE (AQL), 51
NULL (AQL), 37
NullBlob, 60
NullDate, 61
NullMemo, 60
NullNumeric (AQL), 63
NullString, 60
NullTime, 61
NullTimeStamp, 61
NumberToNumber (AQL), 63
NumberToText, 60
NumberToTime (AQL), 62
NUMBOX (элемент управления мастера), 334

O

Old, 242
OPTIONBUTTONS (элемент управления мастера), 314

OR (AQL), 50
Oracle for WorkGroups
 Запросы
 Порядок сортировки , 44
ORDER BY (AQL), 56

P
PROGRESSBAR (элемент управления мастера), 320

R
reports.txt, 93 , 93
Right (AQL), 60
Round (AQL), 63
Rtrim (AQL), 60

S
Second (AQL), 62
SecondsDiff (AQL), 62
SELECT (AQL)
 Ограничения, 52
 Подзапросы - предупреждение, 46
Self (AQL), 38
Sign (AQL), 63
sLvl (поле) - запросы, 65
sLvl - запросы, 40
SMTP (См. Обмен сообщениями)
SQL
 База данных - модификация, 25
 Запросы, 58
SQL по сравнению с AQL, 25
Substring (AQL), 60
Sum (AQL), 59

T
TEXTBOX (элемент управления мастера), 330
TextToNumber (AQL), 63
TextToTime (AQL), 62
TICKEDIT (элемент управления мастера), 334
TIMESPANEDIT (элемент управления мастера), 334
TimeStampToText, 60

TimeToText, 60
Trunc (AQL), 63
tz.scr, 259

U
u2lamlib.dll (файл), 77
UNION (AQL), 46
UPDATE (AQL), 57
Upper (AQL), 60
useSQL92Join (amdb.ini), 35

V
VIM (См. Обмен сообщениями)

W
WeekDay (AQL), 62
WHERE (AQL), 53

Y
Year (AQL), 62