HP Asset Manager

Для операционных систем Windows ® и Linux ®

Версия программного обеспечения: 9.40

Расширенное использование

Дата выпуска документа: Июнь 2013

Дата выпуска программного обеспечения: Июнь 2013

Правовые уведомления

Гарантия

Гарантии на продукты и услуги HP формулируются только в заявлениях о прямой гарантии, сопровождающих эти продукты и услуги. Никакая часть настоящего документа не может быть истолкована как дополнительная гарантия. Компания HP не несет ответственности за содержащиеся здесь технические или редакционные ошибки.

Приводимые в настоящем документе сведения могут быть изменены без предварительного уведомления.

Пояснение об ограниченных правах

Конфиденциальное компьютерное программное обеспечение. Для обладания, использования или копирования необходима действующая лицензия от компании HP. В соответствии с нормами FAR 12.211 и 12.212, коммерческое компьютерное программное обеспечение, документация на компьютерное программное обеспечение и технические данные для коммерческих позиций лицензируются государственным организациям США на условиях стандартной коммерческой лицензии поставщика.

Заявление об авторских правах

© 2002 - 2013 Hewlett-Packard Development Company, L.P.

Заявления о товарных знаках

Adobe® является товарным знаком компании Adobe Systems Incorporated.

Microsoft® и Windows® являются зарегистрированными в США товарными знаками корпорации Microsoft Corporation.

Обновления документации

На титульном листе настоящего документа приведены следующие идентификационные данные.

- Номер версии программного обеспечения для указания версии ПО.
- Дата выпуска документа, которая меняется при каждом обновлении документа.
- Дата выпуска ПО, которая указывает дату выпуска текущей версии программного обеспечения.

Чтобы проверить наличие обновлений или убедиться в том, что используется последняя редакция документа, откройте веб-сайт

http://h20230.www2.hp.com/selfsolve/manuals

Чтобы воспользоваться этим сайтом, необходимо зарегистрировать идентификатор HP Passport и войти в систему. Регистрация HP Passport ID производится на сайте

http://h20229.www2.hp.com/passport-registration.html

или по ссылке **New users - please register** на странице входа в HP Passport.

Оформление подписки в службе поддержки соответствующего продукта также позволит получать обновленные и новые редакции. Обратитесь в торговое представительство компании HP для получения подробной информации.

Поддержка

Используйте веб-сайт технической поддержки программного обеспечения компании HP по адресу

http://www.hp.com/go/hpsoftwaresupport

Этот веб-сайт содержит контактную информацию и дополнительные сведения о продуктах, услугах и поддержке, которые предоставляет HP Software.

Веб-сайт технической поддержки программного обеспечения компании HP предоставляет возможности самостоятельного решения проблем. Это позволяет быстро и эффективно получить доступ к интерактивным средствам технической поддержки, необходимым для управления компанией. Каждый клиент службы поддержки может пользоваться следующими функциями веб-сайта технической поддержки:

- поиск документов базы знаний;
- отправка и отслеживание обращений и запросов на расширение возможностей;
- загрузка исправлений ПО;
- управление договорами на техническую поддержку;
- поиск контактов технической поддержки HP;
- проверка сведений о доступных услугах;
- участие в обсуждениях различных вопросов с другими заказчиками ПО;
- исследование определенных проблем и регистрация для обучения работе с программным обеспечением.

В большинстве случаев для получения поддержки требуется регистрация HP Passport, а также договор на услуги технической поддержки. Чтобы зарегистрироваться для получения идентификатора HP Passport ID, перейдите на веб-сайт

http://h20229.www2.hp.com/passport-registration.html

Дополнительные сведения об уровнях доступа представлены на сайте

http://h20230.www2.hp.com/new_access_levels.jsp

Содержание

Содержание 5
Глава 1: Подробные списки 19
настраиваемые подробные списки 19
Значения настраиваемых подробных списков 19
Открытые настраиваемые подробные списки 20
Закрытые настраиваемые подробные списки 20
Системные подробные списки 20
Глава 2: Сохранение истории 23
Создание записи 24
Изменение поля в таблице или ссылки "1" (например: "Пользователь актива") 24
Добавление n-ссылки к другой таблице (например: "Активы, охватываемые контрактом") 25
Удаление n-ссылки на другую таблицу 25
Изменение n-ссылки на другую таблицу 26
Сохранение истории признаков, связанных с записями 26
Добавление признака 26
Удаление признака 27
Изменение значения признака 27
Создание, удаление или изменение строк истории 27
Инициализация создания строк истории 28
Глава 3: Запросы AQL 29
Инструменты AQL 29
Презентация 29
AQL 29
Запросы в Asset Manager 31
Редактор запросов 31
Обзор 32
Доступ к редактору запросов 33
Создание запроса с помощью редактора запросов 34

Поля, используемые в запросе 36
Запись выражений 36
Константы: Только в клиенте Windows 38
Рекомендации по созданию запросов AQL 38
Представление объединений AQL 39
Причины применения и полезность записей первичного ключа со значением 0 40
Использование значения NULL 42
Self 43
CurrentUser 43
Системные подробные списки 44
Иерархические таблицы 44
Упрощенная нотация AQL 46
Сортировка и индексы: 47
Пример 47
Принудительное использование индексов 47
Порядок сортировки 48
Меры предосторожности 49
Синтаксис AQL 49
Соглашения 50
Синтаксис запросов 50
Элементы запроса 51
оператор FROM 57
Оператор WHERE 58
Оператор GROUP BY 58
Оператор HAVING 59
Оператор ORDER BY 60
Оператор INSERT 60
оператор UPDATE 61
оператор DUPLICATE 61
оператор DELETE 62
Справочная информация по функциям языка AQL 62

Функции AQL группового типа 62
Функции AQL строкового типа 63
Функции AQL типа даты 65
Функции AQL числового типа 68
Функции AQL проверочного типа 69
Примеры запросов: 69
Сравнение поля в основной таблице с каким-либо значением 70
Сравнение ссылки в основной таблице с другой ссылкой 70
Сравнение ссылки в основной таблице с каким-либо значением 70
Сравнение на соответствие полю в таблице, связанной с основной таблицей 70
Иерархические таблицы 70
Запрос, объединяющий два условия 71
Сравнение поля с числами, датами или текстовыми величинами 71
Запрос относительно признаков 71
Поиск записей, соответствующих выражению 71
Поиск незаполненного поля 72
Поиск отсутствия ссылки 72
Запрос с псевдонимом 73
Запрос с переменной 73
Глава 4: SAP Crystal Reports 75
Общий обзор 75
Назначение SAP Crystal Reports (Designer) 75
Место хранения отчетов 76
Место ссылок на отчеты 76
Типы отчетов 76
Установка и настройка клиентов Asset Manager для доступа к отчетам, хранящимся в SAP BusinessObjects Business Intelligence Platform 4.1 77
Предпосылки 77
Установка SAP BusinessObjects Business Intelligence Platform 4.1 и SAP Crystal Reports (Designer) 78
Контрольный список конфигурации для автоматической и ручной установки 81

Конфигурации для доступа веб-клиентов Asset Manager к отчетам, хранящимся в базе данных BusinessObjects Business Intelligence Platform 87

Настройка единого входа (SSO) 95

Объявление URL-адреса доступа к SAP BusinessObjects Business Intelligence Platform 99

Сопоставление имени файла отчета с кодом отчета Crystal 100

Настройка вычисляемого поля sysCoreWebCrystal 100

Миграция с BusinessObjects Enterprise XI 3.1 на BusinessObjects Business Intelligence Platform 4.1 101

Отображение отчета 102

Чтобы отобразить отчет 102

Доступные отчеты 104

Как следует отображать отчет 105

Как приложение Asset Manager создает URL-адреса отчетов 105

Обеспечение доступности отчетов 105

Получение файлов .rpt 106

Хранение отчетов в базе данных Asset Manager 106

Хранение отчетов в SAP BusinessObjects Business Intelligence Platform 109

Изменение отчета Seagate Crystal 110

Отчеты, хранящиеся в базе данных Asset Manager 110

Отчеты, хранящиеся в SAP BusinessObjects Business Intelligence Platform 111

Создание отчета со сведениями 111

Пример использования 111

Конфигурация отчетов в SAP Crystal Reports 112

Печать отчета 113

Отчет со сведениями (только в Windows-клиенте) 113

Графики и отчеты со списками 114

Идентификация отчетов Crystal Reports, свойственных данному модулю 115

Привязка отчета к кнопке на экране (Windows-клиент) или к действиям на экране (веб-клиент) 116

Ограничения 116

Глава 5: Инфопанели 117

Создание инфопанели 117

Пример инфопанели 118

Глава 6: Статистика 121

Создание статистики 121

Использование скрипта вместо запроса 123

Использование скрипта вместо запроса 123

Синтаксис скриптов 124

Оптимизация работы 124

Причины проблем в работе 124

Способ устранения проблем в работе 124

Внедрение этого способа 125

Масштабирование сведений о статистике 125

Определение максимального количества групп для индивидуального представления 125

Просмотр статистики 126

На инфопанелях 126

На домашних страницах компаний 127

Отображение статистики 127

Примеры статистики 127

Статистика, не заданная скриптом 128

Статистика, заданная скриптом 134

Определение настраиваемых цветов для статистики в виде диаграммы 143

Обработка ошибок 143

Практический пример 144

Глава 7: Действия 145

Рекомендации по разработке 145

Использование тестовой функциональной области для настройки действий 145

Пометка веб-служб 146

Определение действия. 146

Функциональная область 147

Создание действия 148

Типы действий 148

Общий процесс создания действия 152

Заполнение вкладки DDE 153
Заполнение вкладки "Обмен сообщениями" 154
Примеры действий 157
Пример действия исполняемого типа 157
Пример действия типа DDE 157
Пример действия типа "Обмен сообщениями" 157
Пример действия типа "Скрипт" 158
Использование переменных 159
Действия, связанные со службой поддержки 160
Действия, определяемые схемами эскалации 160
Действия, определенные в бланках приостановки 161
Тестирование действия 161
Кнопка Вычислить 162
Кнопка Выполнить 162
Выполнение действия 162
Windows-клиент 162
Веб-клиент 163
Множественный выбор в списках 163
Действие типа "Мастер" 163
Действия исполняемого типа 164
Связывание действия с кнопкой на экране 165
Переопределение действия 165
Глава 8: Рабочий процесс 167
Определения 168
Общий обзор 169
Реализация рабочего процесса 170
Использование графического редактора рабочего процесса 171
Операции 171
События 172
Переходы 172
Прочая функциональность 173

Пример рабочего процесса, используемого при утверждении заявки 173
Предназначение 173
Предпосылки 174
Создание операций 177
Настройка событий, создаваемых одновременно с операциями 181
Создание начального события 182
Создание переходов 182
Пример активации экземпляра рабочего процесса 183
Контекст рабочего процесса 185
Определение контекста экземпляра рабочего процесса 186
Объект, на который ссылается экземпляр рабочего процесса 186
Ограничение числа выполняющихся для данного объекта рабочих процессов 186
Роли в рабочем процессе 187
Тип роли в рабочем процессе 188
Определение назначенного для операции 188
Операции рабочих процессов 188
Операция типа "Действие пользователя" 189
Операция типа "Вопрос" 190
Операция типа "Автоматическое действие" 191
Операции типа "Тест/скрипт" 191
Начальная операция 192
Шаблоны операций 192
Инициирование операций 193
Задачи рабочих процессов 193
Создание задач 194
Операция типа "Автоматическое действие" или "Тест/скрипт" 194
Отображение списка всех задач рабочего процесса 194
Выполнение задачи пользователя 195
Назначение задач пользователей 195
Поручение задачи 196
Администрирование задачи рабочего процесса 197

События рабочих процессов 197
Событие системы 198
Событие оповещения 198
Событие пользователя 198
Общие условия активации 201
Обработка событий 202
Применение: реализация синхронной схемы рабочего процесса 208
Заключительное событие 209
Переходы рабочих процессов 209
Оповещения рабочих процессов и лимиты времени 210
Лимит времени 210
Оповещения рабочих процессов 211
Группы выполнения рабочих процессов 212
Приоритет 212
Отслеживание рабочего процесса 212
Отображение экземпляров рабочего процесса в записи 213
Просмотр информации на разных этапах экземпляра 213
Обновление схем и экземпляров рабочего процесса 214
Удаление экземпляров завершенных рабочих процессов 214
Необходимость удаления завершенных рабочих процессов 215
Автоматизация удаления завершенных экземпляров рабочих процессов 215
Техническая информация: модель данных 220
Глава 9: Экспорт данных и создание представлений SQL 221
Определения 221
Скрипты экспорта 221
Запросы экспорта 222
Экспорт данных из базы данных Asset Manager 222
Экспорт данных с помощью скрипта экспорта 223
Экспорт данных с помощью пункта контекстного меню Экспорт списка 223
Управление представлениями SQL в базе данных Asset Manager 224
Рекомендации 224

Определение скрипта экспорта 224
Методология 225
Определение запросов экспорта 225
Формат вывода скрипта экспорта 228
Действия, затрагивающие представления SQL 229
Выполнение скрипта экспорта 229
Выполнение скрипта экспорта в Asset Manager Export Tool 229
Выполнение скрипта экспорта в DOS 230
Глава 10: Скрипты 233
Определение скрипта 233
Обзор 233
Информация об используемой версии языка Basic 234
Нотация доступа к данным 234
Применение скриптов 235
Введение в функции 236
Определение функции 236
Встроенные функции и программируемые функции 236
Типы функций и параметров 238
Классификация функций Basic 239
Первые шаги в написании скриптов 239
Пример сценария 240
Этап 1: создание признака "Руководство" 240
Этап 2: открытие окна редактирования 240
Этап 3: анализ и определение алгоритма 241
Этап 4: написание скрипта Basic 241
Этап 5: проверка скрипта Basic 241
Библиотеки скриптов 242
Концепции 242
Создание библиотеки скриптов 243
Вызов скрипта из библиотеки скриптов 243
Советы и предупреждения 244

Меры предосторожности при использовании программируемых функций 244
Формат констант "Дата+время" в скриптах 245
Формат констант "Продолжительность" в скриптах 245
Доступ для считывания и для записи к системному подробному списку 246
Виртуальная ссылка CurrentUser 247
Виртуальная ссылка "Старая" 248
Комментарии в скрипте на языке Basic 248
Инициирование сообщения об ошибке 248
Использование рекурсивных функций 249
Первый пример 249
Описание проблемы 249
Этап 1: анализ и определение алгоритма 250
Этап 2: написание скрипта Basic 250
Этап 3: проверка скрипта Basic 251
Второй пример 251
Описание проблемы 251
Этап 1: анализ и определение алгоритма 251
Этап 2: написание скрипта Basic 252
Этап 3: проверка скрипта Basic 252
Глава 11: Календари 253
Обзор календарей 253
Сведения о календаре 253
Использование календарей 253
Влияние календарей на некоторые области функциональности 254
Методология, используемая для создания календаря 255
Описание создания календаря 255
Ввод общей информации 255
Заполнение вкладки "Расписание" 256
Заполнение вкладки "Исключения" 257
Предварительный просмотр календаря 259
Глава 12: Часовые пояса 261

Управление часовыми поясами 261
Внедрение часовых поясов 262
Создание часовых поясов 262
Управление часовыми поясами 263
Формат поля "Летнее время" 263
Значения аргумента <Year> 264
Значения аргумента <DaylightInfo> 264
Пример 267
Управление часовыми поясами в Asset Manager Automated Process Manager 269
Тесты, которые необходимо выполнить 269
Частота тестирования 270
Последствия для различных операций 270
Создание базы данных 270
Подключение к базе данных 271
Импорт и экспорт 272
Календари и схемы эскалации 272
Глава 13: Вычисляемые поля 275
Определение вычисляемого поля 275
Полезность вычисляемых полей 275
Создание вычисляемых полей 276
Введение 276
Методы создания 278
Использование вычисляемых полей 281
Использование вычисляемых полей в конфигурациях списков 281
Фильтрация записей таблицы 281
Ссылки на вычисляемые поля 282
Глава 14: Мастера 283
Обзор 283
Соглашения о нотации 284
Определения 284
Структурный шаблон 286

Шаблон страницы мастера 287
Общее число пунктов 288
Общая структура и синтаксис 288
Соединение мастеров 289
Выполнение 289
Параметры 289
Функции Basic 290
Значения, возвращаемые функциями 290
Сцепление строк в скриптах на языке Basic 291
Свойства узла 291
Декларативный шаблон 291
Определение константы как значения свойства 292
Ссылка на свойство 292
Определение скрипта как значения свойства 292
Методы, применимые в свойствах 293
Свойство типа "Таблица" 294
Использование глобальных переменных CurrentTable и CurrentSelection 295
Типы узлов 295
Корневой узел 296
Узел "Страница" 304
Узел перехода 306
Свойства узла перехода 307
Конечный узел 308
Начальный узел 310
Узел таймера 310
Узлы "Длинный" и "Строка" 311
Узел элемента управления 312
Типы элементов управления и связанные свойства 312
Общие свойства 313
Элемент управления CHECKBOX 320
Элемент управления COMBOBOX 320

Элемент управления OPTIONBUTTONS 321
Элемент управления LISTBOX 322
Элемент управления LABEL 329
Элемент управления PROGRESSBAR 330
Элемент управления COMMANDBUTTON 330
Элемент управления DBLISTBOX 331
Элемент управления DBQUERYBOX 335
Элемент управления DBEDIT 339
Элемент управления DBTABLE 341
Элемент управления DBPATH 341
Элемент управления LINKEDIT 341
Элемент управления TEXTBOX 343
Элемент управления CHART 343
Элемент управления FILEEDIT 346
Элемент управления TICKEDIT 347
Элемент управления CALENDAR 347
Элемент управления TIMESPANEDIT 347
Элемент управления NUMBOX 347
Элемент управления COMBOEDIT 348
Элемент управления DATETIMEEDIT 348
Использование графического редактора 349
Обзор интерфейса 350
Создание нового узла 351
Редактирование свойств узла 351
Компиляция, выполнение и отладка мастера 352
Пример создания мастера 352
Пример создания мастера 353
Шаг 1 - Анализ ваших потребностей 353
Шаг 2 - Определение того, как будет организован мастер 355
Пример программирования мастера 358
Часто задаваемые вопросы 364

Не работает {lbxMyListBox.Values.Count} 365
Не работает {lbxMyListBox.Line(lRow)} 365
Не работает {lbxMyListBox.Values.Line({lbxTmp})} 365
Не работает назначение фиксированного значения свойству 365
Выполнение мастера, создающего актив в базе данных, приводит к ошибке 366
При выполнении мастера отображается неполное сообщение об ошибке 366
Если на странице мастера используется управляющий элемент DBLISTBOX, наблюдается снижение производительности 366
Как разрешить или запретить изменение некоторых столбцов в элементе управления LISTBOX? 367
Как сделать так, чтобы мастером открывался подробный экран? 367
В чем разница между свойствами COLNAME и COLTITLE элемента управления LISTBOX? 367
Слово "query" нельзя использовать в элементах мастера 368
Глава 15: Новости 369
Определение позиции новостей 369
Обзор новостей 369
Создание позиции новостей 369
Чтение новостей 370
Важность позиций новостей 370
Сообщение для рассылки 370
Список рассылки новостей 370
Параметр Все группы сотрудников (SQL: bAllGroups) 370
Параметр Включить подгруппы (SLQ bChildGrps) 371
Отображение новостей 371
Активизация бегущей строки новостей 371
Благодарим за отзыв! 373

Глава 1: Подробные списки

Подробный список представляет собой список значений, предлагаемых приложением Asset Manager для заполнения некоторых полей (стандартных полей на экране сведений или значение признака), например, заголовок, положение, страна и бренд. Они хранятся в таблице **Подробные списки** (seItemizedList) 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** (seItemizedList)

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** (seItemizedList).

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

Глава включает следующее:

настраиваемые подробные списки 19

Системные подробные списки 20

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

Поле **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

Поле **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:

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;
- с помощью Asset Manager Application Designer;
- использование файла **database.txt** для описания структуры базы данных. Этот файл находится в каталоге **<папка установки Asset Manager>\ doc\ infos**.

Глава 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, приложение 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. Щелкните правой кнопкой мыши поле или ссылку, изменение которой необходимо отслеживать.
2. В контекстном меню выберите пункт **Настройка объекта**.
3. Перейдите на вкладку **Общее** экрана конфигурации.
4. Установите для поля **Сохранять историю** значение **Да**.
5. Для подтверждения нажмите **OK**.
   Все изменения, касающиеся истории, сохраняются в базе данных сразу после нажатия кнопки **OK** в окне настройки базы данных.

Примечание. Можно также определить, сохраняется ли история для поля или ссылки, с помощью Asset Manager Application Designer. Однако включить отслеживание изменений истории в веб-клиенте нельзя.

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:

- **Изменено** (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 отобразить его, щелкните правой кнопкой мыши список, а затем выберите **Сервисные программы/ Настройка списка**.

> **Примечание.** 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 полей типа "комментарий". Поля с типом "комментарий" обрабатываются не так, как другие поля, так как они по-другому хранятся в базе данных (макс. размер: 32767 символов).

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

Глава включает следующее:

Создание записи 24
Изменение поля в таблице или ссылки "1" (например: "Пользователь актива") 24
Добавление n-ссылки к другой таблице (например: "Активы, охватываемые контрактом") 25
Удаление n-ссылки на другую таблицу 25
Изменение n-ссылки на другую таблицу 26
Сохранение истории признаков, связанных с записями 26

Создание записи

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

Записи Asset Manager:

- **Изменено**: 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:

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

Примечание. 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), заполненное с помощью системного подробного списка. Одна из записей этого подробного списка отображается как **Уведомлено** и хранится в виде значения **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:

- **Изменено**: 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:

- **Изменено**: 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

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

Сохранение истории признаков, связанных с записями

Примечание. В настоящее время данная функциональность в веб-клиенте недоступна.

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 26
Удаление признака 27
Изменение значения признака 27
Создание, удаление или изменение строк истории 27
Инициализация создания строк истории 28

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:

- **Изменено** (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:

- **Изменено**: 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:

- **Изменено**: 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 Asset Manager Automated Process Manager.

Создание, удаление или изменение строк истории

Нельзя сохранять историю создания строк истории.

Инициализация создания 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.

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

Когда для данного поля установлено значение **Да**, приложение Asset Manager автоматически создает строки истории для данного признака. Строки истории можно просматривать на вкладке **История** таблицы, связанной с данным признаком.

Внимание! При удалении записи также удаляются все строки истории — либо во время удаления или с помощью Asset Manager Server. Нельзя сохранять историю создания истории.

Глава 3: Запросы AQL

Глава включает следующее:

Инструменты AQL 29
Презентация 29
Редактор запросов 31
Рекомендации по созданию запросов AQL 38
Сортировка и индексы: 47
Синтаксис AQL 49
Справочная информация по функциям языка AQL 62
Примеры запросов: 69

Инструменты AQL

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

Презентация

В данном разделе описывается язык AQL и перечисляется, где могут использоваться запросы:

AQL 29
Запросы в Asset Manager 31

AQL

AQL ("Advanced Query Language") — язык запросов, который используется 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.

Примечание. 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 предпочтительнее для запросов к базе данных Asset Manager, чем язык SQL, по следующим причинам.

- Независимость от СУБД

  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** в SQL Oracle for WorkGroups и функции **Substring** в 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_lModelId**) и **Полное имя** (**FullName**) на языке AQL было бы написано следующее:

  ```
  SELECT FIRST_ROWS lModelId, 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_lModelId) */ M1.lModelId, M1.FullName
  FROM amModel M1
  ```

  Для Microsoft SQL Server будет использоваться код:

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

  Код IBM DB2:

  ```
  SELECT lModelId, 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 SAP Crystal Reports;
- для создания мастеров;
- если используется Asset Manager API;
- 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.

Приложение 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.

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

Приложение 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.

Примечание. Редактор запросов не может использоваться в мастерах.

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

В этом разделе содержатся следующие сведения.

Обзор 32
Доступ к редактору запросов 33
Создание запроса с помощью редактора запросов 34
Поля, используемые в запросе 36
Запись выражений 36
Константы: Только в клиенте Windows 38

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 используются совместно). Однако нельзя писать запросы непосредственно на языке SQL.

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

- Windows-клиент

  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:

    Примечание. 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, используемых при запросах: ограничения запросов, фильтры запросов, конфигурация списка и так далее;
  - С помощью внешних программ: Asset Manager Export Tool и т.д.

  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 <table> [WHERE <clause>] [ORDER BY <clause>]
  ```

  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**.

> **Примечание.** 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 Asset Manager позволяет определять условия, которыми объединяются поля, выражения вычисления, константы и операторы.

Можно определить одно или несколько условий фильтров.

- Windows-клиент

  Определение условия фильтра в клиенте Windows.

  a. В начальной таблице выберите поле, константу или выражение (**Поле 1**), которые необходимо сравнить с полем, константами или выражением (**Поле 2**).

  b. Подтвердите условие фильтра, переместив его в нижнюю часть экрана с помощью кнопки .

  c. 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**.

  d. Подтвердите запрос, нажав кнопку **Изменить** в сведениях о запросе.

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

  Примечание. 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

  Примечание. 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 в клиенте 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 : В Asset Manager откроется предварительный просмотр результатов запроса в виде списка записей. Справа внизу окна показывается число записей, удовлетворяющих запросу.

Примечание. 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:

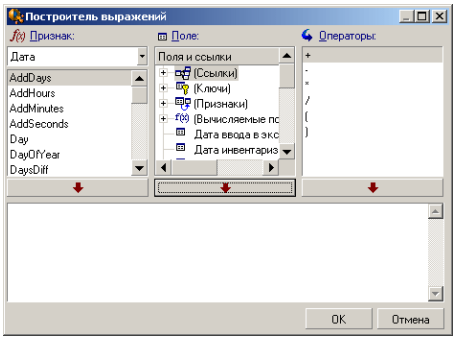
- ввести его непосредственно в соответствующее поле;
- или использовать построитель выражений Asset Manager.

Windows-клиент

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**:

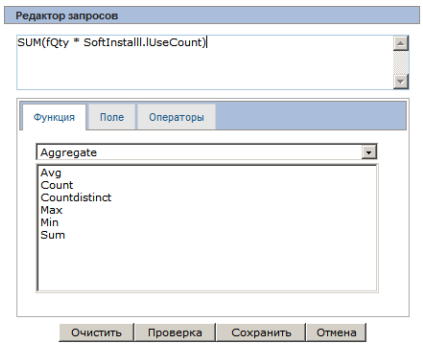
1. Выберите функцию, поле или оператор.
2. Щелкните .

После того, как выражение определено, щелкните **OK**, 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. Построитель 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 в клиенте Windows

Константы 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 значок .
2. Отобразится окно выбора со значениями, присутствующими в базе данных для поля, указанного в качестве условия поиска.

Примечание. 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 и Справочная информация по функциям языка AQL.

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: **[папка установки Asset Manager]/doc/infos**. Чтобы иметь доступ к данному файлу, необходимо установить документацию.

В этом разделе содержатся следующие сведения.

Представление объединений AQL 39
Причины применения и полезность записей первичного ключа со значением 0 40
Использование значения NULL 42
Self 43
CurrentUser 43
Системные подробные списки 44
Иерархические таблицы 44
Упрощенная нотация AQL 46

Представление объединений AQL

Определение

Объединение суммирует несколько таблиц данный 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** (**lModelId**);
- ее **Полное имя** (**FullName**);
- **Имя** таблицы, связанной с брендами (**amBrand**).

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

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

```
SELECT M1.lModelId, M1.FullName, B2.Name FROM amModel M1, amBrand B2 WHERE M1.lB
randId=B2.lBrandId
```

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: См. руководство **Asset Manager — установка и обновление**, 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.AssetTag, l.name FROM amPortfolio a, amEmplDept e, amLocation l WHERE a
.lUserId *= e.lEmplDeptId AND e.lLocaId *= l.lLocaId
```

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. Приложение Asset Manager, таким образом, создает код SQL без внешних объединений:

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

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.

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

Использование значения NULL

Приложение 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> = []

Примечание. Примечание. Если числовое поле не заполнено, его значение равно "0". Аналогичным образом, отсутствие ссылки описывается как "ссылка = 0" или "внешний ключ = 0". Пример: "Location=0" или "lLocaId=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

```
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.

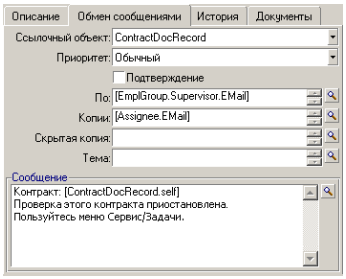
```
SELECT lPortfolioItemId FROM amPortfolio WHERE User = CurrentUser
```

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**.

Примечание. 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: **[папка установки Asset Manager]/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/'
  ```

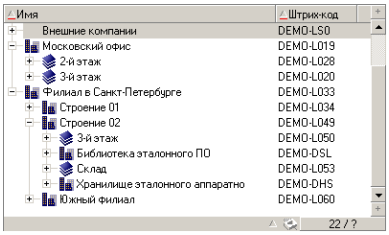
- В таблице местоположений в поле "ПолноеИмя" хранится имя местоположения, предваряемое именами родительских местоположений.

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

"sLvl" fields

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-DSL
Склад	DEMO-L053
Хранилище эталонного аппаратно	DEMO-DHS
Южный филиал	DEMO-L060

22 / ?

Следующим запросом выбираются запись Sales Head Office и ее подкомпоненты:

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

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

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

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

```
SELECT Self FROM amEmplDept WHERE (FullName LIKE '/Sales Head Office/Sales/%') A
ND (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 lLocaId = 0
```

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

Строки описаний

В предложениях SELECT и ORDER By имя SQL ссылки, после которой не следует точка, эквивалентно объединению <имя SQL ссылки >.self, которое 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
```

Функции

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

В этом разделе содержатся следующие сведения.

Пример 47
Принудительное использование индексов 47
Порядок сортировки 48
Меры предосторожности 49

Пример

В случае следующего запроса:

```
SELECT lModelId, 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 другом случае: Ядро базы данных использует индекс "Бренд" и отображает результаты тогда и в том виде, в котором они найдены. Таким образом, первые элементы данных отображаются быстро, но общее время обработки будет большим.

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
```

> **Примечание.** 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
Список без принудительного использования индексов	NULL NULL A a B b C D

В Microsoft SQL Server

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

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.

В этом разделе содержатся следующие сведения.

Соглашения 50
Синтаксис запросов 50
Элементы запроса 51
оператор FROM 57
Оператор WHERE 58
Оператор GROUP BY 58
Оператор HAVING 59
Оператор ORDER BY 60
Оператор INSERT 60
оператор UPDATE 61
оператор DUPLICATE 61
оператор DELETE 62

Соглашения

Ниже приведены соглашения, используемые для описания синтаксиса AQL.

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.
`FROM`	Элементами из букв в верхнем регистре указываются литеральные выражения.

Синтаксис запросов

Простые запросы

`SELECT` [DISTINCT] [FIRST_ROWS] <Список выбора>

[оператор FROM]

[оператор WHERE]

[оператор GROUP BY]

[оператор HAVING]

[оператор ORDER BY]

Подзапросы

Языком AQL поддерживается использование подзапросов вместо полей.

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

(`SELECT` [DISTINCT] <выражение>

[оператор FROM]

[оператор WHERE]

[оператор GROUP BY]

[оператор HAVING]

)

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]

[оператор WHERE]

[оператор GROUP BY]

[оператор HAVING]

[`UNION` | `UNION ALL` | `INTERSECTS` | `MINUS`]

`SELECT` <список выбора>

[оператор FROM]

[оператор WHERE]

[оператор WHERE]

[оператор GROUP BY]

[оператор HAVING]...]

[оператор ORDER BY]

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.]]<поле>

  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):
    <table.[link...]field>
    <alias.[link...]field>

  - Или не объявлять таблицу в операторе `FROM` и вместо этого использовать двоеточие ":":
    <table.[link...]field>
    <table[_alias]:[link[_alias]...]field>

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

  Например, при написании запроса в Asset Manager имеется доступ только к оператору `WHERE`. Начальная таблица запроса является неявной (таблица, к которой применяется фильтр, поле **Таблица** (**TableName**) в сведениях о запросе и т.д.). 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'

  Примечание. Используйте escape-последовательность **''** для представления одинарных кавычек в текстовой константе. Например:

```
'a city with a '' in its name'
```

- 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;
  - Используется 24-часовой формат времени (а не 12-часовой формат с временем до и после полудня (am/pm)).
  - 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:

  #гггг-мм-дд чч:мм:сс#

  #гггг-мм-дд#

  #чч:мм:сс#

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

Переменные

Вместо указания константы (см. Константы) можно ввести переменную, значение которой заполняется пользователем при выборе фильтра, использующего этот запрос.

Для определения переменных может использоваться разный синтаксис:

- $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.

  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
- функциональные возможности
- 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:

AQL - логические операторы

Оператор	Значение
AND	Логическое AND
OR	Логическое OR

Чтобы оптимизировать запрос, иногда разумнее исключить логические операторы, если возможно использование оператора сравнения. В следующем примере показано, как оптимизировать фильтр запроса, используемый для выбора позиций портфеля, в поле **Назначение** которого (имя SQL: seAssignment) установлено значение **Ожидается поставка** или **Возврат для обслуживания**. Данные два элемента системного подробного списка имеют значения "3" и "4" соответственно. Таким образом, возможна следующая запись:

```
(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.

AQL - операторы сравнения

Оператор	Значение
=	Равно
<> =!	Отлично от
>	Больше, чем
<	Меньше, чем
>=	Больше или равно
=<	Меньше или равно
=*	Правое внешнее объединение. Использование данного оператора ограничено из-за особенностей обработки ссылок с помощью AQL
*=	Левое внешнее объединение. Использование данного оператора ограничено из-за особенностей обработки ссылок с помощью AQL.

AQL - операторы сравнения, продолжение

Оператор	Значение
LIKE NOT LIKE	Выполняется аналогично оператору "=" и позволяет также использовать подстановочные знаки. 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 IS NOT NULL	Проверяется, значение поля NULL или нет. Приложение 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:

  - С помощью следующего запроса Asset Manager Export выводится список позиций портфеля, бренд которых используется в месте размещения Milwaukee. (Данный запрос не включает суб-местоположений):

```
SELECTDISTINCT lModelId, Model.Brand FROM amPortfolio WHERE Model.Brand = ANY
(SELECT Model.lBrandID FROM amPortfolio WHERE Location.FullName LIKE '/Milwauk
ee 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 экспортируемым столбцам.

> **Примечание.** 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> (<подзапро
с>)
```

Оператор GROUP BY

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

Синтаксис
`GROUP BY` <выражение бег агрегатов> [, <выражение бег 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` — примеры

Приведенным ниже запросом возвращается полное число брендов, присутствующих в базе данных. Для каждого актива со связанным брендом приложение 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(lAstId) 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(lAstId) FROM amAsset GROUP BY Model.Brand.Name HA
VING Model.Brand.Name > 'B'
```

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

```
SELECT Model.Brand.Name, count(lAstId) 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(lAstId) FROM amAsset GROUP BY Model.Brand.Name HA
VING 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]...) | **AQL sub-query**)

Данный оператор включен в Asset Manager API — **AmDbExecAql**.

Дополнительные сведения о Asset Manager API 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")            lErr = AmSetFieldLongValue(hr
Alarm, "bSecondLevel", 0)
            lErr = AmSetFieldLongValue(hrAlarm, "dtTrig1", AmGetFieldLongValue(h
rAsset, 2)-lDaysBefore*86400)
```

```
lErr = AmSetFieldLongValue(hrAlarm, "lAction1Id", lActionId)
lErr = AmSetFieldLongValue(hrAlarm, "lMonitObjId", lAstId)
lErr = AmSetFieldStrValue(hrAlarm, "MonitoredField", "dWarrEnd")
lErr = AmSetFieldStrValue(hrAlarm, "MonitoredTable", "amAsset")
lErr = AmSetFieldLongValue(hrAlarm, "sDaysBefore1", lDaysBefore)
lErr = AmInsertRecord(hrAlarm)
```

Программный код мастера с использованием оператора `INSERT`

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

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] [оператор WHERE]

Пример

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

Код действия без использования оператора `UPDATE`

```
hr = AmGetRecordFromMainId("amPOrder", [lPOrdId])
lErr = AmSetFieldLongValue(hr, "seStatus", "$(IDS_POSTATUS_ORDERED)")
lErr = AmUpdateRecord(hr)
```

Код действия с использованием оператора `UPDATE`

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

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] [оператор WHERE]

оператор DELETE

Этот оператор позволяет удалять поля записей в таблице базы данных.

Синтаксис
`DELETE` [оператор FROM] [оператор WHERE]

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

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

В этом разделе содержатся следующие сведения.

Функции AQL группового типа 62
Функции AQL строкового типа 63
Функции AQL типа даты 65
Функции AQL числового типа 68
Функции AQL проверочного типа 69

Функции AQL группового типа

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.

AQL - функции группового типа, продолжение

Функция	Описание
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

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 <строке1> находится в позиции 1. Если <строка1> не встречается в <строке2>, функция возвращает 0.
DateToText(<дата>)	<Дата> преобразуется 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)`

AQL - функции строкового типа, продолжение

Функция	Описание
Left(<строка>, <n>)	Возвращается 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. **Примечание.** Программно он представляет собой значение NULL, приведенное к типу объекта Blob.
NullMemo( )	Создается объект "Заметка" со значением NULL. **Примечание.** Программно он представляет собой значение NULL, приведенное к типу объекта "Заметка".
NullString( )	Создается объект "Строка" со значением NULL. **Примечание.** Программно она представляет собой значение NULL, приведенное к типу объекта "Строка".
NumberToText(<число>)	<Число> преобразуется 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 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

AQL - функции типа даты

Функция	Описание
AddDays(<дата>, <число>)	Добавляется указанное число дней к полю типа "Дата" или "Дата и время".
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) **Примечание.** Функция HoursDiff возвращает различные результаты в зависимости от используемой базы данных, так как разные базы данных по-разному округляют минуты. Пример: 59 минут в DB2 = 0 часов, в то время как 59 минут в MSSQL = 1 час.

AQL - функции типа даты, продолжение

Функция	Описание
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. **Примечание.** Программно он представляет собой значение NULL, приведенное к типу объекта "Дата".
NullTime( )	Создается объект "Время" со значением NULL. **Примечание.** Программно он представляет собой значение NULL, приведенное к типу объекта "Время".
NullTimeStamp( )	Создается объект "Дата и время" со значением NULL. **Примечание.** Программно он представляет собой значение NULL, приведенное к типу объекта "Дата и время".
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>)	Преобразование текста в дату. Дополнительные (и необязательные) параметры языка и формата обеспечивают доступ к форматам базы данных и дают возможность указать ожидаемый формат для преобразования, а также настройки поддержки языков. Дополнительные сведения см. в документации по базе данных.

AQL - функции типа даты, продолжение

Функция	Описание
WeekDay(<дата>)	Возвращается число дней в неделе для поля типа "Дата" или "Дата и время". Данное число зависит от настроек сервера. Например, в Microsoft SQL по умолчанию используется настройка (1=воскресенье, 2=понедельник, ..., 7=воскресенье). Настройка по умолчанию в Oracle – (1=понедельник, ..., 7=воскресенье).
Year(<дата>)	Возвращается число дней в неделе для поля типа "Дата" или "Дата и время" (например: 2000).

Примеры использования функции типа даты

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

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

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

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

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

Функции AQL числового типа

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>, <b>)	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. **Примечание.** 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> имеет значение null (=0).
TextToNumber (<текст>)	<Текст> преобразуется 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

AQL - функции проверочного типа

Функция	Описание
IsNull(<a>, <b>)	Если "a" имеет значение "Null", "a" замещается на "b". Типы "a" 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.

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

В данном разделе содержатся следующие примеры:

Сравнение поля в основной таблице с каким-либо значением 70

Сравнение ссылки в основной таблице с другой ссылкой 70

Сравнение ссылки в основной таблице с каким-либо значением 70

Сравнение на соответствие полю в таблице, связанной с основной таблицей 70

Иерархические таблицы 70

Запрос, объединяющий два условия 71
Сравнение поля с числами, датами или текстовыми величинами 71
Запрос относительно признаков 71
Поиск записей, соответствующих выражению 71
Поиск незаполненного поля 72
Поиск отсутствия ссылки 72
Запрос с псевдонимом 73
Запрос с переменной 73

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.Name
```

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/%') AND (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 = 'M
adison site')
```

Сравнение поля с числами, датами или текстовыми величинами

Например: все наряды 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#) A
ND (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 самой большой покупной цене, содержащейся в базе данных. Обратите внимание, что в основном запросе используется подзапрос, чтобы идентифицировать максимальную цену.

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

Поиск незаполненного поля

Пример: Все сотрудники без номера телефона. Обратите внимание, что пустая строка представляется двумя одинарными кавычками ('').

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

Поиск отсутствия ссылки

Случай с одной ссылкой

Пример: Все позиции портфеля, не назначенные пользователям. Обратите внимание, что отсутствие ссылки обозначается с помощью "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.lAstId FROM amAsset A1 WH
ERE A1.lModelId = amModel.lModelId))
```

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.lModelId FROM amModel A1
WHERE A1.lParentId = amModel:lModelId))) 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.lModelId) FROM amModel a WHERE a.lParentId = lModelId)"), чтобы выбрать модели без подмоделей. При проверке для нескольких ссылок для каждой модели выбирается ее идентификатор lModelId и подсчитываются все модели, идентификатор lParentId которых равен lModelId.

  Примечание. 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 lModelId FROM amModel WHER
E (FullName LIKE (p.FullName + '%/')) AND (Nature.Name = 'Computer')))
```

Примечание. При попытке использования этого запроса с Asset Manager Export Tool, отобразится 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 (Train
ings_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 (Trainin
gs.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 (Training
s.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.semf FROM amPortfolio portfolio, amModel FullN
ame
```

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 содержа
ться 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 '\'
```

Глава 4: SAP Crystal Reports

В данной главе объясняется, как отображать и печатать отчеты в Asset Manager с помощью SAP Crystal Reports.

Примечание. Работа с SAP Crystal Reports недоступна в Linux -версии приложения Asset Manager.

Глава включает следующее:

Общий обзор 75

Установка и настройка клиентов Asset Manager для доступа к отчетам, хранящимся в SAP BusinessObjects Business Intelligence Platform 4.1 77

Миграция с BusinessObjects Enterprise XI 3.1 на BusinessObjects Business Intelligence Platform 4.1 101

Отображение отчета 102

Обеспечение доступности отчетов 105

Изменение отчета Seagate Crystal 110

Создание отчета со сведениями 111

Печать отчета 113

Идентификация отчетов Crystal Reports, свойственных данному модулю 115

Привязка отчета к кнопке на экране (Windows-клиент) или к действиям на экране (веб-клиент) 116

Ограничения 116

Общий обзор

В этом разделе содержатся следующие сведения.

Назначение SAP Crystal Reports (Designer) 75

Место хранения отчетов 76

Место ссылок на отчеты 76

Типы отчетов 76

Назначение SAP Crystal Reports (Designer)

SAP Crystal Reports (Designer) используется для создания и изменения отчетов (файлы **.rpt**).

Отчеты, созданные с помощью SAP Crystal Reports (Designer), можно просматривать в клиенте Windows и веб-клиенте Asset Manager, однако создавать или изменять отчеты в приложении Asset Manager нельзя.

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 **SAP BusinessObjects Business Intelligence Platform** .
  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 отправляются в SAP BusinessObjects Business Intelligence Platform.
  SAP BusinessObjects Business Intelligence Platform извлекает данные из базы данных Asset Manager и отображает отчет в клиенте Windows или браузере Internet Explorer (веб-клиент).

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 хранится в **SAP BusinessObjects Business Intelligence Platform**: 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 клиентов **Asset Manager** для доступа к отчетам, хранящимся в **SAP BusinessObjects Business Intelligence Platform 4.1**

В этом разделе содержатся следующие сведения.

Предпосылки 77

Установка SAP BusinessObjects Business Intelligence Platform 4.1 и SAP Crystal Reports (Designer) 78

Контрольный список конфигурации для автоматической и ручной установки 81

Конфигурации для доступа веб-клиентов Asset Manager к отчетам, хранящимся в базе данных BusinessObjects Business Intelligence Platform 87

Настройка единого входа (SSO) 95

Объявление URL-адреса доступа к SAP BusinessObjects Business Intelligence Platform 99

Сопоставление имени файла отчета с кодом отчета Crystal 100

Настройка вычисляемого поля sysCoreWebCrystal 100

Предпосылки

Создание подключения к базе данных Asset Manager

Чтобы включить доступ SAP BusinessObjects Business Intelligence Platform к базе данных Asset Manager, необходимо создать подключение ODBC с драйвером Asset Manager ODBC.

Если Asset Manager и BusinessObjects Business Intelligence Platform развернуты на разных компьютерах, необходимо установить следующие два компонента на компьютере, на котором выполняется BusinessObjects Business Intelligence Platform.

- Windows-клиент Asset Manager
- Драйвер Asset Manager ODBC

  См. руководства **Установка и обновление** Asset Manager и **Администрирование** Asset Manager, глава**Доступ к базе данных через ODBC**.

Кроме того, выполните следующие шаги для создания подключения ODBC.

1. Запустите Windows-клиент приложения Asset Manager.
2. Создайте подключение к базе данных Asset Manager (меню **Файл/ Управление**

соединениями...).

См. руководство **Администрирование** Asset Manager, глава **Создание, изменение и удаление базы данных базы данных Asset Manager**, раздел **Создание подключения с помощью Asset Manager**.
Необходимо запомнить имя данного подключения. Оно понадобится позже.

3. Протестируйте подключение (кнопка **Тест** в окне управления подключениями).

Установка SAP BusinessObjects Business Intelligence Platform 4.1 и SAP Crystal Reports (Designer)

SAP BusinessObjects Business Intelligence Platform и SAP Crystal Reports (Designer) можно установить несколькими способами.

- С помощью **OEM-версии**, доступной в файлах **AssetManager_CRS_9.40.zip** и **AssetManager_CRD_9.40.zip**.

  Установку можно выполнить вручную или автоматически.

- При помощи **полной коммерческой версии**, которая уже имеется.

  См. документацию SAP Crystal Reports.

  Поддерживаемые версии SAP Crystal Reports см. в таблице поддержки Asset Manager.

Примечание. Настройки будут устанавливаться для каждого из отчетов.

Примечание. С Asset Manager можно использовать только приложение SAP Crystal Reports, установленное в среде Windows. Это связано с тем, что в SAP Crystal Reports используется драйвер Asset Manager ODBC.

В этом разделе описывается установка OEM-версии SAP Crystal Reports (Designer) для Asset Manager и SAP BusinessObjects Business Intelligence Platform для Asset Manager из предоставленных файлов **AssetManager_CRD_9.40.zip** и **AssetManager_CRS_9.40.zip**.

Установка SAP Crystal Reports (Designer) и SAP BusinessObjects Business Intelligence Platform для Asset Manager (автоматический режим)

Установка в автоматическом режиме выполняется с использованием предварительно определенных опций.

Чтобы установить SAP BusinessObjects Business Intelligence Platform, выполните следующие действия.

1. Запустите скрипт **AssetManagerReportingInstall.bat**, расположенный в корневой папке предоставленного файла **AssetManager_CRS_9.40.zip**.

   Примечание. Для выполнения скрипта необходимо использовать параметр "Запуск от имени администратора".

2. Будет предложена установка следующих языковых версий.

   ```
   Please choose one of the options below:
   1- Install English version
   2- Install German version
   3- Install Spanish version
   4- Install French version
   5- Install Russian version
   6- Install Italian version
   7- Install Japanese version
   8- Install Korean version
   9- Install Simplified Chinese version
   10- Install Nederlands version
   11- Install Portuguese version
   0-      Exit
   Your choice:
   ```

   Введите цифру, соответствующую языковой версии, которую необходимо установить.

3. Имеются два варианта выбора:

   ```
   Asset Manager Reporting Installer
   ================================
   Выберите один из предлагаемых ниже вариантов:
   1- Установить SAP BusinessObjects Business Intelligence Platform для Asset M
   anager
   0- Exit
   Your choice:1
   ```

   Для продолжения установки введите '1'.

4. Выберите метод установки, когда будут предложены следующие параметры.

   ```
   Asset Manager Reporting Installer
   ================================
   Установить SAP BusinessObjects Business Intelligence Platform для Asset Mana
   ger:
   1- Автоматическая установка
   2- Ручная установка
   0- Назад
   Ваш выбор: 1
   ```

   Для выбора режима автоматической установки введите '1'.

5. Введите пароль администратора для BusinessObjects Business Intelligence Platform.

6. Установка будет выполнена с использованием предварительно определенных опций. Процесс установки будет приостанавливаться несколько раз; его необходимо будет возобновлять вручную.

7. Чтобы импортировать Crystal Reports для Asset Manager, введите пароль администратора, настроенный в шаге 5.

Если требуется создать или изменить отчеты Crystal Reports, можно открыть SAP Crystal Reports (Designer) для Asset Manager, запустив файл `AssetManagerReportingInstall.bat`, который находится в корневом каталоге **AssetManager_CRD_9.40.zip**. Шаги установки SAP Crystal Reports (Designer) аналогичны шагам, описанным выше.

Установка SAP Crystal Reports (Designer) и SAP BusinessObjects Business Intelligence Platform для Asset Manager вручную

Если требуется настроить установку SAP BusinessObjects Business Intelligence Platform для Asset Manager и/или SAP Crystal Reports (Designer) для Asset Manager, нужно выполнить установку вручную. Например, ручную установку можно использовать , чтобы настроить порт подключения к серверу CMS, порты Tomcat, тип базы данных и т.д.

Подробную информацию можно найти в руководствах по бизнес-объектам (BusinessObjects), которые можно найти на сайте: http://help.sap.com/businessobject/product_guides/.

Если отсутствует доступ к серверу документации, следующие документы по SAP BusinessObjects Business Intelligence Platform можно найти в папке **<установочный носитель Asset Manager>:\Doc**.

- Заметки о выпуске (Release Notes)

- Руководство по установке (английская версия)

Чтобы установить SAP BusinessObjects Business Intelligence Platform для Asset Manager или SAP Crystal Reports для Asset Manager вручную, выполните следующие действия.

1. Запустите скрипт **AssetManagerReportingInstall.bat**, расположенный в корневой папке предоставленного файла **AssetManager_CRS_9.40.zip**.

2. Будет предложена установка следующих языковых версий.

```
Выберите один из предлагаемых ниже вариантов:
1- Install English version
2- Install German version
3- Install Spanish version
4- Install French version
5- Install Russian version
6- Install Italian version
7- Install Japanese version
8- Install Korean version
9- Install Simplified Chinese version
10- Install Nederlands version
11- Install Portuguese version
```

```
0-      Exit
Ваш выбор:
```

Введите цифру, соответствующую языковой версии, которую необходимо установить.

3. Имеются два варианта выбора:

```
Asset Manager Reporting Installer
=================================
Выберите один из предлагаемых ниже вариантов:
1- Установить SAP BusinessObjects Business Intelligence Platform для Asset M
anager
0- Выход
Ваш выбор: 1
```

Для продолжения установки введите '1'.

4. Выберите метод установки, когда будут предложены следующие параметры.

```
Asset Manager Reporting Installer
=================================
Установить SAP BusinessObjects Business Intelligence Platform для Asset Mana
ger:
1- Unattended installation
2- Manual installation
0- Back
Your choice:2
```

Для выбора режима ручной установки введите '2'.

5. Далее выполните процедуру, указанную мастером установки.

   Дополнительные сведения о процедуре установки SAP можно найти в руководстве `Business Intelligence platform Installation Guide for Windows` на сайте http://help.sap.com/bobip.

Если требуется создать или изменить отчеты Crystal Reports, можно открыть SAP Crystal Reports (Designer) для Asset Manager, запустив файл `AssetManagerReportingInstall.bat`, который находится в корневом каталоге **AssetManager_CRD_9.40.zip**. Шаги установки SAP Crystal Reports (Designer) аналогичны шагам, описанным выше.

Контрольный список конфигурации для автоматической и ручной установки

Ручная установка SAP BusinessObjects Business Intelligence Platform для Asset Manager требует выполнения дополнительных процедур настройки вручную. В следующей таблице содержится контрольный список конфигураций.

Если некоторые настройки не выполнены автоматически в режиме автоматической установки, проверьте следующую таблицу и выполните их вручную.

Примечание. Независимо от того, какой режим выбран, прочитайте главу целиком перед началом установки.

	Где проверять конфигурацию	Критерии	Реализация в процессе автоматической установки	Этапы процесса ручной реализации
1. Файл **u2lamlib.dll**	**<Полный путь к папке установки BusinessObjects>\SAP BusinessObjects Enterprise <версия>\win32_x86**	Файл **u2lamlib.dll** импортируется в эту папку	✓	См. раздел "Импорт файла u2lamlib.dll (требуется для установки как в автоматическом, так и в ручном режиме) " на стр.87.
2. Демонстрационные отчеты, поставляемые с Asset Manager 9.40, импортируются в BusinessObjects Business Intelligence Platform	Экран **Folders** (Папки) центральной консоли управления SAP BusinessObjects	Создается папка **Asset Manager**, куда также включаются демонстрационные отчеты.	✓[1]	Отчеты могут импортироваться с помощью LCMBIAR-файла или отдельных RPT-файлов. См. раздел "Импорт LCMBIAR-файла с помощью средства SAP Promotion Management Tool" на стр.88. См. раздел "Импорт RPT-файлов Asset Manager вручную" на стр.89.

	Где проверять конфигурацию	Критерии	Реализация в процессе автоматической установки	Этапы процесса ручной реализации
3. Учетная запись Asset Manager создана в BusinessObjects Business Intelligence Platform	Экран **Users and Groups** (Пользователи и группы) центральной консоли управления	Создается пользователь **AssetManager** с уровнем доступа **Полный доступ**.	✓[2]	См. раздел "Создание пользователя интеграции Asset Manager в SAP BusinessObjects Business Intelligence Platform (требуется для установки вручную)" на стр.89.

	Где проверять конфигурацию	Критерии	Реализация в процессе автоматической установки	Этапы процесса ручной реализации
4. Указывается источник данных отчетов Asset Manager	Страница **Конфигурация базы данных** для параметров по умолчанию папки **Asset Manager**	Установлен параметр **Использовать указанную здесь пользовательскую информацию для входа в базу данных** и выбраны следующие свойства источника данных. • Тип базы данных: Выберите драйвер базы данных, ODBC • Сервер: базы данных Asset Manager • База данных: Например, AMDemo94en • Пользователь: Admin • Префикс таблицы: используйте префикс таблицы по умолчанию • При просмотре	✓ **Примечание.** Свойство базы данных необходимо обновить вручную.	См. раздел "Задайте свойство отчета — "Конфигурация базы данных" (требуется для установки как в автоматическом, так и в ручном режиме)" на стр.91.

	Где проверять конфигурацию	Критерии	Реализация в процессе автоматической установки	Этапы процесса ручной реализации
		отчета: Предлагать пользователя для входа в базу данных (этот параметр требуется для механизма единой регистрации).		
5. Реализована единая регистрация	**<Папка установки сервера веб-приложения BusinessObjects Business Intelligence Platform>\lib**	В эту папку импортируются файлы **am-reporting.common-94.jar** и **am-reporting.serverside-94.jar**.	✓ **При-мечание.** Требуется вручную обновить файл web.xml приложения Asset Manager Web Tier и "Настройте пароль для единого входа:" на стр.98.	См. раздел "Настройка единого входа (SSO)" на стр.95.
	Файл **web.xml** в Asset Manager Web Tier	Обновлен раздел **SSO filter** (Фильтр SSO).		
	файл **web.xml** приложения BusinessObjects Business Intelligence Platform	Добавлен раздел **SSO filter** (Фильтр SSO).		
	<Папка установки веб-приложения BusinessObjects Business Intelligence Platform>\WEB-INF\classes	Файл **reporting.properties** импортируется в эту папку.		

	Где проверять конфигурацию	Критерии	Реализация в процессе автоматической установки	Этапы процесса ручной реализации
6. Настроено вычисляемое поле **sysCoreWebCrystal**	Экран **Вычисляемые поля** клиента Asset Manager (ссылка для переходов **Администрирование/ Система/ Вычисляемые поля**)	Обновляется вычисляемое поле **Отобразить отчет** (sysCoreWebCrystal).		См. раздел "Настройка вычисляемого поля sysCoreWebCrystal" на стр.100.
7. Объявляется URL-адрес сервера приложений, на котором размещено приложение SAP BusinessObjects Business Intelligence Platform	Запустите мастер **Изменение URL-адресов серверов приложений...** (BstBackEndOpt) (ссылка для переходов **Администрирование/ Действия пользователя/ Изменение URL-адресов серверов приложений...**).	**URL-адрес сервера** сервера **SAP BusinessObjects Crystal Reports Server**: **http или https://<имя сервера SAP BusinessObjects Business Intelligence Platform>:<порт, используемый SAP BusinessObjects Business Intelligence Platform>**.		См. раздел "Объявление URL-адреса доступа к SAP BusinessObjects Business Intelligence Platform" на стр.99.
8. **Имя файла** отчета в Asset Manager сопоставлено с кодом отчета в SAP BusinessObjects Business Intelligence Platform	Экран отчетов (ссылка для переходов **Администрирование/ Отчеты/ Отчеты**)	Поле **Файл** заполняется значением **Код** отчета на странице **Свойства** параметров отчета по умолчанию.		См. раздел "Сопоставление имени файла отчета с кодом отчета Crystal" на стр.100.

	Где проверять конфигурацию	Критерии	Реализация в процессе автоматической установки	Этапы процесса ручной реализации

- [1]: В режиме автоматической установки отчеты Asset Manager автоматически импортируются посредством развертывания файла LCMBIAR (Business Intelligence Archive Resource) в базе данных SAP BusinessObjects Business Intelligence Platform с помощью скрипта **tool_step2_deployReports.bat** (расположен в папке **install**). Таким образом, в тех редких случаях, когда импорт отчетов завершается неудачей, можно вручную запустить скрипт **tool_step2_deployReports.bat** для повторного импорта отчетов.
- [2]: В режиме автоматической установки пользователь AssetManager создается автоматически; однако требуется вручную предоставить права полного доступа этому пользователю — см. раздел "Назначение прав" на стр.90.

Конфигурации для доступа веб-клиентов **Asset Manager** к отчетам, хранящимся в базе данных **BusinessObjects Business Intelligence Platform**

Если вы выполнили установку предоставленной OEM-версии SAP BusinessObjects Business Intelligence Platform для Asset Manager в автоматическом режиме, не требуется выполнять все процедуры настройки, описанные в этом разделе (вручную требуется настроить только веб-клиент Asset Manager и сведения о домене).
Сведения о процедурах, необходимых для автоматической установки см. в разделе "Контрольный список конфигурации для автоматической и ручной установки" на стр.81.

Однако, если вы выполнили ручную установку предоставленной OEM-версии SAP BusinessObjects Business Intelligence Platform для Asset Manager или установили полную коммерческую версию SAP BusinessObjects Business Intelligence Platform, для интеграции необходимы следующие дополнительные этапы настройки.

Импорт файла u2lamlib.dll (требуется для установки как в автоматическом, так и в ручном режиме)

- Если Asset Manager и SAP BusinessObjects Business Intelligence Platform установлены на одном сервере, настройки файла **u2lamlib.dll** не требуется.
- Если Asset Manager и SAP BusinessObjects Business Intelligence Platform установлены на разных серверах, выполните следующие действия.

a. Распакуйте и откройте архив **AssetManager-CRS-9.40.zip**.

b. Отобразите содержимое папки **lib\EN**.

c. Скопируйте файл **u2lamlib.dll**.

d. Вставьте этот файл в папку **<полный путь к папке установки BusinessObjects>\SAP BusinessObjects Enterprise <версия>\win32_x86**. Например:

```
C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise
XI 4.0\win32_x86
```

Импорт отчетов Asset Manager в SAP BusinessObjects Business Intelligence Platform (требуется для установки вручную)

Отчеты Asset Manager можно импортировать в SAP BusinessObjects Business Intelligence Platform несколькими способами.

Импорт отчетов в BusinessObjects Business Intelligence Platform 4.1 из старых BIAR-файлов

Архивные BIAR-файлы, созданные в предыдущих версиях продуктов SAP BI, не работают в BusinessObjects Business Intelligence Platform 4.x. Вместо них приложение BusinessObjects Business Intelligence Platform использует архивные LCMBIAR-файлы.

Таким образом, если имеются BIAR-файлы отчетов Asset Manager, и вы планируете обновление до BusinessObjects Business Intelligence Platform 4.1, сначала необходимо запустить программу SAP Upgrade Management Tool, чтобы импортировать отчеты, содержащиеся в BIAR-файлах. Дополнительные сведения о программе SAP Upgrade Management Tool см. на сайте http://scn.sap.com/community/bi-platform/blog/2013/04/16/bi4-upgrade-management-tool.

После завершения импорта BIAR-файла, отчеты Asset Manager будут доступны в папке **Asset Manager**.

Импорт LCMBIAR-файла с помощью средства SAP Promotion Management Tool

В версии Asset Manager, совместимой с BusinessObjects Business Intelligence Platform 4.x, в папке установки Crystal Reports доступен LCMBIAR-файл с именем `am_reports_<язык>.lcmbiar`. В этом файле содержатся все стандартные отчеты Asset Manager и соответствующие сведения. Чтобы импортировать этот LCMBIAR-файл, выполните следующие действия.

1. Войдите в центральную консоль управления, откройте средство Promotion Management Tool.

2. Нажмите **Import** (Импорт), а затем выберите **Import file** (Импорт файла).

3. Нажмите кнопку "Обзор", выберите файл `am_reports_<язык>.lcmbiar` в папку установки Crystal Report, например, `C:\AssetManager_CRS_9.40\lib\reports`, а затем нажмите **OK**.

4. Укажите значение для поля **Назначение** и нажмите **Создать**. Будет создано задание.

5. На странице нового задания нажмите **Promote**.

6. На вкладке **Security settings** выберите **Promote Security**, установите все три флажка и нажмите **Save**.

7. На вкладке **Test Promote** нажмите **Test Promote**. Убедитесь в том, что результаты теста соответствуют вашим ожиданиям.

8. На вкладке **Summary** нажмите **Promote**.

9. Обновите страницу и проверьте статус задания.

10. После того, как статус изменится на **Success**, выберите вкладку **Папки**.

11. Нажмите **Все папки**, откройте **Asset Manager**, в разделе **User Security** убедитесь, что пользователь **Asset Manager** имеет полный доступ.

12. Все отчеты доступны в папке **Asset Manager**.

Импорт RPT-файлов Asset Manager вручную

В приложение SAP BusinessObjects Business Intelligence Platform можно также импортировать отдельные отчеты Asset Manager (RPT-файлы). Для этого должно быть установлено приложение SAP Crystal Reports Designer, которое должно иметь доступ к серверу, на котором размещено приложение Asset Manager (для доступа к отчетам).

В этом случае:

1. Запустите SAP Crystal Reports Designer **Report Update Wizard** (**Пуск/ Все программы/ SAP Business Intelligence/ SAP Crystal Reports 2013/ Report Upload Wizard**).

2. Выполните мастер, выбрав отчеты для импорта (**<папка установки Asset Manager>\Datakit\bestprac\reports\rpt**) и указав расположение (создайте папку **Asset Manager** для отчетов) в SAP BusinessObjects Business Intelligence Platform, куда требуется поместить отчеты.

Создание пользователя интеграции Asset Manager в SAP BusinessObjects Business Intelligence Platform (требуется для установки вручную)

- Если импорт отчетов выполнен с помощью предоставленного архивного LCMBIAR-файла, в SAP BusinessObjects Business Intelligence Platform будет импортирован пользователь отчетов с именем "Asset Manager".

- В противном случае, можно создать пользователя интеграции вручную, см. раздел "Ручное создание" ниже.

Ручное создание

Ручное создание пользователя интеграции Asset Manager с SAP BusinessObjects Business Intelligence Platform предусматривает два этапа.

1. **Создание пользователя**

   Чтобы создать пользователя интеграции Asset Manager с SAP BusinessObjects Business Intelligence Platform, выполните следующие действия.

   a. Запустите центральную консоль управления.

   b. Выберите тип аутентификации **Enterprise** и выполните вход с учетной записью **Administrator** (пароль был создан в процессе установки).

   c. Щелкните ссылку **Пользователи и группы** в группе **Организовать**.

   d. Выберите **Диспетчер/ Создать/ Создать пользователя** на экране **Пользователи и группы**.

   e. Заполните следующие свойства:

      - Имя учетной записи: AssetManager
      - Полное имя: AssetManager
      - Пароль: введите пароль учетной записи AssetManager и его подтверждение
      - Выберите **Пароль с неограниченным сроком действия**.
      - Снимите флажок **Требовать смену пароля при следующем входе в систему** .
      - Снимите флажок **Запретить смену пароля пользователем**.
      - Тип подключения: Одновременно работающий пользователь

   f. Нажмите кнопку **Создать**. Появится сообщение о том, что создана учетная запись AssetManager.

2. Назначение прав

   Отчеты Asset Manager уже должны быть импортированы в SAP BusinessObjects Business Intelligence Platform.

   Чтобы назначить все права для отчетов Asset Manager; пользователю AssetManager:

   a. Запустите центральную консоль управления.

   b. Щелкните **Папки** (в группе **Организовать**).

c. Щелкните правой кнопкой мыши папку, в которую были импортированы отчеты Asset Manager (имя папки указывается во время импорта отчетов).

d. Выберите в контекстном меню **Безопасность пользователя**.

e. Щелкните **Добавить участников**.

f. Выберите **AssetManager AssetManager** в поле **Доступные пользователи/группы**.

g. Щелкните **>** для добавления пользователя в поле **Выбранные пользователи/группы**.

h. Щелкните **Добавить и назначить безопасность**.

i. Не изменяйте выбор по умолчанию для **Наследование** и выберите вкладку **Уровни доступа** в открывшемся окне.

j. Выберите **Полный контроль** в таблице **Доступные уровни доступа**.

k. Щелкните **>** для добавления уровня в таблицу **Назначенные уровни доступа**.

l. Нажмите кнопку **OK**.

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

Задайте свойство отчета — "Конфигурация базы данных" (требуется для установки как в автоматическом, так и в ручном режиме)

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

По умолчанию используется значение, задаваемое в конфигурации базы данных, — "AMDemo94<язык>". Если демонстрационная база данных не используется, обновите конфигурацию базы данных с помощью одного из следующих методов.

Измените исходную базу данных с помощью модуля Central Management Console в SAP BusinessObjects Business Intelligence Platform

1. Запустите центральную консоль управления.

2. Щелкните **Папки** в группе **Организовать**.

3. Дважды щелкните папку, в которую были помещены импортированные файлы, например, папку **Asset Manager**.

4. Дважды щелкните отчет, например, демонстрационную базу данных **Анализ уровней потребительских запасов**.

5. В навигационном меню выберите **Параметры по умолчанию/ Конфигурация базы данных**.

6. Выберите параметр **Использовать указанную здесь пользовательскую информацию для входа в базу данных**.

7. Заполните следующие свойства:

   - Тип базы данных: Выберите драйвер базы данных, ODBC
   - Сервер: базы данных Asset Manager
   - База данных: Например, AMDemo94en
   - Пользователь: Admin
   - Пароль: пароль пользователя базы данных

     Примечание. По умолчанию используется **<пустой>** пароль, если это поле было оставлено пустым.

   - Префикс таблицы: используйте префикс таблицы по умолчанию
   - При просмотре отчета: Предлагать пользователя для входа в базу данных (этот параметр требуется для механизма единой регистрации).

8. нажмите кнопку **Сохранить**.

Изменение исходной базы данных с помощью SAP Crystal Reports (Designer)

1. Запустите SAP Crystal Reports (Designer).

2. Откройте файл **.rpt** для модификации.

3. В меню **База данных** выберите **Установить местоположение источника данных**.

4. Разверните узел **Отчеты** и выберите **Базы данных Asset Manager** в поле «Текущий источник данных».

5. Выберите имя базы данных, например, AMDemo94en. Нажмите кнопку **Далее**.

6. Заполните поле **ИД пользователя**, используя следующий формат: **<ИД пользователя>@<База>**, например: **Admin@AMDemo94en**.

7. В Asset Manager отобразится диалоговое окно **Подключение к базе данных**. Введите пароль (если пользователь - **Admin**, пароль является пустым по умолчанию). Далее щелкните **Открыть**.

8. Если существуют исходные базы данных для дополнительных отчетов, выполните аналогичную операцию для каждой из них.

9. Закройте окно.

Изменение исходной базы данных с помощью командной строки

Отчеты Asset Manager уже должны быть импортированы в SAP BusinessObjects Business Intelligence Platform.

> **Примечание.** Поскольку файлы в ZIP-архиве доступны только для чтения, скопируйте содержимое ZIP-файла на локальный диск, а затем выполните следующие действия.

Чтобы автоматически модифицировать исходную базу данных:

1. Откройте командную строку DOS и перейдите в каталог **install** (или в соответствующий локальный каталог, если содержимое ZIP-файла было скопировано на рабочую станцию).

   ```
   cd <имя диска>:\install
   ```
   Например:

   ```
   cd D:\install
   ```

2. Если приложение SAP BusinessObjects Business Intelligence Platform было установлено не в папку по умолчанию, укажите папку установки SAP BusinessObjects Business Intelligence Platform с помощью следующей команды:

   ```
   set BOBJ_InstallDir=<папка установки SAP BusinessObjects Business Intelligen
   ce Platform>
   ```
   Например:

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

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

   ```
   set AMRootReportsFolder=<папка импорта отчетов>
   ```
   Например:

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

4. Вы собираетесь выполнить скрипт **tool_step3_buildReportsList.bat**, чтобы скомпилировать список отчетов Asset Manager, загруженных в SAP BusinessObjects Business Intelligence Platform. 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
   ```

5. Выполните скрипт **tool_step3_buildReportsList.bat** с помощью следующем команды:

   ```
   buildReportsList.bat
   ```

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

> **Примечание.** Если используется установка вручную и определен порт CMS, необходимо изменить свойство **CMSport**
> в файле resource.bat папки lib.

6. Чтобы исключить отчеты из импортирования в базу данных Asset Manager, откройте файл, сгенерированный скриптом **tool_step3_buildReportsList.bat**, и вставьте символ **#** в начало каждой строки отчета, который необходимо исключить.

7. 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, DE, ES, FR, RU, IT, JP, KO, CHS, PT, NL**.

8. Запустите проводник Windows.

9. Перейдите в папку **\lib\reports**.

10. Скопируйте файл **am_reportsDSDef_xxx.dat**, где **xxx** представляет язык установки SAP Crystal Reports.

11. Вставьте его в любую папку.

12. Отредактируйте файл **am_reportsDSDef_xxx.dat**.

13. Введите параметр после **база данных**, указав имя подключения к базе данных Asset Manager (см. раздел "Создание подключения к базе данных Asset Manager" на стр.77).
    Не модифицируйте другие параметры.

14. 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 измененного файла am_reportsDSDef_xxx.dat >
    ```

15. Введите следующую команду для выполнения скрипта **tool_step4_changeReportsDS.bat**:

    ```
    tool_step4_changeReportsDS.bat
    ```

    При выполнении скрипта необходимо ввести пароль администратора Crystal Report.

> **Примечание.** Если используется установка вручную и определен порт CMS,

необходимо изменить свойство **CMSport**
в файле resource.bat папки lib.

16. Теперь вы связали пользовательскую исходную базу данных с каждым отчетом Asset Manager в SAP BusinessObjects Business Intelligence Platform.

Настройка единого входа (SSO)

Настройка единого входа позволяет избежать необходимости повторных аутентификаций (при подключении к веб-клиенту Asset Manager, при доступе к отчету, или в случае необходимости получения данных для отчета из базы данных Asset Manager).

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

1. Скопируйте файлы **am-reporting.common-94.jar** и **am-reporting.serverside-94.jar** (находятся в папке **lib\sso**) в папку **lib** BusinessObjects Business Intelligence Platform для веб-приложения **Asset Manager**. Пример. **C:\Program Files (x86)\SAP BusinessObjects\tomcat\webapps\clientapi\WEB-INF\lib**.

2. Обновите файл **web.xml** в **Asset Manager** Web tier (например, C:\Tomcat\webapps\AssetManager\WEB-INF\web.xml). Найдите раздел **SSO filter** и убедитесь, что имеются следующие две конфигурации. Если данные конфигурации отсутствуют, добавьте их вручную.

```
<!-- Define SSO Filter and associated mapping -->
<filter>
<filter-name>SSO Filter</filter-name>
<filter-class>com.hp.sw.bto.reporting.sso.client.SSOClientFilter</filter-cla
ss>
</filter>
```

```
<!-- Crystal Reports Server SSO filter mapping   -->
<!-- MUST be declared after Acegi filter mapping -->
<filter-mapping>
<filter-name>SSO Filter</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>
```

3. Обновите файл **web.xml** приложения BusinessObjects Business Intelligence Platform (например, C:\Program Files(x86)\SAP BusinessObjects\tomcat\webapps\clientapi\WEB-INF). Найдите раздел **SSO filter** и убедитесь, что имеются следующие четыре конфигурации. Если данные конфигурации отсутствуют, добавьте их вручную.

```
 <!-- Define SSO Filter and associated mapping -->
<!-- Define Asset Manager SSO Filter and associated mapping -->
<filter>
<filter-name>HP Asset Manager SSO Authentication Filter</filter-name>
<filter-class>com.hp.sw.bto.reporting.sso.server.AuthenticationFilter</filte
r-class>
</filter>



<filter-mapping>
<filter-name>HP Asset Manager SSO Authentication Filter</filter-name>
<url-pattern>/ErsViewerServlet</url-pattern>
</filter-mapping>



<!-- The mapping for the JSP servlet -->
<servlet>
<servlet-name>HP Asset Manager SSO Authentication Servlet</servlet-name>
<servlet-class>com.hp.sw.bto.reporting.sso.server.DbLogonViewerServlet</serv
let-class>
</servlet>



<servlet-mapping>
<servlet-name>HP Asset Manager SSO Authentication Servlet</servlet-name>
<url-pattern>/ErsViewerServlet/*</url-pattern>
</servlet-mapping>
```

4. Скопируйте файл **reporting.properties** (находится в папке **lib\sso**) в папку **WEB-INF\classes** веб-приложения BusinessObjects Business Intelligence Platform (пример: C:\Program Files (x86)\SAP BusinessObjects\tomcat\webapps\clientapi\WEB-INF\classes). Вы можете отредактировать этот файл и настроить его параметры, включая:

Имя параметра	Значение по умолчанию	Описание параметра
server.bo.shared_secret	Неприменимо	Скопируйте и вставьте значение общего секрета из модуля SAP Central Management Console в этот параметр.

Имя параметра	Значение по умолчанию	Описание параметра
server.bo.admin_account	Administrator	Учетная запись администратора SAP Crystal Reports (используется только для автоматического создания пользователя).
server.bo.cms_port	7400	Номер порта базы данных SAP BusinessObjects Business Intelligence Platform.
server.bo.default_user	AssetManager	Если данный параметр определен, подключение к SAP BusinessObjects Business Intelligence Platform будет выполнено с использованием этого имени пользователя, а не имени пользователя, предоставленного механизмом единого входа (SSO). Если имя пользователя не существует, оно автоматически создается, если был активирован параметр **on_the_fly_creation**. В противном случае, попытка подключения завершится сбоем.
server.bo.user.on_the_fly_creation	true	Авторизовать или отклонить автоматическое создание пользователя в SAP BusinessObjects Business Intelligence Platform (пользователь будет создан или не будет создан в базе данных SAP BusinessObjects Business Intelligence Platform во время своего первого подключения, если пользователь еще не существует в базе данных).
server.bo.user.can_change_password	true	Свойства SAP Crystal Reports для пользователя, созданного автоматически.
server.bo.user.must_change_password	false	Idem
server.bo.user.password_expires	false	Idem
server.bo.user.is_named	false	Idem

Имя параметра	Значение по умолчанию	Описание параметра
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.

5. Настройте пароль для единого входа:

   a. Запустите центральную консоль управления.

   b. Выберите тип аутентификации **Enterprise** и выполните вход с учетной записью **Administrator** (пароль был создан в процессе установки).

   c. Щелкните **Authentication** (Аутентификация) в группе **Manage** (Управление).

   d. Дважды щелкните запись **Enterprise**, затем выберите параметр **Trusted Authentication is enabled** (Активирована доверенная аутентификация).

   e. Нажмите **New Shared Secret**.

   f. Нажмите **Download Shared Secret**, а затем откройте в текстовом редакторе файл `TrustedPrincipal.conf`.

   g. Скопируйте и вставьте значение общего секрета в свойство **server.bo.shared_secret** в файле **reporting.properties**.

   h. Щелкните **Обновить**.

Внимание! Механизм единого входа не будет работать, если хост, используемый для приложения Asset Manager, не использует тот же домен DNS, что и хост для SAP BusinessObjects Business Intelligence Platform. 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 SAP BusinessObjects Business Intelligence Platform, а также при подключении к базе данных Asset Manager.

Кроме того, для корректного функционирования единого входа необходимо ввести имя домена в:

- Имя сервера SAP BusinessObjects Business Intelligence Platform: "Объявление URL-адреса доступа к SAP BusinessObjects Business Intelligence Platform" ниже.
- Имя сервера Asset Manager Web Tier: **http://<имя сервера HP Asset Manager Web Tier с именем домена>:<порт, используемый HP Asset Manager Web Tier>/AssetManager**

Объявление URL-адреса доступа к SAP BusinessObjects Business Intelligence Platform

1. Запустите клиент Windows или веб-клиент Asset Manager.
2. Подключитесь к базе данных с правами администратора (Admin).
3. Запустите мастер **Изменение URL-адресов серверов приложений...** (BstBackEndOpt) (ссылка на навигационной панели **Администрирование/ Действия пользователя/ 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 **SAP BusinessObjects Crystal Reports Server**, заполните данное поле значением **SAP BusinessObjects Crystal Reports Server**, а затем нажмите кнопку **Добавление сервера приложения**.
Список приложений, строка, чей столбец **Приложение** равняется **SAP BusinessObjects Crystal Reports Server**, столбец **Адрес URL сервера**	http или https://<имя сервера SAP BusinessObjects Business Intelligence Platform>:<порт, используемый SAP BusinessObjects Business Intelligence Platform> Пример: `http://BusinessObjectServer:7080` **Внимание!** Для корректного функционирования единого входа необходимо включить имя домена в имя сервера SAP BusinessObjects Business Intelligence Platform.

Поле	Значение
Страница **Сводка изменений**	Проверьте содержимое страницы.
Щелкните **Завершить**	
Windows-клиент: нажмите **OK**	

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

Сопоставление имени файла отчета с кодом отчета Crystal

1. Запустите клиент Windows или веб-клиент Asset Manager.
2. Подключитесь к базе данных с правами администратора (Admin).
3. Выберите навигационное меню **Администрирование/ Домашняя бизнес-страница**.
4. Выберите домен, который необходимо добавить к отчету Crystal. Например, "Portfolio".
5. Выберите вкладку **Отчеты**.
6. Выберите отчет, который будет отображаться на домашней бизнес-странице, например: Анализ уровней потребительских запасов.
7. Отобразите сведения каждого из выбранных отчетов и обновите значение поля **Файл** кодом отчета Crystal (доступно в значении **Код** на странице **Свойства** параметров отчета по умолчанию в модуле Central Management Console веб-приложения BusinessObjects Business Intelligence Platform). Например, для анализа уровней потребительских запасов используется отчет Crystal с кодом 1200.

Настройка вычисляемого поля sysCoreWebCrystal

Чтобы включить доступ к отчетам с помощью веб-клиента Asset Manager, необходимо изменить поле **sysCoreWebCrystal**.

Чтобы модифицировать вычисляемое поле **sysCoreWebCrystal**:

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

4. Замените синтаксис BASIC следующими скриптами:

```
Dim strURL as String
Dim strServerCrystal as String
Dim strFileName as String
strFileName = [FileName]
strURL = ""
if IsNumeric(strFileName) then
strServerCrystal = amBackEndServerPath("SAP BusinessObjects Crystal Reports
Server")
if Len(strServerCrystal)>0 and Len(strFileName)>0 then
strURL = strServerCrystal
strURL = strURL + "/clientapi/ErsViewerServlet?reportId="
strURL = strURL + strFileName
end if
end if
RetVal = strURL
```

Миграция с BusinessObjects Enterprise XI 3.1 на BusinessObjects Business Intelligence Platform 4.1

Пользовательскую систему BusinessObjects Enterprise XI 3.1 (в том числе отчеты, права пользователей, и т.д.) можно перенести в BusinessObjects Business Intelligence Platform 4.1, что позволит корректно отображать отчеты Crystal Reports в клиенте Windows и веб-клиенте Asset Manager 9.40.

1. На сервере, на котором установлено приложение BusinessObjects Business Intelligence Platform 4.1, откройте средство Upgrade Management Tool.

2. Нажмите **Complete Upgrade**, а затем **Next**.

3. В раскрывающемся списке **Upgrade Scenario** выберите **Live to Live**.

4. В разделе **Source** введите соответствующие сведения о сервере BusinessObjects Enterprise XI 3.1.

5. В разделе **Destination** введите соответствующие сведения о сервере BusinessObjects Business Intelligence Platform 4.1.

   Примечание. Используйте <имя_хоста>:<порт> в качестве имени CMS.

6. Нажмите **Next**.

7. Нажмите **Start**.

Отображение отчета

В этом разделе содержатся следующие сведения.

Чтобы отобразить отчет 102
Доступные отчеты 104
Как следует отображать отчет 105
Как приложение Asset Manager создает URL-адреса отчетов 105

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.

   Примечание. Можно выбрать только одну запись.

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

   Совет. 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. Щелкните **Предварительный просмотр**.

4. Выберите запись, которая является объектом отчета.
5. Нажмите кнопку **OK**.

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

Разверните ссылку навигационной панели, которая соответствует функциональной области отчета.

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

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. Чтобы обновить отчет: щелкните значок .
   Чтобы задать частоту автоматического обновления отчета: щелкните значок правой кнопкой мыши.
   Чтобы изменить масштаб (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	Windows-клиент	Веб-клиент
Место хранения отчета	• База данных Asset Manager • SAP BusinessObjects Business Intelligence Platform	SAP BusinessObjects Business Intelligence Platform
	Внимание! Для веб-клиента и клиента Windows отчет доступен через SAP BusinessObjects Business Intelligence Platform, если BusinessObjects Business Intelligence Platform был объявлен с помощью мастера **Изменение URL-адресов серверов приложений...** (BstBackEndOpt), и если поле отчета **Файл** (база данных) содержит числовое значение.	
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

Windows-клиент:

- Если файл **.rpt** сохранен в базе данных Asset Manager: отчет отображается клиентом Windows.
- Если **RPT**-файл хранится не в базе данных Asset Manager, а в BusinessObjects Business Intelligence Platform: отчет отображается в веб-браузере.

Веб-клиент: отчет отображается веб-клиентом.

Как приложение **Asset Manager** создает **URL**-адреса отчетов

Приложение 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 SAP BusinessObjects Business Intelligence Platform, определенный мастером **Изменение URL-адресов серверов приложений...** (BstBackEndOpt)

  "Объявление URL-адреса доступа к SAP BusinessObjects Business Intelligence Platform" на стр.99.

- Текстовая строка, указанная в скрипте вычисляемого поля

  "Настройка вычисляемого поля sysCoreWebCrystal" на стр.100.

- Поле отчета **Файл** (FileName)

  "Сопоставление имени файла отчета с кодом отчета Crystal" на стр.100.

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

> **Примечание.** По умолчанию вычисляемое поле **Отобразить отчет** (sysCoreWebCrystal) настраивается для получения доступа к SAP BusinessObjects Business Intelligence Platform 4.1.

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
- SAP BusinessObjects Business Intelligence Platform

Чтобы определить, в какой базе данных нужно сохранить отчет Crystal Reports, см. раздел Место хранения отчетов.

В этом разделе содержатся следующие сведения.

Получение файлов .rpt 106
Хранение отчетов в базе данных Asset Manager 106
Хранение отчетов в SAP BusinessObjects Business Intelligence Platform 109

Получение файлов .rpt

Для Crystal Reports существуют несколько источников:

- Отчеты, создаваемые самостоятельно с помощью BusinessObjects Business Intelligence Platform.
- Отчеты, которые уже хранятся в базе данных Asset Manager.
  Если вы хотите переместить их в базу данных BusinessObjects Business Intelligence Platform, необходимо поочередно экспортировать их при помощи кнопки **Экспорт** на странице со сведениями об отчете в клиенте Windows.
- Отчеты о бизнес-данных Asset Manager.
  Эти отчеты расположены в подпапке **datakit** каталога установки Asset Manager.

  Примечание. 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.

Хранение отчетов в базе данных Asset Manager

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

Последовательный импорт отчетов в существующую базу данных Asset Manager

Для вставки отчетов друг за другом:

1. Запустите Windows-клиент приложения 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 выполните следующие действия.

1. Запустите Asset Manager Application Designer.
2. Выбрать меню **Файл/ Открыть**.
3. Выбрать опцию **Открыть файл описания базы данных - создать новую базу данных**.
4. Выберите файл **gbbase.xml**, расположенный в подпапке **config** каталога 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 выполните следующие действия.

1. Запустите Asset Manager Application Designer.
2. Выбрать меню **Файл/ Открыть**.

3. Выбрать опцию **Открыть файл описания базы данных - создать новую базу данных**.

4. Выберите файл **gbbase.xml**, расположенный в подпапке **config** каталога установки Asset Manager.

5. Запустите мастер создания баз данных (меню **Действие/Создать базу данных**).

6. Заполните страницы мастера следующим образом (перемещайтесь по страницам мастера 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	Импортировать бизнес-данные.
Использовать расширенные опции создания.	Не нужно выбирать эту опцию.

Страница "Параметры создания"

Поля	Значение
Пароль	Ввести пароль администратора. **Примечание.** Администратор базы данных Asset Manager – запись в таблице **Сотрудники и подразделения** (amEmplDept), для которой в поле **Имя** (Name) задано значение **Admin**. Имя входа подключения к базе данных хранится в поле **Имя пользователя** (UserLogin). Именем администратора является **Admin**. Пароль записывается в поле **Пароль** (LoginPassword).

Страница "Данные для импорта"

Поля	Значение
Доступные данные	Выбрать опцию **Crystal Reports**.
Остановить импорт при возникновении ошибки	Выбрать эту опцию для остановки импорта, если есть какие-то проблемы.
Файл журнала	Полное имя файла, в котором регистрируются все операции по импорту, включая сообщения об ошибках и предупреждения.

7. Выполнить указанные опции с помощью мастера (клавиша **Завершить**).

Хранение отчетов в SAP BusinessObjects Business Intelligence Platform

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 Central Management Console в BusinessObjects Business Intelligence Platform.
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.

     > **Совет.** Чтобы получить идентификаторы отчетов, можно также выполнить запрос, подобный приведенному ниже, с помощью конструктора запросов BusinessObjects Business Intelligence Platform:
     >
     > ```
     > SELECT SI_NAME, SI_ID FROM CI_INFOOBJECTS WHERE SI_PROGID = 'CrystalEn
     > terprise.Report'
     > ```
     >
     > Еще можно использовать файл, сгенерированный **buildReportsList.bat**: См. раздел "Изменение исходной базы данных с помощью командной строки" на стр.93.

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

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

6. Запустите клиент 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	имя SQL поля или ссылки	Комментарий
Метка	Название	Имя отчета в BusinessObjects Business Intelligence Platform **Примечание.** Это поле необязательно, но может оказаться полезным.
Файл	FileName	Идентификатор отчета в BusinessObjects Business Intelligence Platform

Изменение отчета Seagate Crystal

В этом разделе содержатся следующие сведения.

Отчеты, хранящиеся в базе данных Asset Manager 110
Отчеты, хранящиеся в SAP BusinessObjects Business Intelligence Platform 111

Отчеты, хранящиеся 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 BusinessObjects Business Intelligence Platform 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, хранящиеся в SAP BusinessObjects Business Intelligence Platform

Измените отчет непосредственно при помощи BusinessObjects Business Intelligence Platform, не изменяя его код в SAP BusinessObjects Business Intelligence Platform (см. раздел "Хранение отчетов в SAP BusinessObjects Business Intelligence Platform" на стр.109).

Создание отчета со сведениями

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

В этом разделе содержатся следующие сведения.

Пример использования 111
Конфигурация отчетов в SAP Crystal Reports 112

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 SAP Crystal Reports

Чтобы получить отчет со сведениями, используйте приведенную ниже процедуру (пример выполнялся в SAP 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 отчетным>Id
   ```

   > **Примечание.** Необходимо соблюдать регистр имен SQL таблиц.

   Например, формула для создания контекстного отчета для таблицы активов:

   ```
   amAssetId
   ```

   > **Примечание.** 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 учетом регистра) позволяет идентифицировать пользователя, печатающего отчета. При печати данная формула берет значение номера идентификатора (т.е. значение поля с именем SQL: "lEmplDeptId" для текущего имени) пользователя, подключенного 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
   ```

   > **Примечание.** 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 контекстной таблицы>.<имя SQL поля, являющегося первичным ключом>}
   = @<имя поля формулы>}
   ```

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

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

С помощью описанной выше процедуры приложение 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 SAP Crystal Report.

Печать отчета

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

В этом разделе содержатся следующие сведения.

Отчет со сведениями (только в Windows-клиенте) 113
Графики и отчеты со списками 114

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.

   Примечание. Можно выбрать только одну запись.

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

   Совет. 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. Нажмите кнопку **OK**.
6. Щелкните значок **Печать**.

С помощью 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.

   Совет. 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. Нажмите кнопку **OK**.
5. Щелкните значок **Печать**.

С помощью 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 приложения 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 **Домен** (Domain) к столбцам списка.
6. Нажмите кнопку **OK**.
7. Отсортируйте список по столбцу **Домен**.
8. Отчеты для какого-либо модуля идентифицируются по имени области.
   Пример: **/Asset lifecycle/Software Asset Management/Reports/**.

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**, 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**.

Ограничения

- Использование отчетов SAP 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 SAP BusinessObjects Business Intelligence Platform.

  Этот тип ограничений описан в руководстве **Web Implementation**, глава **Differences between the Windows client and the Web client** (Различия между Windows-клиентом и веб-клиентом), раздел **Reporting** (Отчеты).

- В приложении Asset Manager не поддерживаются динамические параметры. Запуск отчета Crystal Report с динамическим параметром может привести к тому, что приложением Asset Manager перестанет отвечать.

  Примечание. Динамические параметры используются для интерактивного выбора данных из базы данных при выполнении отчета.

Глава 5: Инфопанели

В приложении Asset Manager можно 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.

Глава включает следующее:

Создание инфопанели 117

Пример инфопанели 118

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.

   Примечание. Дополнительные сведения см. в главе Действия в данном руководстве.

4. На вкладке **Индикаторы** выберите индикаторы, которые необходимо добавить к инфопанели.
   Информация о привязке индикатора хранится в таблице **Элемент инфопанели** (amDashboardItem).
   Два типа объектов используются в качестве индикаторов для инфопанелей в Asset Manager: отчеты и статистика. В этом разделе не рассматривается создание статистики или отчетов: Создание отчетов SAP Crystal Reports подробно описано в главе SAP Crystal Reports данного руководства, а использование статистики — в главе

Статистика.

Для привязки индикатора:

a. Выберите в сведениях об инфопанели вкладку **Индикаторы** и нажмите значок (клиент Windows) или **Добавить** (веб-клиент).

b. Выберите **Имя**, чтобы указать индикатор в инфопанели.

c. Определите **Тип** индикатора (отчет или статистика).

d. Наконец, выберите **Отчет** или **Статистика**.

Примечание. Щелкните (клиент Windows) или **Удалить** (веб-клиент) для удаления индикатора с 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**.

Примечание. Параметр **Статистика/Периодичность обновления статистики** доступен в меню **Изменить/ Параметры** в клиенте Windows.

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 работы в компании

Данные индикаторы были созданы в разделе Статистика, не заданная скриптом главы Статистика. См. сведения в этом разделе для создания статистики, необходимой в этом примере.

Этап 1: создание инфопанели

1. Выберите меню навигации **Администрирование/ Действия** и щелкните **Создать** для создания нового действия.

2. Заполните поля новой записи следующими данными:

Поле	Значение
Имя	Инфопанель ОК
Тип	Инфопанель

3. Подтвердите создание с помощью кнопки **Создать** (клиент Windows) или **Сохранить** (веб-клиент).

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 (клиент Windows) или **Добавить** (веб-клиент).

2. Заполните поле **Имя** индикатора. Можно использовать имена статистики, которую вы хотите добавить.

3. Выберите **Статистику** как **Тип** индикатора.

4. Нажмите (клиент Windows) или список выбора (веб-клиент) рядом с полем **Статистика**, а затем выберите статистику, которую необходимо добавить.

5. Нажмите **Добавить** (клиент Windows: или кнопку **Добавить** с двойными стрелками, чтобы добавить несколько индикаторов).

Если используется клиент Windows, нажмите кнопку **Изменить**.

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, в приложении Asset Manager откроется окно подразделений и сотрудников, которое фильтрует данные в поле **Г-н/Г-жа** (MrMrs), используя **Г-н** как значение для фильтра.

Примечание. Для отчетов навигация недоступна.

Глава 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. См. раздел Просмотр статистики

Не следует путать **статистику**, описанную в этой главе, со **статистикой SAP Crystal Reports**, описанной в главе Отображение отчета при помощи автоматического обновления (только в Windows-клиенте):

- Статистика SAP Crystal Reports используется для отображения отчета SAP 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 SAP Crystal Reports**.

Глава включает следующее:

Создание статистики 121

Использование скрипта вместо запроса 123

Оптимизация работы 124

Масштабирование сведений о статистике 125

Определение максимального количества групп для индивидуального представления 125

Просмотр статистики 126

Отображение статистики 127

Примеры статистики 127

Определение настраиваемых цветов для статистики в виде диаграммы 143

Создание статистики

Чтобы создать статистику:

1. Отобразите статистику (ссылка **Администрирование/ Отчеты/ 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).

   См. раздел Оптимизация работы.

4. Определить статистику можно одним из двух способов:
   - Используя стандартный **запрос**:
     i. Не устанавливайте флажок в ячейке **Скрипт** (bScript).
     ii. Во вкладке **Запрос** 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 партиям.
   - Написав сложный **скрипт** для отображения статистики
     i. Установите флажок в ячейке **Скрипт** (bScript).
     ii. Во вкладке **Скрипт** укажите скрипт, используемый для вычисления статистических данных.

         См. параграф Использование скрипта вместо запроса, чтобы узнать о различиях между скриптами и запросами.

         см. параграф Статистика, заданная скриптом, 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. См. раздел Просмотр статистики).

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 нажмите **Создать** (клиент Windows) или **Сохранить** (веб-клиент).

Если используется клиент Windows, диаграмму можно просмотреть с помощью следующих действий.

1. Нажмите кнопку **Проверка**.
2. Нажмите **ESC**, чтобы закрыть диаграмму.

Использование скрипта вместо запроса

Как указано в параграфе Создание статистики, существует два способа описания статистики:

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

В этом разделе содержатся следующие сведения.

Использование скрипта вместо запроса 123
Синтаксис скриптов 124

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

См.:

- примеры в разделе Статистика, заданная скриптом, чтобы узнать, какой синтаксис используется при написании скриптов.
- Определение настраиваемых цветов для статистики в виде диаграммы

Оптимизация работы

В этом разделе содержатся следующие сведения.

Причины проблем в работе 124
Способ устранения проблем в работе 124
Внедрение этого способа 125

Причины проблем в работе

Статистика отображается посредством инфопанелей.

Когда отображается инфопанель, приложение Asset Manager получает 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 высчитываются 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, приложение 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 приложением Asset Manager Automated Process Manager, а не 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 приложение Asset Manager Automated Process Manager будет запускать статистику через установленные графиком интервалы для всей статистики, для которой включен параметр **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

Примечание. Эта функция доступна только в клиенте Windows.

Не рекомендуется 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

В этом разделе содержатся следующие сведения.

На инфопанелях 126
На домашних страницах компаний 127

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. Подтвердите создание с помощью кнопки **Создать** (клиент Windows) или **Сохранить** (веб-клиент).

5. Во вкладке **Индикаторы** создайте индикаторы статистики, выполнив следующую процедуру:

   a. Щелкните (клиент Windows) или **Добавить** (веб-клиент).

   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. Если используется клиент Windows, обновите инфопанель (кнопка **Изменить**). Инфопанель создана и ее можно использовать.

Чтобы использовать эту статистику в инфопанелях, прочитайте следующую главу: Инфопанели.

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.

Глава Инфопанели данного руководства содержит сведения об использовании статистики 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/ Отчеты/ 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	Средняя цена компьютеров по центру затрат и частоте ЦП
Вкладка **Общее**		
Тип представления	seType	Горизонтальная гистограмма
Key	bLegend	Отметьте этот флажок
Вкладка **Запрос**		
Таблица	TableName	Компьютеры (amComputer)
Группировать по	GroupBy	Portfolio.CostCenter.Title

Поле или ссылка (метка)	Поле или ссылка (имя SQL)	Значение
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	lCPUSpeedMHz
Использовать выражение	bUseAgregExpr	Отметьте этот флажок
Выражение	AqlAgregExpr	Avg(Portfolio.mAvgPrice)

4. Подтвердите создание статистики с помощью кнопки **Создать** (клиент Windows) или **Сохранить** (веб-клиент).

5. Если используется клиент Windows, диаграмму можно просмотреть с помощью следующих действий.

   a. Нажмите кнопку **Проверка**.

   b. Нажмите **ESC**, чтобы закрыть диаграмму.

Пример 2. Разбивка по половому признаку

Необходимо увидеть разбивку сотрудников компании по половому признаку. Для выполнения группировки достаточно использования одного критерия (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. Подтвердите создание фильтра с помощью кнопки **Создать** (клиент Windows) или **Сохранить** (веб-клиент).

4. Если используется клиент Windows, закройте все окна после завершения.

Теперь создайте статистику:

1. Отобразите статистику (ссылка **Администрирование/ Отчеты/ 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 по половому признаку
Вкладка **Общее**		
Key	bLegend	Отметьте этот флажок
Тип представления	seType	Сектора
Объемное отображение	b3D	Отметьте этот флажок
Вкладка **Запрос**		
Таблица	TableName	Сотрудники и подразделения (amEmplDept)
Filter	Filter	Только сотрудники
Группировать по	GroupBy	MrMrs

4. Подтвердите создание статистики с помощью кнопки **Создать** (клиент Windows) или **Сохранить** (веб-клиент).

5. Если используется клиент Windows, диаграмму можно просмотреть с помощью следующих действий.

   a. Нажмите кнопку **Проверка**.

   b. Нажмите **ESC**, чтобы закрыть диаграмму.

6. Закройте все окна после завершения.

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/ Отчеты/ 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 местоположению
Вкладка **Общее**		
Тип представления	seType	Горизонтальная гистограмма
Key	bLegend	Отметьте этот флажок
Вкладка **Запрос**		
Таблица	TableName	Сотрудники и подразделения (amEmplDept)
Filter	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	Отметьте этот флажок

4. Подтвердите создание статистики с помощью кнопки **Создать** (клиент Windows) или **Сохранить** (веб-клиент).

5. Если используется клиент Windows, диаграмму можно просмотреть с помощью следующих действий.

   a. Нажмите кнопку **Проверка**.

   b. Нажмите **ESC**, чтобы закрыть диаграмму.

6. Закройте все окна после завершения.

Пример 4. Разбивка сотрудников по времени работы в компании

В данном примере требуется разбивка данных по трудовому стажу. Как и в предыдущем примере, при необходимости выполняется фильтрация и группировка сотрудников по трудовому стажу. Трудовой стаж не является значением, хранящимся в базу данных. Но в таблице **Подразделения и сотрудники** имеется поле с указанием даты, когда сотрудник начал работу в компании (**Дата приема на работу**). Это значение можно использовать для расчета трудового стажа сотрудника. При этом используется вычисляемое поле, значение которого затем применяется для группировки сведений.

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. Подтвердите создание с помощью кнопки **Создать** (клиент Windows) или **Сохранить** (веб-клиент).

4. закройте все окна.

Теперь создайте статистику:

1. Отобразите статистику (ссылка **Администрирование/ Отчеты/ 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 работы в компании
Вкладка **Общее**		
Key	bLegend	Отметьте этот флажок
Тип представления	seType	Кривые
Вкладка **Запрос**		
Таблица	TableName	Сотрудники и подразделения (amEmplDept)
Filter	Filter	Только сотрудники
Группировать по	GroupBy	Выберите вычисляемое поле **Приоритет**, созданное ранее

4. Подтвердите создание с помощью кнопки **Создать** (клиент Windows) или **Сохранить** (веб-клиент).

5. Если используется клиент Windows, диаграмму можно просмотреть с помощью следующих действий.

   a. Нажмите кнопку **Проверка**.

   b. Нажмите **ESC**, чтобы закрыть диаграмму.

6. Закройте все окна после завершения.

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/ Отчеты/ 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 '/F
rance/%'" 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 создания этого запроса будет использоваться параметр **Скрипт** (установите соответствующий флажок). Данный параметр позволяет написать скрипт, создающий данные статистики. API-интерфейс **XmlAttribute** используется для преобразования предопределенных XML-сущностей (см. руководство **Programmer's Reference**).

1. Отобразите статистику (ссылка **Администрирование/ Отчеты/ 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	Количество сотрудников в стране
Скрипт	bScript	Отметьте этот флажок

Поле или ссылка (метка)	Поле или ссылка (имя SQL)	Значение
Вкладка **Общее**		
Тип представления	seType	Вертикальная гистограмма
Key	bLegend	Отметьте 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. Подтвердите создание с помощью кнопки **Создать** (клиент Windows) или **Сохранить** (веб-клиент).

5. Если используется клиент Windows, диаграмму можно просмотреть с помощью следующих действий.

   a. Нажмите кнопку **Проверка**.

   b. Нажмите **ESC**, чтобы закрыть диаграмму.

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 li
ke '/Germany/%' and MrMrs='Mr'" value="9"/>
  <subgroup Name="England" filtertable="amEmplDept" filter="Location.FullName li
ke '/England/%' and MrMrs='Mr'" value="10"/>
</group>
<group Name="Mrs.">
  <subgroup Name="France" filtertable="amEmplDept" filter="Location.FullName lik
e '/France/%' and MrMrs='Mrs'" value="3"/>
  <subgroup Name="Germany" filtertable="amEmplDept" filter="Location.FullName li
ke '/Germany/%' and MrMrs='Mrs'" value="7"/>
  <subgroup Name="England" filtertable="amEmplDept" filter="Location.FullName li
ke '/England/%' and MrMrs='Mrs'" value="7"/>
</group>
<group Name="Miss">
  <subgroup Name="France" filtertable="amEmplDept" filter="Location.FullName lik
e '/France/%' and MrMrs='Miss'" value="2"/>
  <subgroup Name="Germany" filtertable="amEmplDept" filter="Location.FullName li
ke '/Germany/%' and MrMrs='Miss'" value="4"/>
  <subgroup Name="England" filtertable="amEmplDept" filter="Location.FullName li
ke '/England/%' and MrMrs='Miss'" value="1"/>
</group>
```

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

1. Отобразите статистику (ссылка **Администрирование/ Отчеты/ 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	Количество сотрудников по типу обращения и стране
Скрипт	bScript	Отметьте этот флажок
Вкладка **Общее**		

Поле или ссылка (метка)	Поле или ссылка (имя SQL)	Значение
Key	bLegend	Отметьте этот флажок
Тип представления	seType	Вертикальная гистограмма
Вкладка **Скрипт**		

Поле или ссылка (метка)	Поле или ссылка (имя SQL)	Значение
Скрипт данных	Скрипт	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>"

Поле или ссылка (метка)	Поле или ссылка (имя SQL)	Значение
		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>"
		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. Подтвердите создание с помощью кнопки **Создать** (клиент Windows) или **Сохранить** (веб-клиент).

5. Если используется клиент Windows, диаграмму можно просмотреть с помощью следующих действий.

   a. Нажмите кнопку **Проверка**.

   b. Нажмите **ESC**, чтобы закрыть диаграмму.

6. Закройте все окна после завершения.

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, использующаяся 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, использующимся для основной группы.

Серии 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
const 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 DISTINCT Model.Name FROM amTraining WHERE M
odel.Name <> '''' ,"¤","|")
strSeries = "Number of training sessions" & "|" & "Number of training days" & "|
" & "Cost"
```

Примечание. Можно определять:

- **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 (на этапе 2) необходимо добавить основной фильтр к фильтрам, используемым для получения значений, а также добавить основное выражение к каждому выражению подгруппы.

Данные функции не будут использоваться в этом примере, так как они не требуются. Обратитесь к базе данных для получения примеров использования данных функций.

Примечание. 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 lColumnValue 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 lSerieValue 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(lTrainingId) 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
lSerieValue = 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) & " " & Xml
Attribute("filtertable", strTable) & " " & XmlAttribute("filter", strFilter) & "
" & XmlAttribute("value", lSerieValue) & "/> "
lColumnValue = lColumnValue + lSerieValue
next
RetVal = RetVal & "<group " & XmlAttribute("name", strColumn) & " " & XmlAttribu
te("value", lColumnValue) & "> "
RetVal = RetVal & subgroup
RetVal = RetVal & "</group> "
next
```

Этап 4: просмотр статистики

Если используется клиент Windows, нажмите кнопку **Проверка...** для отображения статистики.

Определение настраиваемых цветов для статистики в виде диаграммы

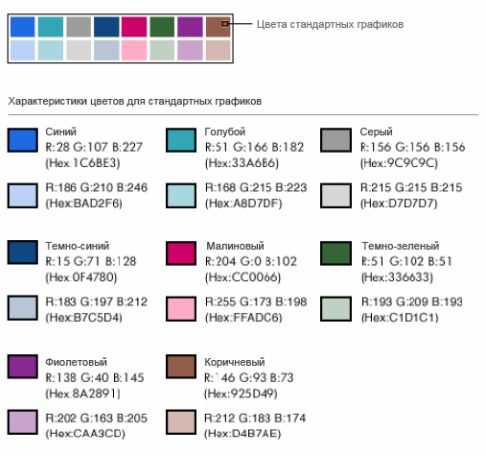
Можно задать настраиваемые цвета, используемые для запроса со скриптами, который возвращает результаты статистики в виде диаграммы для набора подгрупп или серий. (Имеет смысл определять цвета только в запросах, включающих несколько подгрупп).

Чтобы определить цвет для серии, используйте параметр **color=** с инструкцией определения подгруппы. Пример.

```
  <subgroup Name="Germany" filtertable="amEmplDept" filter="Location.FullName li
ke '/Germany/%' and MrMrs='Mr'" value="9" color="#cc0066"/>
```

> **Примечание.** 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) :



В этом разделе содержатся следующие сведения.

Обработка ошибок 143
Практический пример 144

Обработка ошибок

Для разрешения отклонений в определении цветов используются следующие правила.

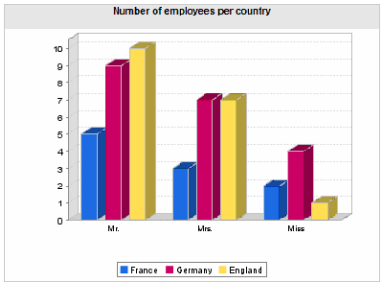
- Если параметр **color=** не определен для одной или нескольких подгрупп в запросе, к данным подгруппам будут применяться цвета по умолчанию.
- Если параметр **color=** использует нераспознанное значение, будет применяться цвет по умолчанию.
- Цвета для подгруппы можно определить только один раз. (Любая 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.



Глава 7: Действия

В этой главе объясняется, как определять действия с помощью Asset Manager.

Используйте навигационное меню **Администрирование/ Действия** для создания действий.

Если используется клиент Windows, действия можно выполнять с помощью пункта меню **Инструменты/ Действия** или контекстного списка "Действия" на панели инструментов.

Если используется веб-клиент, действия запускаются из раскрывающегося меню **Действие** на экране списка таблиц.

Глава включает следующее:

Рекомендации по разработке 145

Определение действия. 146

Создание действия 148

Примеры действий 157

Использование переменных 159

Действия, связанные со службой поддержки 160

Тестирование действия 161

Выполнение действия 162

Связывание действия с кнопкой на экране 165

Переопределение действия 165

Рекомендации по разработке

Глава включает следующее:

Использование тестовой функциональной области для настройки действий 145

Пометка веб-служб 146

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 (в том числе 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**).
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 выполнено развертывание Asset Manager Web.

См. руководство **Tailoring**, глава **Customizing the database**, раздел **Development best practices/ Tag the Web services**.

Определение действия.

Действие позволяет полностью или частично автоматизировать задачи, выполняемые с базой данных Asset Manager.

Имеются несколько типов действий:

- Исполняемый
- DDE
- Обмен сообщениями
- Скрипт: изменение объекта в базе данных Asset Manager

- Мастер
- Печать
- Распространение ПО
- Действие
- Инфопанель

Действия должны определяться первыми, чтобы выполняться при их выборе в списке.

Примечание. Можно определить область для действия, а также распределять по категориям области в соответствии с их функциями с помощью функциональных областей.

Функциональная область

Приложение 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/ Управление модулями** в клиенте Windows.

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.

Примечание. 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. Щелкните **Создать**.
3. Назначьте **Имя** своей функциональной области. Данное имя - имя, которое будет появляться в навигационной панели клиентов Windows и веб-клиентов. По умолчанию приложение Asset Manager назначает данной функциональной области **Имя SQL**. При необходимости это значение можно изменить.

4. Можно также выбрать **Родительскую область** для функционального домена, если необходимо.

5. Подтвердите создание с помощью кнопки **Создать** (клиент Windows) или **Сохранить** (веб-клиент).

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 148
Общий процесс создания действия 152
Заполнение вкладки DDE 153
Заполнение вкладки "Обмен сообщениями" 154

Типы действий

Приложение Asset Manager позволяет определять несколько типов действий:

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

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.

Действием **DDE** отправляется запрос DDE к приложению сервера DDE (или совместимому с DDE приложению), способному обрабатывать запросы DDE.

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 стандартного сообщения VIM, MAPI или SMTP приложение Asset Manager использует:

- поля **Учетная запись** (имя SQL: MailLogin) и **Пароль** (имя SQL: MailPassword) вкладки **Обмен сообщениями** сведений о сотруднике, который выполнил вход в базу данных Asset Manager (таблица "Сотрудники и подразделения") для идентификации отправителя сообщения;
- поле **Электронная почта** (имя SQL: EMail) на вкладке **Общее** сведений о сотруднике, чтобы идентифицировать получателя сообщения.

Чтобы отправить сообщение с помощью внутренней системы обмена сообщениями, приложение Asset Manager использует поля **Имя для входа** и **Пароль** на вкладке **Профиль** сведений отправителя и получателя.

Примечание. Адрес пользователя Asset Manager во внутренней системе обмена сообщениями такой же, что и его **Имя для входа**.

Внимание! Администратор должен создать пользователя с именем Admin и заполнить поля **Учетная запись**, **Эл. 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, поле или ссылке.

Операции, выполняемые действием данного типа, описываются скриптом на языке Basic; это позволяет использовать сложные функции, подобные функциям в Asset Manager API.

Примечание. 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, **AmSetFieldStrValue()** 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;

Примечание. 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 **Оборудование** (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**.

Дополнительные сведения о распространении ПО: см. руководство **Интеграция с инструментами для распространения ПО и управления конфигурацией**.

Для выполнения действия типа **Распределение ПО**:

1. Отобразите компьютеры (ссылка **Управление портфелем/ Конфигурации активов/ ИТ-оборудование/ ИТ-оборудование** в панели навигации).
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 следующее действие.
   - Windows-клиент: меню **Инструменты/ Действия/ <имя действия>**
   - Веб-клиент: выберите <Имя действия> из раскрывающегося списка **Действие** на экране оборудования.

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:

1. Создайте действие типа **Распределение ПО**.
   Данным действием создается запланированная задача для нескольких компьютеров.

2. 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.

Общий процесс создания действия

1. Выберите навигационное меню **Администрирование/ Действия**.

2. Щелкните **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).

   Примечание. Если поле **Имя SQL** не заполнено, оно автоматически заполняется

приложением.

6. Заполнит 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 **Создать** (Windows-клиент) или **Сохранить** (веб-клиент).

Примечание. 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 для поля **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

для исполняемых файлов.

- 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.

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, приложение 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) на вкладке **Общее** и относящемуся к сотруднику, открывшему базу данных Asset Manager (в таблице "Сотрудники и подразделения").

Примечание. Нельзя получить подтверждение для сообщения, посланного через внутреннюю систему обмена сообщениями Asset Manager или через систему обмена сообщениями MAPI или SMTP.

Отправка сообщения в формате HTML

Примечание. Формат HTML применим только для сообщений, отправляемых с помощью протокола SMTP.

Чтобы отправить сообщение в формате HTML, выберите параметр **Формат HTML** (имя SQL: bHTML) и введите содержимое сообщения с тегами HTML.

Указание адреса

Есть несколько способов указания адреса:

Адрес в формате <механизм обмена сообщениями>:<адрес для отправки сообщения>

Можно использовать следующие <механизмы обмена сообщениями>.

- 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 формате <имя для входа в Asset Manager>

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 в Asset Manager: "Cathy", "Gerald" and "Philip".
2. Поля **Эл. почта** показывают MAPI:CathyBernard@taltek.com для Cathy и VIM:Gerald Colombo/Taltek для Gerald. Поле **Эл. почта** для "Philip" пусто.
3. Если у отправителя имеется учетная запись MAPI, сообщение будет отправлено Cathy через интерфейс MAPI, а двум другим получателям — через внутреннюю систему обмена сообщениями Asset Manager.
4. Если у отправителя имеется учетная запись VIM, сообщение будет отправлено Gerald через интерфейс VIM, а двум другим 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

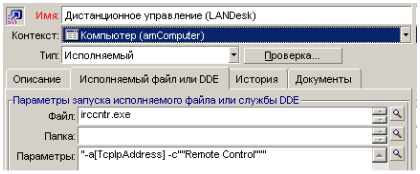
В данном разделе приведены следующие примеры действий Asset Manager:

Пример действия исполняемого типа 157
Пример действия типа DDE 157
Пример действия типа "Обмен сообщениями" 157
Пример действия типа "Скрипт" 158

Пример действия исполняемого типа

На следующем экране описывается неконтекстное действие, с помощью которого запускается приложение Asset Manager Automated Process Manager и выполняется подключение к базе данных **amdemo**:

Действие исполняемого типа - окно сведений



Пример действия типа DDE

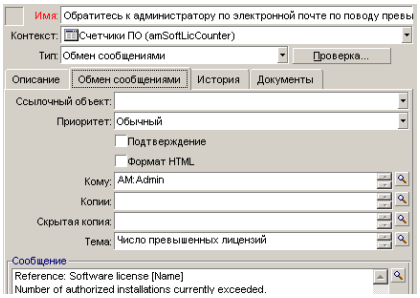
Действия типа **DDE** применяются во многих случаях:

- 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 записи;
- автоматическая отправка запроса наряда на работу;

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:

Действие типа "Обмен сообщениями" с ссылочным объектом - окно сведений



Пример действия типа "Скрипт"

Создание действия типа **Скрипт** заключается в написании скрипта на языке Basic, которым изменяется база данных Asset Manager.

> **Примечание.** 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 навигационное меню **Администрирование/ Действия** и нажмите кнопку **Создать** на экране действий.

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 **Создать** (Windows-клиент) или **Сохранить** (веб-клиент).

3. Если используется клиент Windows, можно нажать значок на вкладке **Скрипт** для отображения окна составителя скриптов.

Для программируемой функции с именем `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 подключаться	bIsCnxClient	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, "bIsCnxClient", 1)
  AmInsertRecord(lrec)
```

Примечание. 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 [bIsCnxClient]=1
```

Примечание. Чтобы выполнить данное действие, пользователь должен открыть таблицу "Виды" и щелкнуть **Новый**. После выполнения скрипта пользователь должен также щелкнуть **Создать** (клиент Windows) или **Сохранить** (веб-клиент), 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:

```
RetVal=12001
```

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]**.

Совет. При использовании клиента Windows щелкните значок увеличения справа от заполняемого поля для получения справки по вводу этих переменных.

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! Для функционирования переменных в поле **Контекст** сведений о действии

должна отображаться таблица в Asset Manager, а перед выполнением действия необходимо выбрать запись в списке записей таблицы.

Действия, связанные со службой поддержки

Несколько основных операций в службе поддержки могут теперь автоматически инициировать действия (открытие и закрытие записи, основные изменения и так далее).

Например, может отправляться сообщение вызывающему при открытии записи.

В этом разделе содержатся следующие сведения.

Действия, определяемые схемами эскалации 160

Действия, определенные в бланках приостановки 161

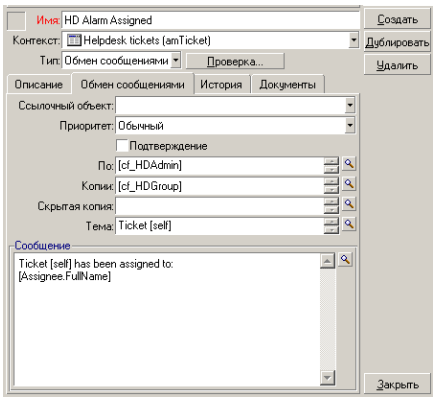
Действия, определяемые схемами эскалации

Схемой эскалации, связанной с записью, определяется большинство автоматических действий, связанных с данной записью (**Эскалация** (поле SQL: EscalScheme) на вкладке **Отслеживание** записи).

Поэтому необходимо определить для конкретных случаев действия, которые должны инициироваться в соответствии с потребностями организации.

Ниже приведен пример действия:

Действие типа "Обмен сообщениями", связанное со схемой эскалации - окно сведений



Также, как и для всех действий, можно вставлять переменные, связанные с содержимым данного поля. Можно также использовать вычисляемые поля для назначения членов группы службы поддержки:

Назначение с помощью вычисляемых полей членов группы службы поддержки

Переменная	Описание
cf_HDAdmin	Назначение администраторов службы поддержки
cf_HDGroup	Назначение лица, ответственного за группу службы поддержки, связанную с записью.
cf_HDInCharg	Назначение ответственного за запись (назначенного).
cf_HDContact	Назначение контакта, связанного с записью службы поддержки.
[self]	Если поле **Контекст** (имя SQL: ContextTable) не заполнено, появляется имя действия. Если в поле **Контекст** указывается имя таблицы Asset Manager, возвращается строка описания записи, выбранной в таблице.

Существует иерархия таких получателей: Администратор Asset Manager в данной иерархии находится выше администраторов службы поддержки, которые, в свою очередь, выше ответственных за запись, а те, в свою очередь, выше контактов для записи.

Если для какой-либо записи одного из данных получателей не существует, то сообщение отправляется получателям, которые выше на одну ступень в иерархии, и так далее.

Примечание. Действия, определенные во вкладках**Переходы** и **Назначение** в сведениях о схеме эскалации, инициируются агентом Asset Manager, тогда как действия, определенные на вкладке **Оповещения** инициируются приложением Asset Manager Automated Process Manager.

Действия, определенные в бланках приостановки

При приостановке записи определяются:

- максимальная продолжительность приостановки в поле **Общее время приостановки** (имя SQL: tsTotalSuspTime);
- Действие, которое автоматически инициируется приложением Asset Manager Automated Process Manager, если запись остается приостановленной по достижении предельного срока приостановки.

Тестирование действия

Примечание. В настоящее время эта функциональность в веб-клиенте недоступна.

Для тестирования действия при его создании используется кнопка **Тест** в верхнем правом углу сведений о тестируемом действии.

В приложении Asset Manager отображается окно предварительного просмотра, в котором можно выбрать контекст для предварительного просмотра действия. Данное окно содержит три кнопки: **Вычислить Выполнить** и **Отмена**.

В этом разделе содержатся следующие сведения.

Кнопка Вычислить 162
Кнопка Выполнить 162

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

В этом разделе содержатся следующие сведения.

Windows-клиент 162
Веб-клиент 163
Множественный выбор в списках 163
Действие типа "Мастер" 163
Действия исполняемого типа 164

Windows-клиент

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 как минимум одно действие, то в контекстном меню будет показан пункт **Действия**.

Веб-клиент

Действие можно выполнить следующим образом.

- С помощью контекстного меню (доступного с помощью щелчка правой кнопкой мыши) на экране списка/сведений. Если для открытой таблицы/записи доступно хотя бы одно действие, в контекстном меню будет отображен пункт **Действия**.
- Из раскрывающегося списка **Действия** на экране списка/сведений.

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
Запускается на рабочей станции клиента Windows веб-браузер по умолчанию и подключается по адресу URL, указанному в поле **Файл** (ActionFile).	Отображается страница, соответствующая 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
Запускается на рабочей станции Windows-клиента Проводник Windows и подключается к узлу ftp, указанному в поле **Файл** (ActionFile).	Отображается узел ftp, соответствующий URL-адресу, указанному в поле **Файл** (ActionFile) рабочей области веб-клиента.

Если поле Файл в сведениях о действии начинается с mailto

Запускается приложение обмена сообщениями по умолчанию на рабочей станции клиента Windows или веб-клиента и создается новое сообщение с помощью параметров, указанных после **mailto:** в поле **Файл** (ActionFile).

Если поле Файл 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 выполнение 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**.

Переопределение действия

Функция "Переопределение" позволяет переопределять и настраивать системные действия (скрипты и мастера) в приложении Asset Manager.

Например, если требуется изменить функциональность действия **Закрытие записи**, можно просто скопировать существующее системное действие **Закрытие записи** в новое действие **Закрытие моей записи**, изменить скопированное действие, а затем настроить

переопределение действия по умолчанию **Закрытие записи** действием **Закрытие моей записи**. После этого при каждом запуске действия **Закрытие записи** вместо действия по умолчанию будет выполняться настроенное действие **Закрытие моей записи**.

Эта функция полностью прозрачна для пользователей. Если системное действие переопределяется настраиваемым действием, имя действия в клиентском меню не изменяется, хотя фактически будет выполняться настраиваемое действие.

Примечание. Функция "Переопределение" доступна только в клиенте Windows.

Чтобы использовать эту функцию, выполните следующие действия.

1. В меню **Инструменты** выберите **Действия** и щелкните **Изменить**. Откроется окно **Сведения о действии**.
2. Выберите действие, поле со списком **Переопределено** будет доступно в правой области.
3. Откройте поле со списком **Определено**, в котором будут доступны все настраиваемые действия, которые можно использовать для переопределения выбранного системного действия. Выберите нужное действие и сохраните изменение.

Примечание. Настраиваемое действие не должно содержать контекстной таблицы или должно содержать такую же контекстную таблицу, как системное действие. В противном случае возникнет ошибка. Кроме того, после переопределения действия область настраиваемых действий изменить нельзя.

Известная проблема

При попытке открыть устаревшую базу данных с помощью Asset Manager 9.40 в клиенте генерируется следующее сообщение об ошибке:

"Неизвестное поле 'lOverriddenById' в таблице 'Actions (amAction)'."

Если затем открыть экран **amAction** и запустить действие, произойдет сбой клиента.

Данная проблема возникает из-за того, что функция переопределения пытается получить доступ к полю, которое отсутствует в прежних версиях Asset Manager.

Глава 8: Рабочий процесс

Примечание. Эта глава полностью применима к клиенту Windows. Однако некоторые из функциональных возможностей в настоящее время недоступны в веб-клиенте.

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;

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.

Рабочий процесс - упрощенная схема

Глава включает следующее:

Определения 168
Общий обзор 169
Реализация рабочего процесса 170
Использование графического редактора рабочего процесса 171
Пример рабочего процесса, используемого при утверждении заявки 173
Контекст рабочего процесса 185
Роли в рабочем процессе 187
Операции рабочих процессов 188
Задачи рабочих процессов 193
События рабочих процессов 197
Переходы рабочих процессов 209
Оповещения рабочих процессов и лимиты времени 210
Группы выполнения рабочих процессов 212
Отслеживание рабочего процесса 212
Удаление экземпляров завершенных рабочих процессов 214
Техническая информация: модель данных 220

Определения

В данном разделе даются определения нескольких ключевых терминов, используемых в рабочем процессе:

Экземпляр 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 приложением

Asset Manager Automated Process 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 операций - пользователи, ответственные за выполнение задач, явившихся результатом операций рабочих процессов.

Группа выполнения 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 приложении Asset Manager является определение схем рабочих процессов с помощью графического редактора через пункт меню навигации **Администрирование/ Рабочие процессы/ Схемы рабочего процесса**. Схемами рабочих процессов определяются операции, события и переходы. Они могут ссылаться на операции и сотрудников (назначенных рабочих процессов) в приложении Asset Manager.

Схемы рабочих процессов интерпретируются механизмами рабочих процессов. Рассматриваемые механизмы рабочих процессов выполняются либо приложением Asset Manager Automated Process Manager, либо агентами Asset Manager.

Реагируя на события, механизмы рабочих потоков инициируют и отслеживают экземпляры рабочих процессов:

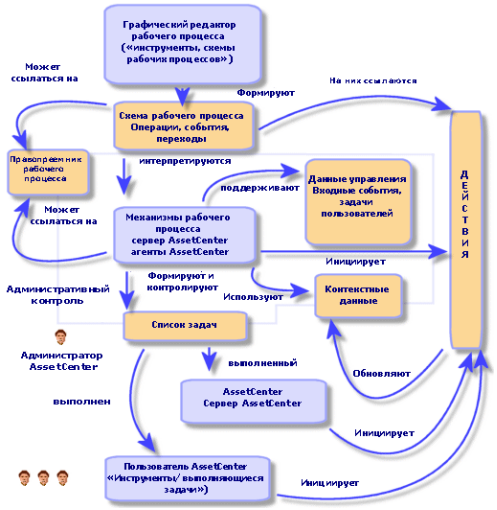
- механизмы рабочего процесса создают задачи для выполнения;
- ими отслеживаются такие задачи и события, инициирующие операции;

- 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 приводится обзор того, как реализуется рабочий процесс в Asset Manager:

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:
   - Роли в рабочем процессе.
   - Действия.
3. Создайте схемы рабочего процесса, для которых определите:
   - Операции, события и переходы.
   - Оповещения.

4. Определите соответствующие группы выполнения рабочих процессов. Свяжите каждую схему рабочего процесса с группой выполнения рабочего процесса.

5. Запустите приложение 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, а также параметры отслеживания.

Использование графического редактора рабочего процесса

Для доступа к схемам рабочих процессов используется пункт навигационного меню **Администрирование/Рабочие процессы/Схемы рабочего процесса**. Данный пункт меню зарезервирован для администраторов 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 171
События 172
Переходы 172
Прочая функциональность 173

Операции

Чтобы создать операцию:

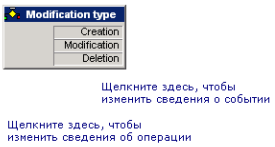
- Нажмите кнопку , а затем щелкните в графической зоне. Отобразятся сведения об операции.

Чтобы удалить операцию:

- либо выберите операцию, щелкнув ее, а затем нажмите клавишу Delete;
- либо выберите операцию и нажмите кнопку .

Чтобы изменить сведения об операции:

- Дважды щелкните операцию.



События

События, содержащиеся в операциях, бывают двух типов.

Событие типа "База данных" или "Периодическое"

Чтобы добавить исходящее событие к операции:

- Выберите операцию, а затем нажмите кнопку .

Чтобы удалить из операции исходящее событие типа **База данных** или **Периодическое**:

- выберите событие, затем нажмите клавишу Delete;
- или выберите событие и щелкните кнопку .

Чтобы изменить сведения о событии типа **База данных** или **Периодическое**:

- Дважды щелкните событие.

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 щелкните кнопку .

Чтобы изменить сведения о переходе:

- Дважды щелкните переход, чтобы отобразить сведения о нем.

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.

Предназначение 173
Предпосылки 174
Создание операций 177
Настройка событий, создаваемых одновременно с операциями 181
Создание начального события 182
Создание переходов 182
Пример активации экземпляра рабочего процесса 183

Предназначение

Предназначение данной схемы рабочего процесса - автоматизация обработки заявки на поставку в соответствии со следующим:

Рабочий процесс - проверка заявки

Схема рабочего процесса состоит из следующих шагов:

1. Экземпляр рабочего процесса запускается, как только имеется заявка на поставку для проверки (1), т.е. когда поле **Статус заявки** (имя SQL: seStatus) в сведениях заявки на поставку имеет значение **Ожидается утверждение**.

2. Сначала заявка проходит техническую проверку (2). Данный шаг заключается в представлении заявки ответственному подразделения 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 отправке заявки финансовому контролеру, г-ну Gerald Colombo. Ему также направляется уведомление по электронной почте, а также 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 утверждается, в сведениях этой заявки приложение Asset Manager устанавливает для поля **Статус заявки** значение **Проверено**.
   Если заявка на поставку отклоняется (6b), в сведениях заявки приложение Asset Manager устанавливает для поля **Статус заявки** значение **Отклонено**.

Предпосылки

Необходимо подключиться к базе данных с именем входа **Admin** и настроить систему обмена сообщениями (см. руководство **Администрирование**, раздел **Обмен сообщениями**).

Создание назначенных рабочих процессов

В данной схеме рабочего процесса участвуют следующие назначенные:

- ответственный за подразделение запрашивающего;
- Финансовый контролер компании, г-н 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 **Администрирование/ Рабочие процессы/ Роли в рабочем процессе**, а затем щелкните **Создать**.

Ответственный за подразделение запрашивающего

Данное лицо рассчитывается скриптом. Для его определения заполните экран сведений, как показано ниже.

- Назначение: Ответственный за подразделение запрашивающего
- Контекст: Заявки (amRequest)
- Тип: Рассчитанное лицо
- Скрипт: RetVal = [Requester.Parent.Supervisor]

Финансовый контролер

Данное лицо указано как г-н Gerald Colombo. Чтобы указать его в качестве назначенного, заполните экран сведений следующим образом.

- Назначение: Финансовый контролер
- Контекст: (нет таблиц)
- Тип: Назначенное лицо
- Назначенный: Colombo, Gerald

Создание действий

Схемами рабочих процессов вызываются многочисленные действия. Чтобы их создать, выберите пункт меню **Администрирование/Действия**.

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. Для его создания заполните экран сведений, как показано ниже.

- Имя: Заявка на техническую проверку
- Контекст: Задачи рабочего процесса (amWfWorkItem)
- Тип: Обмен сообщениями
- Вкладка "Обмен сообщениями":
  - Приоритет: Нормальное
  - Кому: [Assignee.EMail]
  - Тема: Утверждение заявки
  - Сообщение: Вам необходимо утвердить или отклонить заявку на поставку.

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. Для его создания заполните экран сведений, как показано ниже.

- Имя: Отправка напоминания о финансовой проверке
- Контекст: Задачи рабочего процесса (amWfWorkItem)
- Тип: Обмен сообщениями
- Вкладка "Обмен сообщениями":
  - Приоритет: Нормальное
  - Кому: [Assignee.EMail]
  - Тема: Утверждение заявки
  - Сообщение: Вы утверждаете заявку [ReqNumber]?

Проверка заявки на поставку

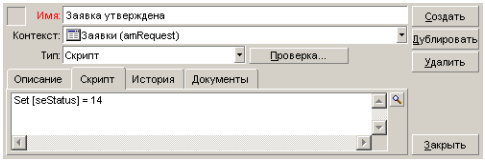
Данное действие используется на уровне операции **Заявка утверждена**, которая будет создана позже.

Оно устанавливает в сведениях о заявке для поля **Статус заявки** (имя SQL: seStatus) значение **Проверено**. Это действие имеет тип **Скрипт**.

Поле **Статус заявки** является системным подробным списком. Чтобы получить список его значений, отобразите справку по этому полю:

1. Переместите курсор на поле **Статус заявки** в сведениях заявки.
2. Установите фокус на этом поле и нажмите сочетание клавиш **Shit+F1**: Значение, отображаемое как **Проверено**, хранится в базе данных как **14**.

При использовании клиента Windows выполняется следующее действие.

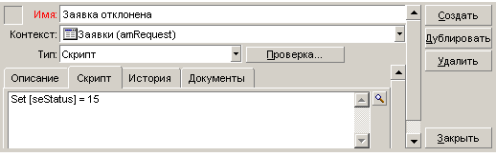


Отклонение заявки на поставку

Данное действие используется на уровне операции **Заявка отклонена**, которая будет создана позже.

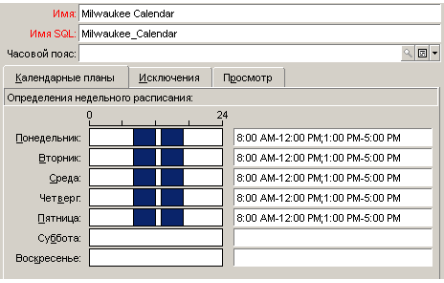
Оно аналогично действию **Заявка утверждена**, но поле **Статус заявки** (имя SQL: seStatus) на вкладке **Общее** в сведениях заявки должно иметь значение **Отклонено**.

При использовании клиента Windows выполняется следующее действие.



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:



Подготовка схемы рабочего процесса

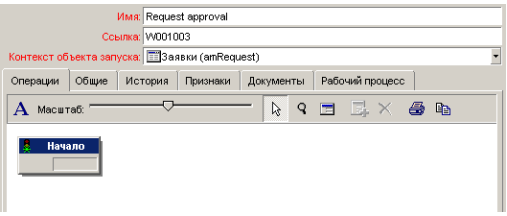
1. Щелкните пункт меню **Администрирование/ Рабочие процессы/ Схемы рабочего процесса**.

2. Щелкните **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 приложением Asset Manager в графическом редакторе на вкладке **Операции**.



Создание операций

Операции можно создавать графически на вкладке **Операции** сведений о рабочем процессе:

1. Щелкните .

Создание операции Техническая проверка

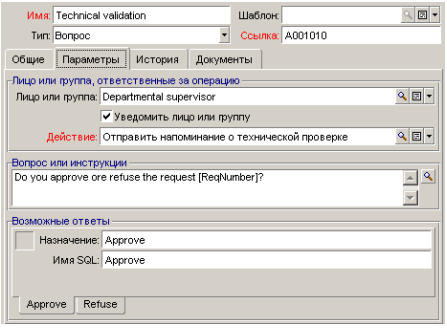
1. Назначьте имя операции **Техническая проверка**.

2. Так как операция представляет собой подачу заявки на утверждение ответственному подразделения, 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:



2. Укажите задаваемый вопрос:

   a. Текст вопроса ссылается на номер заявки на поставку.

   b. Возможны два ответа: отклонить или утвердить. Чтобы добавить подвкладку, описывающую ответ на вопрос, щелкните правой кнопкой мыши область метки подзакладок и выберите **Добавить связанную запись**.

3. Укажите, кому адресован запрос, в поле **Лицо или группа** (имя SQL: Assignee). В данном случае назначенным является ответственный за подразделение запрашивающего. Данное лицо было создано в таблице ролей в рабочем процессе в предыдущих шагах.

4. Чтобы назначенный был уведомлен о необходимости рассмотреть заявку:

   a. Установите для поля **Уведомить лицо или группу** (имя SQL: bNotifAssignee) значение **Да**.

   b. Укажите выполняемое действие: это действие **Заявка на техническую проверку**, созданное на промежуточном шаге. Данное действие автоматически инициируется при подаче заявки на поставку для технического утверждения.

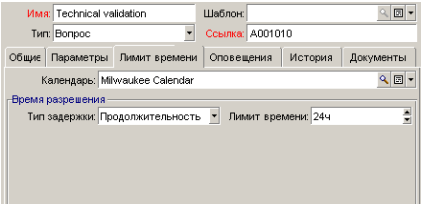
Определение лимита времени для выполнения операции Техническая проверка

Примечание. Перед установкой лимита времени необходимо установить параметр **Записать задачу в журнал** на вкладке **Общее**.

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.



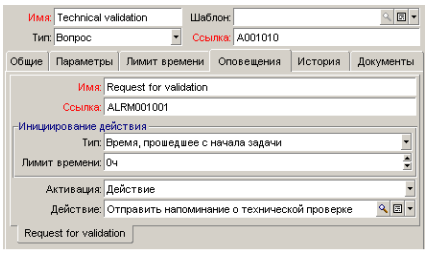
Определение оповещения для операции Техническая проверка

> **Примечание.** Перед установкой оповещения необходимо установить параметр **Записать задачу в журнал** на вкладке **Общее**.

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 техническую проверку**:

Можно определить дополнительные оповещения с помощью команды **Добавить связанную запись** в контекстном меню.



События

После создания операции приложение Asset Manager создает два системных события – **Утвердить** и **Отказать**, соответствующие двум возможным ответам на вопрос:

Когда происходят данные события, агент Asset Manager регистрирует их в таблице элементарных событий рабочих процессов (имя SQL: amWfOccurEvent).

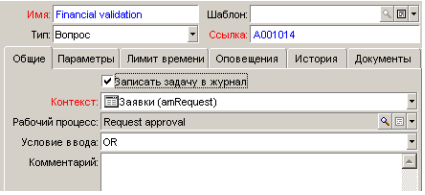
Ниже указаны операции, инициируемые непосредственно Asset Manager приложением Asset Manager Automated Process Manager в соответствии с конфигурацией событий рабочих процессов:

- Если в поле **Обработка** (имя SQL: seProcessingMode) установлены значения **Записать событие в журнал и обработать немедленно** или **Обработать событие немедленно без записи в журнал**, приложение Asset Manager инициирует следующее действие.

- Если в поле **Обработка** (имя SQL: seProcessingMode) установлено значение **Зарегистрировать событие и обработать сервером**), приложение 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.



- Назначенный отличается: в данном случае это финансовый контролер, г-н Gerald Colombo (назначенное лицо). Данное лицо создано на предыдущих этапах в таблице ролей рабочего процесса с ролью финансового контролера. Чтобы уведомить его, выберите действие **Отправка напоминания о финансовой проверке**, созданное ранее (поле **Действие**).

- Задержки и оповещения создаются, в основном, таким же образом, как для операции технической проверки.

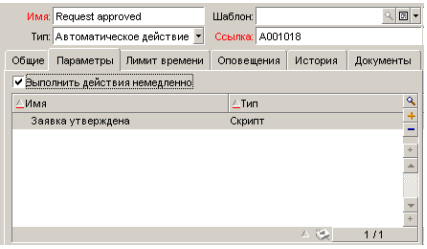
Создание операции Заявка утверждена

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.

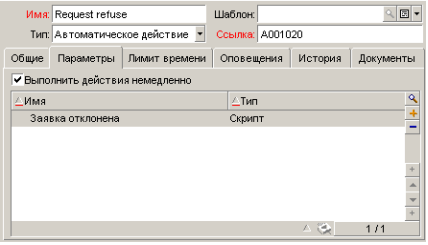
Данная операция, таким образом, является операцией типа **Автоматическое действие**: действие, которое должно выполняться - действие **Заявка утверждена**, созданное в предыдущем шаге.



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 в предыдущем шаге.



Настройка событий, создаваемых одновременно с операциями

Когда в течение предыдущих шагов были созданы операции, были также созданы следующие события.

- Операция **Техническая проверка**:
  - Событие **Утверждение**
  - Событие **Отказ**
- Операция **Финансовая проверка**:
  - Событие **Утверждение**
  - Событие **Отказ**
- Операция **Заявка утверждена**:
  - Событие **Выполнено**
- Операция **Заявка отклонена**:
  - Событие **Выполнено**

Примечание. В клиенте Windows можно выполнять только следующие действия.

Чтобы не зависеть от приложения Asset Manager Automated Process Manager, можно выбрать все события друг за другом и выполнить следующие действия.

1. Дважды щелкните имя события.
2. Выберите вкладку **Общее** на экране сведений события.

3. Выберите значение **Записать событие в журнал и обработать немедленно** для сохранения поля **Обработка** (имя SQL: seProcessingMode).

4. Нажмите кнопку **Изменить**.

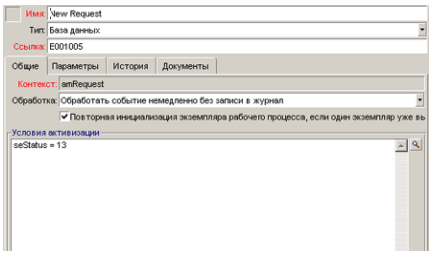
Создание начального события

События, которыми инициируется экземпляр рабочего процесса, связаны с операцией **Запуск**.

Чтобы определить событие **Запуск**, дважды щелкните пустую область события в операции **Запуск**.

Примечание. В клиенте Windows можно выполнять только следующие действия.

1. В данном случае экземпляр рабочего процесса инициируется, если для поля **Статус заявки** (имя SQL: seStatus) в сведениях заявки задано 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:

   a. Установите флажок **После обновления** (имя SQL: bUpdate).

   b. Укажите поле **seStatus** в поле **Отслеживаемые поля** (имя SQL: MonitFields).

Создание переходов

После создания операций их необходимо связать переходами.

Примечание. В клиенте Windows можно выполнять только следующие действия.

Для создания перехода:

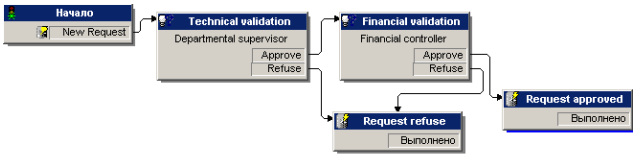
1. Щелкните начальное событие перехода.
2. Удерживая кнопку мыши нажатой, перетащите его в целевую операцию.

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:

Рабочий процесс - схема проверки заявки



Пример активации экземпляра рабочего процесса

Далее необходимо проверить, что схема рабочего процесса функционирует правильно.

Для этого необходимо выполнить следующее:

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**, относящееся 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 следующие операции, предоставьте Филиппу Чавезу (Philip Chavez) административные права для базы данных: Откройте вкладку **Профиль** в сведениях соответствующей записи и установите флажок **Административные права** (имя SQL: bAdminRight). Укажите **Имя для входа** (имя SQL: UserLogin) и пароль для Филиппа Чавеза (Philip Chavez).

6. Присвойте значение **AM:Chavez** полю **Эл. 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. Откройте пункт меню навигации **Жизненный цикл актива/ Жизненный цикл закупок/ Заявки / Заявки на поставку** в приложении Asset Manager.

3. Щелкните **Создать**.

4. В поле **Запрашивающий** (имя SQL: Requester) на вкладке **Общее** сведений о заявке выберите запись **Carpenter**.

5. Задайте для поля **Статус заявки** (имя SQL: serStatus) в сведениях заявки значение **Ожидается утверждение**.

6. Подтвердите заявку на поставку: происходит начальное событие, и приложение Asset Manager регистрирует их в таблице элементарных событий рабочих процессов (имя SQL: amWfOccurEvent).

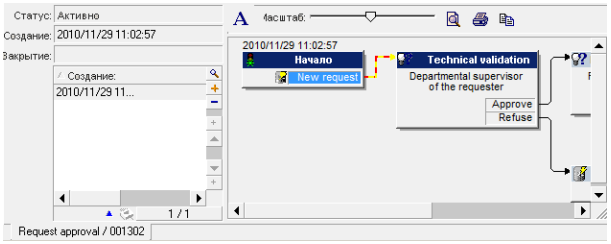
Контроль экземпляра

Чтобы проверить правильность функционирования экземпляра рабочего процесса, запустите приложение Asset Manager и подключитесь к демонстрационной базе данных с помощью имени входа Philip Chavez, ответственного подразделения для Jerome Carpenter.

Просмотр экземпляра рабочего процесса

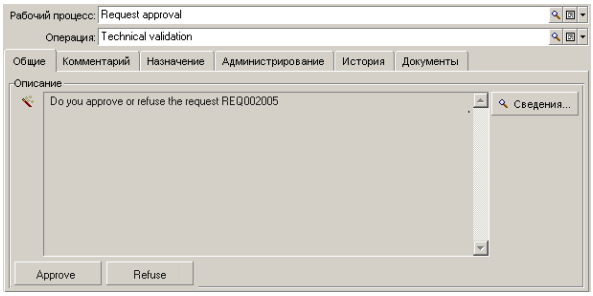
Отобразите сведения о созданной заявке на поставку: на вкладке **Рабочий процесс** перечисляются текущие экземпляры рабочих процессов для данной заявки. Каждый экземпляр описывается на отдельной подзакладке.

- Левая часть подвкладки 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 навигации **Жизненный цикл актива/ Управление инфраструктурой/ Мои задачи рабочего процесса**: ответственный подразделения Philip Chavez может таким образом просматривать сведения о назначенной задаче:



2. 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 приложением Asset Manager Automated Process Manager).
   Можно щелкнуть **Сведения**, чтобы получить доступ к сведениям о заявке, ставшей причиной данной задачи.
3. 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

В этом разделе содержатся следующие сведения.

Определение контекста экземпляра рабочего процесса 186
Объект, на который ссылается экземпляр рабочего процесса 186

Ограничение числа выполняющихся для данного объекта рабочих процессов 186

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: Экземпляр рабочего процесса инициируется при утверждении заявки на поставку. Создается заказ на поставку в соответствии с этой заявкой. Если заявка R1 утверждена, она является ссылочным объектом начального события. Ссылочный объект затем становится заказом на поставку PO1, т.е. заказом, генерируемым из заявки на поставку

Ограничение числа выполняющихся для данного объекта рабочих процессов

Параметр "Единственный активный экземпляр рабочего процесса для объекта" (имя SQL: bUniqueActive)

В приложении Asset Manager можно ограничить число одновременно выполняющихся экземпляров рабочего процесса для данного объекта с помощью параметра **Единственный активный экземпляр рабочего процесса для объекта** на вкладке **Общие** 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:

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="2">Параметр Один активный экземпляр рабочего процесса для объекта на вкладке Общее схемы рабочего процесса.</td></tr>
<tr><td>Проверено</td><td>Не проверено</td></tr>
<tr><td rowspan="2">Параметр Повторная инициализация экземпляра рабочего процесса, если один экземпляр уже выполняется на вкладке Общее исходящего события операции Запуск.</td><td>Проверено</td><td colspan="2">Если один экземпляр рабочего процесса для объекта уже выполняется, он будет остановлен и запущен новый экземпляр рабочего процесса.</td></tr>
<tr><td>Не проверено</td><td>Если один экземпляр рабочего процесса для объекта уже выполняется, событие будет проигнорировано (новый экземпляр рабочего процесса запущен не будет).</td><td>Создан новый экземпляр рабочего процесса.</td></tr>
</table>

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.

> **Примечание.** 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.

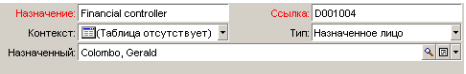
Тип роли в рабочем процессе

Есть несколько возможных типов ролей в рабочем процессе (поле **Тип** (имя SQL: seType) в сведениях о роли в рабочем процессе):

Назначенное лицо

В данном случае назначенный выбирается непосредственно в таблице подразделений и сотрудников.

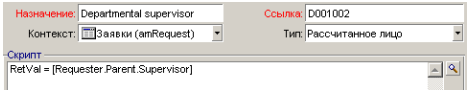
Пример:



Рассчитанное лицо

В данном случае назначенный присутствует в таблице подразделений и сотрудников, но рассчитывается с помощью скрипта.

Пример:



Группа

В этом случае поле **Лицо или группа** (имя SQL: Assignee) выбирается в таблице групп сотрудников (amEmplGroup).

Вычисляемая группа

В этом случае поле **Лицо или группа** (имя SQL: Assignee) рассчитывается с помощью скрипта из таблицы групп сотрудников (имя SQL: amEmplGroup).

Определение назначенного для операции

Поле **Лицо или группа** (поле SQL: Assignee) на вкладке **Параметры** позволяет определять назначенного для операций с типом **Вопрос** и **Действие пользователя**.

Операции рабочих процессов

Операции можно разделить на две группы:

- Требующие участия пользователя: операции типа **Вопрос** и **Действие пользователя** (поле **Тип** (имя SQL: seType) в верхней части 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:

Операция типа "Действие пользователя" 189
Операция типа "Вопрос" 190
Операция типа "Автоматическое действие" 191
Операции типа "Тест/скрипт" 191
Начальная операция 192
Шаблоны операций 192
Инициирование операций 193

Операция типа "Действие пользователя"

Данные операции требуют участия пользователя, называемого "назначенный". Назначенный показывается в поле **Лицо или группа** (имя SQL: Assignee).

Для его определения указываются:

- инструкции для выполнения;
- мастер для выполнения.

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 соответствующим образом выбрать параметр **Уведомить лицо или группу** (имя SQL: bNotifAssignee) на вкладке **Параметры**.

  > **Примечание.** Действие, уведомляющее назначенного, инициируется, как только будет создана выполняемая задача, т.е. сразу после активации перехода, инициирующего операцию.

  Назначенный использует пункт меню навигации **Жизненный цикл актива/ Управление инфраструктурой/ Мои задачи рабочего процесса** для доступа к сведениям о задачах, подлежащих выполнению.

> **Примечание.** 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 "Вопрос"

Данные операции требуют участия пользователя, указанного в поле **Лицо или группа** (имя SQL: Assignee).

Для определения операций типа **Вопрос** указываются:

- вопрос или инструкции;
- возможные ответы.

Примеры:

- В ходе процесса утверждения поставки ответственному подразделения 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

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 соответствующим образом заполнить поле **Лицо или группа** (имя SQL: bNotifAssignee) на вкладке **Параметры**.

   Примечание. Действие, уведомляющее назначенного, инициируется, как только будет создана выполняемая задача, т.е. сразу после активации перехода, инициирующего операцию.

   Назначенный использует пункт меню навигации **Жизненный цикл актива/ Управление инфраструктурой/ Мои задачи рабочего процесса** 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 или удалить ответ, щелкните правой кнопкой мыши область меток подвкладки и выберите в контекстном меню **Добавить связанную запись** или **Удалить ссылку**.

Примечание. Каждым ответом автоматически создается исходящее событие операции.

Операция типа "Автоматическое действие"

Данные операции выполняются автоматически приложением 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.

Укажите здесь список выполняемых действий.

> **Примечание.** 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 параметров, эти действия обрабатываются приложением Asset Manager Automated Process Manager или агентом Asset Manager.

- Если выбран параметр **Выполнить действия немедленно** (имя SQL: bExecImmediately), механизм рабочего процесса, который активизирует переход, инициирующий операцию, немедленно выполняет действия операции.
- В противном случае задачи будут выполнены приложением Asset Manager Automated Process Manager во время следующего цикла проверки.

Операции типа "Тест/скрипт"

Данные операции автоматически выполняются приложением 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.

Укажите:

- тестовый скрипт для выполнения.
- возможные результаты. Каждый результат описывается на отдельной подзакладке. Он идентифицируется по своему описанию и имени SQL. Чтобы добавить или удалить возможный результат, щелкните правой кнопкой мыши область меток подвкладки и выберите в контекстном меню **Добавить связанную запись** или **Удалить ссылку**.

Примечание. Внимание: имена SQL каждого результата должны соответствовать возвращаемым значениям тестового скрипта.

Примечание. Каждым результатом автоматически создается исходящее событие операции.

Выполнение

Механизм рабочего процесса, который активизирует переход, инициирующий операцию, немедленно выполняет действия операции. В зависимости от выбранных параметров, эти действия обрабатываются приложением Asset Manager Automated Process Manager или агентом 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, приложение Asset Manager Automated Process Manager или агент Asset Manager выполняет действия.
- В противном случае задачи будут выполнены приложением 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 **Администрирование/ Рабочие процессы/ Операции рабочего процесса**.

Примечание. Предостережение: чтобы информация, содержащаяся в сведениях о шаблоне операции (тип операции и т.д.), 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: seInCond) на вкладке **Общее** сведений об операции. Данное условие относится к переходам, которыми инициируется операция.

- Если есть только один переход, которым может инициироваться операция, должен активироваться только данный переход (приложением Asset Manager или Asset Manager Automated Process Manager), чтобы инициировалась данная операция.
- Если есть несколько переходов, которыми может инициироваться операция:
  - Если условие ввода операции - **AND**, для инициирования операции должны быть активизированы все переходы.
  - Если условие ввода операции - **OR**, для инициирования операции необходимо активизировать только один из переходов.

Примечание. Если условия ввода являются сложными (комбинациями условий **AND** и **OR**), для их выполнения можно создать последовательность промежуточных операций типа **Тест/скрипт**.

Задачи рабочих процессов

В данном разделе объясняется, как создаются и выполняются задачи рабочих процессов.

Создание задач 194
Операция типа "Автоматическое действие" или "Тест/скрипт" 194
Отображение списка всех задач рабочего процесса 194
Выполнение задачи пользователя 195
Назначение задач пользователей 195
Поручение задачи 196
Администрирование задачи рабочего процесса 197

Создание задач

Когда активизируется переход, инициирующий 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 приложение Asset Manager Automated Process Manager или агент Asset Manager.

В противном случае приложение Asset Manager Automated Process Manager через регулярные интервалы времени проверяет, есть ли необходимость в выполнении задач рабочего процесса, и выполняет их, если необходимо.

Частота, с которой приложением Asset Manager Automated Process Manager отслеживает функции рабочего процесса, определяется параметрами 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.

Выполнение задачи пользователя

Назначенное лицо получает доступ к своим задачам по ссылке **Жизненный цикл актива/ Управление инфраструктурой/ 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, назначенные могут получить доступ к порученным им задачам по ссылке **Жизненный цикл актива/ Управление инфраструктурой/ Мои задачи рабочего процесса** на навигационной панели. См. раздел Поручение задачи.

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

Для создания поручения используйте мастер **Создание/изменение поручений...** (sysCoreDelegation), доступ к которому можно получить в сведениях о сотрудниках с помощью ссылки **Делегировать** (клиент Windows) или с помощью пункта **Делегировать** из раскрывающегося меню **Действия...** (веб-клиент).

См. руководство **Общие таблицы**, глава **Подразделения и сотрудники**, раздел **Создать поручение**.

Примечание. Если для поручения не была указана дата окончания (dtEndDeleg), поручение является постоянным.

Примечание. Для активации поручения выберите параметр **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.

Совет. При использовании клиента Windows можно просматривать существующие поручения в сведениях о сотрудниках. Выберите вкладку **Поручения**, а затем подвкладки **Просмотр порученных поручений** и **Просмотр принятых поручений** . Для получения дополнительных сведений см. руководство **Интерфейс пользователя**, глава **Средство просмотра планировщика**.

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 **Жизненный цикл актива/ Управление инфраструктурой/ Мои задачи**

рабочего процесса на навигационной панели.

- Сотрудник Б видит все назначенные задачи рабочего процесса по ссылке **Жизненный цикл актива/ Управление инфраструктурой/ Мои задачи рабочего процесса** на навигационной панели, а также все задачи рабочего процесса, переданные ему/ей посредством поручения по ссылке **Жизненный цикл актива/ Управление инфраструктурой/ 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 **Жизненный цикл актива/ Управление инфраструктурой/ 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 рабочих процессов

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**.

В этом разделе содержатся следующие сведения.

Событие системы 198
Событие оповещения 198
Событие пользователя 198
Общие условия активации 201
Обработка событий 202
Применение: реализация синхронной схемы рабочего процесса 208
Заключительное событие 209

Событие системы

События **Системы** автоматически определяются в приложении 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 именами "Да" и "Нет".

Событие оповещения

События оповещения для операции создаются, когда определяются оповещения операций, которыми инициируются 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).

> **Примечание.** 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**

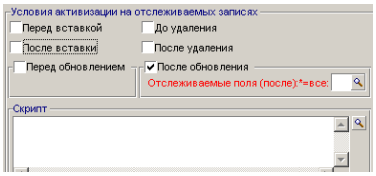
Событие типа "База данных"

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.

Вкладка параметров действия типа "База данных"



Параметры, которыми инициируется событие типа **База данных**, описаны на вкладке **Параметры** сведений о событии. Указывается следующая информация.

- Записи для мониторинга (Это могут быть записи в таблице, указанной в контексте, или связанные записи). Если записи для мониторинга – это записи, которые связаны с таблицей, определенной 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 функцию 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 функцию **После вставки**.

  - параметр **После вставки** (имя SQL: bInsert), позволяет активизировать переход после вставки новой записи;

    > **Внимание!** Эту функцию нельзя использовать, если схеме рабочего процесса необходимо выполнить действие типа скрипта для заполнения поля или ссылки при помощи функции **set()**.
    >
    > Вместо этого используйте функцию **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.
- Если выбран 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)**;

  Примечание. Если ввести * в поле **Отслеживаемые поля (до)**, то будут отслеживаться изменения в **любом** поле.

- Если выбран параметр **После обновления** (имя SQL: bUpdate), можно указать поля, которые при изменении должны учитываться в поле **Отслеживаемые поля (после)** (имя SQL: MonitFields). Чтобы указать несколько имен полей, используйте запятые в качестве разделителей. Если поле оставлено пустым, измененные поля не будут учитываться.

  Примечание. Если ввести * в поле **Отслеживаемые поля (после)**, то будут отслеживаться изменения в **любом** поле.

  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 клиентским компьютером 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.

Дополнительные сведения см. в разделе Обработка событий.

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.lUserId] <> lUserId 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 приложение Asset Manager Automated Process Manager инициализирует событие.

Частота, с которой приложение 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)

> **Примечание.** 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:

Ограничение числа выполняющихся для данного объекта рабочих процессов - различные возможные случаи

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="2">Параметр Один активный экземпляр рабочего процесса для объекта на вкладке Общие схемы рабочего процесса.</td></tr>
<tr><td>Проверено</td><td>Не проверено</td></tr>
<tr><td rowspan="2">Параметр Повторная инициализация экземпляра рабочего процесса, если один экземпляр уже выполняется на вкладке Общие исходящего события операции Запуск.</td><td>Проверено</td><td colspan="2">Если один экземпляр рабочего процесса для объекта уже выполняется, он будет остановлен и запущен новый экземпляр рабочего процесса.</td></tr>
<tr><td>Не проверено</td><td>Если один экземпляр рабочего процесса для объекта уже выполняется, событие будет проигнорировано (новый экземпляр рабочего процесса запущен не будет).</td><td>Создан новый экземпляр рабочего процесса.</td></tr>
</table>

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.

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**.	Приложение Asset Manager Automated Process Manager инициирует событие, если выполнены условия активации. Частота инициирования определяется на вкладке **Параметры** сведений о событии. Сразу после возникновения события приложение Asset Manager Automated Process Manager сохраняет его в таблице с SQL-именем **amWfOccurEvent**. Позже переход активируется приложением Asset Manager Automated Process Manager (частота, с которой Asset Manager Automated Process Manager отслеживает переходы, подлежащие активации, определяется параметрами в Asset Manager Automated Process Manager).	Приложение Asset Manager Automated Process Manager инициирует событие, если выполнены условия активации. Частота инициирования определяется на вкладке **Параметры** сведений о событии. Сразу после возникновения события приложение Asset Manager Automated Process Manager сохраняет его в таблице с SQL-именем **amWfOccurEvent** . Переход немедленно активируется приложением Asset Manager Automated Process Manager.	Приложение Asset Manager Automated Process Manager инициирует событие, если выполнены условия активации. Частота инициирования определяется на вкладке **Параметры** сведений о событии. Когда происходит событие, оно не записывается в журнал в таблицу с именем SQL **amWfOccurEvent** , но переход немедленно активируется приложением Asset Manager Automated Process Manager.

Различные способы, которыми может быть обработано событие, продолжение

	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 системное событие, инициированное приложением Asset Manager (в результате операции типа**Вопрос** или **Действие пользователя**, в результате операции **Автоматическое действие** или **Тест/скрипт**, выполняемой приложением Asset Manager)	Как только происходит событие, оно записывается в журнал в таблицу с именем SQL **amWfOccurEvent** клиентским компьютером Asset Manager. Позже переход активируется приложением Asset Manager Automated Process Manager (частота, с которой Asset Manager Automated Process Manager отслеживает переходы, подлежащие активации, определяется параметрами 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 активируется немедленно клиентским компьютером Asset Manager.

Различные способы, которыми может быть обработано событие, продолжение

	Записать событие в журнал и обработать на сервере	Записать событие в журнал и обработать немедленно	Обработать событие немедленно без записи в журнал
Событие системы, инициируемое приложением Asset Manager Automated Process Manager (результат операции типа **Тест / скрипт** или **Автоматическое действие**, выполняемой приложением Asset Manager Automated Process Manager), или событие оповещения об операции	Как только происходит событие, оно записывается в журнал в таблицу с именем SQL **amWfOccurEvent** приложением Asset Manager Automated Process Manager. Позже переход активируется приложением Asset Manager Automated Process Manager (частота, с которой Asset Manager Automated Process Manager отслеживает переходы, подлежащие активации, определяется параметрами в Asset Manager Automated Process Manager).	Как только происходит событие, оно записывается в журнал в таблицу с именем SQL **amWfOccurEvent** приложением Asset Manager Automated Process Manager. Переход немедленно активируется приложением Asset Manager Automated Process Manager.	Когда происходит событие, оно не записывается в журнал в таблицу с именем SQL **amWfOccurEvent**, но переход немедленно активируется приложением Asset Manager Automated Process Manager.

Различные способы, которыми может быть обработано событие

	Записать событие в журнал и обработать на сервере	Записать событие в журнал и обработать немедленно
Событие типа **Периодическое**.	Приложение Asset Manager Automated Process Manager инициирует событие, если выполнены условия активации. Частота инициирования определяется на вкладке **Параметры** сведений о событии. Сразу после возникновения события приложение Asset Manager Automated Process Manager сохраняет его в таблице с SQL-именем **amWfOccurEvent**. Переход немедленно активируется приложением Asset Manager Automated Process Manager.	Приложение Asset Manager Automated Process Manager инициирует событие, если выполнены условия активации. Частота инициирования определяется на вкладке **Параметры** сведений о событии. Сразу после возникновения события приложение Asset Manager Automated Process Manager не сохраняет его в таблице с SQL-именем **amWfOccurEvent**. Но переход немедленно активируется приложением Asset Manager Automated Process Manager.

Различные способы, которыми может быть обработано событие, продолжение

	Записать событие в журнал и обработать на сервере	Записать событие в журнал и обработать немедленно
Событие типа **База данных** или системное событие, инициированное приложением Asset Manager (в результате операции типа**Вопрос** или **Действие пользователя**, в результате операции **Автоматическое действие** или **Тест/ скрипт**, выполняемой приложением 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 активируется немедленно клиентским компьютером Asset Manager.
Событие системы, инициируемое приложением Asset Manager Automated Process Manager (результат операции типа **Тест / скрипт** или **Автоматическое действие**, выполняемой приложением Asset Manager Automated Process Manager), или событие оповещения об операции	Как только происходит событие, оно записывается в журнал в таблицу с именем SQL **amWfOccurEvent** приложением Asset Manager Automated Process Manager. Переход немедленно активируется приложением Asset Manager Automated Process Manager.	Когда происходит событие, оно не записывается в журнал в таблицу с именем SQL **amWfOccurEvent**, но переход немедленно активируется приложением 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:

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. Активизируется 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.

Пример схемы асинхронного рабочего процесса

Заключительное событие

Определение

Заключительным событием заканчивается экземпляр рабочего процесса, даже если остаются задачи, которые должны быть выполнены.

Пример:

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.

В этом разделе содержатся следующие сведения.

Лимит времени 210
Оповещения рабочих процессов 211

Лимит времени

Лимит времени для выполнения операции рабочего процесса определяется на вкладке **Лимит времени** сведений об операции.

Примечание. 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, приложение Asset Manager воспринимает его как единицы 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.

Примечание. 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**).

Примечание. 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 приложением 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.

Программа Asset Manager Automated Process Manager отслеживает создание новых групп выполнения рабочих процессов.

Как только программа Asset Manager Automated Process Manager обнаруживает новую группу выполнения рабочих процессов "G", она 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 приложения Asset Manager Automated Process Manager.

После обнаружения группы выполнения рабочих процессов приложение Asset Manager Automated Process Manager отслеживает и выполняет соответствующие правила рабочих процессов (оповещения отслеживания, обработка событий типа **Периодическое**, активация переходов, выполнение задач и так далее).

В данном разделе содержатся следующие темы:

Приоритет 212

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

В этом разделе содержатся следующие сведения.

Отображение экземпляров рабочего процесса в записи 213
Просмотр информации на разных этапах экземпляра 213
Обновление схем и экземпляров рабочего процесса 214

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 происшедшие события.
  В правой части отображается рабочая схема. Выполняемые операции показываются как мигающие. Следующие шаги при этом затенены.
- Веб-клиент:
  В верхней части указываются происшедшие 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 A.

  Совет. Если используется клиент Windows, можно увеличить схему с помощью функции масштабирования.

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 хода выполнения (цветные стрелки), дата завершения операции, и т.д. Данная информация обновляется динамически: изменения автоматически отображается в графическом представлении экземпляра рабочего процесса (нажмите клавишу **F5** для обновления экрана).

- Информация, относящаяся к схеме рабочего процесса: список и положение ячеек и ссылок и т.д. Данная информация управляется **кэшем**, хранящимся в памяти на сервере Asset Manager Web Tier, и периодически обновляется (по умолчанию — каждые **24 часа**). Каждый элемент в кэше управляется отдельно.

  a. Когда вы открываете схему рабочего процесса впервые, схема рабочего процесса размещается в кэш и дата ее сохранения записывается.

  b. 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.

Чтобы изменить частоту обновления кэша, см. 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

В этом разделе содержатся следующие сведения.

Необходимость удаления завершенных рабочих процессов 215
Автоматизация удаления завершенных экземпляров рабочих процессов 215

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 приложения 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. Настройка приложения 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

Параметр	Значение
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 нужно выбирать эту опцию.
Описание (вкладка **Справка**)	<LI>Определяет, сколько времени должно пройти до того, как можно будет удалить устаревшие экземпляры рабочих процессов.
Пример	<LI>-1: экземпляры рабочих процессов никогда не удаляются.<LI>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 навигации **Администрирование/Действия**.
2. Щелкните **Создать**.
3. Заполните следующие поля:

Имя	имя SQL:	Значение
Имя	Имя	Удалить завершенные экземпляры

Имя	имя SQL:	Значение
Контекст	ContextTable	Схемы рабочих процессов (amWfScheme)
Тип	seActionType	Скрипт
имя SQL:	SQLName	DeleteFinishedWfInstances
Скрипт действия.	Скрипт	См. (*) ниже.

(*) Скрипт действия:

```
Const NumberOfInstanceToDelete = 50
Dim lRc As Long
Dim i As Long
i = 0
If [AutoCleaningDelay] >= 0 Then
  Dim hqWfInstance As Long
  hqWfInstance = AmQueryCreate()
  lRc = AmQueryExec(hqWfInstance, "SELECT lWfInstanceId FROM amWfInstance WH
ERE lWfSchId = "& [lWfSchId] & " And seStatus = 1 AND ADDSECONDS(dtCompleted,
" & [AutoCleaningDelay] & ") < GetDate()" )
  Do While (lRc = 0 And i < NumberOfInstanceToDelete)
    Dim hrWfInstance As Long
    hrWfInstance = AmGetRecordHandle(hqWfInstance)
    lRc = AmDeleteRecord(hrWfInstance)
    lRc = AmReleaseHandle(hrWfInstance)
    lRc = AmQueryNext(hqWfInstance)
    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 **Администрирование/Рабочий процесс/Схемы рабочего процесса**.

2. Щелкните **Создать**.

3. Заполните следующие поля:

Имя	имя SQL:	Значение
Имя	Имя	Удалить завершенные рабочие процессы

Имя	имя SQL:	Значение
Ссылка	Ref	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 приложении Asset Manager Automated Process Manager (например, **ADMIN**).
Единственный активный экземпляр рабочего процесса для объекта	bUniqueActive	Выбрать этот параметр.
Не сохранять экземпляры в базе данных	bTransient	Не нужно выбирать эту опцию.

4. Нажмите кнопку **Создать**.
5. Выберите вкладку **Операции**.
6. Выберите операцию **Запуск** и щелкните .
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 нужно выбирать эту опцию.
Условие AQL	AQLCond	`AutoCleaningDelay >= 0`

8. Выберите и заполните вкладку **Параметры** в соответствии с потребностями.
9. щелкните **Добавить**.
10. Щелкните правой кнопкой мыши и выберите меню **Добавить событие**.
11. Заполните следующие поля:

Метка	имя SQL:	Значение
Имя	Имя	Очистить экземпляры рабочих процессов
Тип	seType	Автоматическое действие
Записать задачу в журнал	bLogWorkItem	Выбрать этот параметр.
Контекст	ContextTable	Схемы рабочих процессов (amWfScheme)
Условие ввода	seInCond	OR

12. щелкните **Добавить**.
13. Выберите вкладку **Параметры**.
14. Заполните следующие поля:

Метка	имя SQL:	Значение
Выполнить действия немедленно	bExecImmediately	Выбрать этот параметр.
Действия	Действия	Удалить завершенные рабочие процессы

15. Щелкните **Закрыть**.
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. Щелкните **Закрыть**.
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 приложения Asset Manager Automated Process Manager для автоматического выполнения схемы рабочего процесса.

1. Запустите 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 приложение Asset Manager Automated Process Manager активным, если необходимо автоматическое выполнение рабочего процесса.

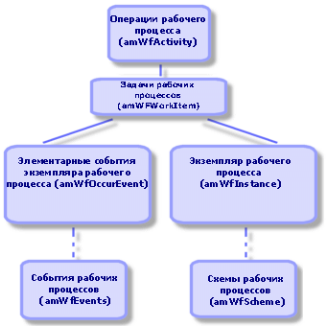
Техническая информация: модель данных

На следующей диаграмме представлены основные таблицы, необходимые для рабочего процесса, и связи между ними. Таблицы указываются их метками и именем SQL:

Рабочий процесс - основные таблицы, позволяющие определить схему рабочего процесса

На следующей диаграмме представлены основные таблицы, необходимые, когда выполняется экземпляр рабочего процесса:

Рабочий процесс - основные таблицы, необходимые, когда выполняется экземпляр рабочего процесса



Глава 9: Экспорт данных и создание представлений SQL

В этой главе объясняется, как экспортировать данные Asset Manager и управлять представлениями SQL базы данных:

Определения 221

Экспорт данных из базы данных Asset Manager 222

Управление представлениями SQL в базе данных Asset Manager 224

Рекомендации 224

Определение скрипта экспорта 224

Выполнение скрипта экспорта 229

Определения

В этом разделе содержатся следующие сведения.

Скрипты экспорта 221

Запросы экспорта 222

Скрипты экспорта

Скрипты экспорта позволяют экспортировать данные или (повторно) создавать /удалять представления SQL с помощью Asset Manager Export Tool или **amexp.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 средства 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 **Сервисные программы/ Экспорт списка**. Данный пункт меню доступен, если отображается хотя бы один список или список вкладок. Он позволяет экспортировать активный список.

Примечание. В операционных системах Windows Vista, Windows 8, Windows Server 2008, Windows Server 2008 R2 и Windows Server 2012 средству Asset Manager Export Tool для изменения некоторых локальных файлов требуются права администратора. Таким образом, для запуска средства Asset Manager Export Tool рекомендуется использовать параметр **Запуск от имени администратора**.

В этом разделе содержатся следующие сведения.

Экспорт данных с помощью скрипта экспорта 223
Экспорт данных с помощью пункта контекстного меню Экспорт списка 223

Экспорт данных с помощью скрипта экспорта

1. Запустите средство Asset Manager Export Tool с помощью меню **Запуск** или из группы программ Asset Manager.

2. Определите скрипт экспорта, для которого установлен режим "Экспорт данных":

   a. На вкладке **Запросы** запишите запросы, определяющие данные для экспорта.

   b. На вкладке **Форматирование** укажите формат текстовых файлов, в которые будут экспортироваться данные.

   c. С помощью пунктов меню **Файл/Сохранить скрипт** или **Файл/Сохранить скрипт как** сохраните скрипт.

3. Выполните скрипт экспорта:
   - непосредственно в Asset Manager Export Tool с помощью кнопки **Выполнить скрипт**

   - или запустив приложение **amexpl.exe** в DOS.

Примечание. Чтобы сохранить согласованность ограничений доступа, определенных в приложении Asset Manager, средство Asset Manager Export Tool или **amexpl.exe** можно запускать 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**.

Примечание. 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

С помощью средства Asset Manager Export Tool можно создавать, создавать повторно или удалять представления SQL в базе данных Asset Manager. Затем внешними инструментами такие представления могут использоваться вместо текстовых файлов.

> **Примечание.** Предостережение: представления SQL, которые скрипты экспорта позволяют создавать, изменять или удалять, отличаются от представлений Asset Manager в плане терминологии. Представление SQL эквивалентно инструкции SQL CREATE VIEW.

Создание, повторное создание или удаление представлений SQL в базе данных Asset Manager.

1. Запустите средство Asset Manager Export Tool.
2. Определите скрипт экспорта, для которого установлен режим "Создать/удалить представления SQL":
   a. На вкладке **Запросы** запишите запросы, определяющие данные для извлечения.
   b. На вкладке **Представления** укажите действия, которые необходимо выполнить: создать, изменить или удалить представления, непосредственно исполнить получающийся в результате скрипт SQL или сохранить файл.
   c. Сохраните скрипт экспорта.
3. Выполните скрипт экспорта:
   - непосредственно в модуле 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 используйте модуль Asset Manager Export Tool.

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 225
Определение запросов экспорта 225
Формат вывода скрипта экспорта 228
Действия, затрагивающие представления SQL 229

Методология

Чтобы создать или изменить скрипт экспорта:

1. Запустите средство 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 определите, требуется ли 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 модуле 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.

Примечание. 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(lModelId) 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.

> **Примечание.** 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 приложение 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.

> **Примечание.** 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 кодировку ANSI, OEM(DOS), UFT-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**.

> **Примечание.** 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:
  a. Щелкните кнопку , чтобы указать имя и путь файла.
  b. 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 229
Выполнение скрипта экспорта в DOS 230

Выполнение скрипта экспорта в Asset Manager Export Tool

Чтобы выполнить скрипта экспорта из Asset Manager Export Tool:

1. Запустите средство 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 завершен успешно, отобразится сообщение: "Скрипт выполнен успешно". В случае возникновения ошибок отобразится 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 модуля 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:<script>
-cnx:<cnx> [-login:<login>]
[-password:<password>]
```

-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 соединениями**).

-login: имя для входа администратора базы данных (Admin или пользователя с административными правами).

-password: пароль для имени входа.

Строки между символами <> не должны содержать пробелов.

Например:

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

Глава 10: Скрипты

В данной главе объясняется, как использовать скрипты.

Определение скрипта 233
Применение скриптов 235
Введение в функции 236
Классификация функций Basic 239
Первые шаги в написании скриптов 239
Библиотеки скриптов 242
Советы и предупреждения 244
Первый пример 249
Второй пример 251

Определение скрипта

В этом разделе содержатся следующие сведения.

Обзор 233
Информация об используемой версии языка Basic 234
Нотация доступа к данным 234

Обзор

Слово скрипт обычно означает программу, написанную на языке высокого уровня. В приложении 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 т.д.;
  - скрипты на языке Basic, которыми выполняются задачи, особенно в действиях.

    Примечание. 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 приложения 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 ApplicationsTM. 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 поддерживается функции доступа к файлам;
- имеется ограниченная поддержка функций даты и времени;
  Это особенно относится к Linux.
- недоступны элементы управления языка **Visual Basic for Applications**.

Примечание. 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:

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

Пример из таблицы моделей:

```
[Category.FullName]
```

Примечание. Для восстановления номера кода ссылки можно использовать следующий синтаксис:

```
[Link.Link]
```

Если необходимо сослаться на ссылку, можно использовать или имя SQL ссылки, или имя ее ключа.

Пример:

```
RetVal=[Contact.Location] или RetVal=[Contact.lLocaId]
```

В обоих примерах возвращается один и тот же результат, код ссылки.

Применение скриптов

Приложение Asset Manager позволяет связывать со скриптами на языке Basic следующие свойства:

- для конфигурирования значений полей по умолчанию (команда **Настроить объект** в контекстном меню);
- для значения по умолчанию признака, связанного с таблицей;
- в вычисляемых полях типа Basic;
- для конфигурирования полей (команда (**Настроить объект** 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

В этом разделе содержатся следующие сведения.

Определение функции 236
Встроенные функции и программируемые функции 236
Типы функций и параметров 238

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> (<Parameter> As <тип параметра>[, ..., <Parameter> As <ти
п параметра>]) As <тип функции>

<Программа (скрипт), выполняемая функцией. Эта программа должна определять возвр
ащаемое 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 приложении 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

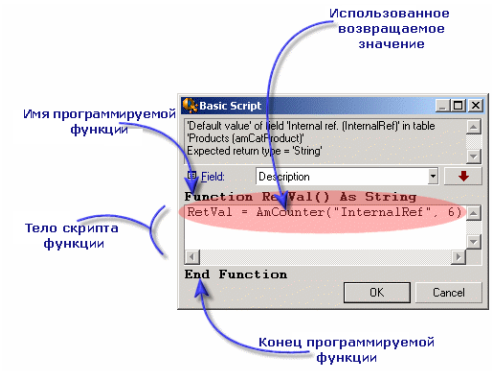
Программируемые функции - позиции программного обеспечения, которые пользователь может написать самостоятельно. Пользователь должен в явной форме определить значение, которое должно возвращаться (также называемое "возвращаемым значением"), в переменной **RetVal** с помощью программируемой функции в следующем формате:

```
RetVal=<выражение>
```

> **Примечание.** Приложение Asset Manager не будет 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:

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.

> **Примечание.** 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 приложение 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 приложение Asset Manager отображает сообщение об ошибке при выполнении функции.

Список типов

В следующей таблице приводится сводка различных типов, доступных для функции или параметра:

Функции/параметры - типы

Тип	Описание
Integer	Целое число от -2,147,483,648 до +2,147,483,647.
Long	Целое число от -2,147,483,648 до +2,147,483,647.
Double	8-байтное число с плавающей точкой.
Строка	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;
- функции типа Integer возвращают только целые значения (например,. 0, 1, 8, 12).
- функции типа String возвращают только строки символов (например: "Здание21") между кавычками.

Примечание. 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 240
Этап 1: создание признака "Руководство" 240
Этап 2: открытие окна редактирования 240
Этап 3: анализ и определение алгоритма 241
Этап 4: написание скрипта Basic 241

Этап 5: проверка скрипта Basic 241

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".

Этап 1: создание признака "Руководство"

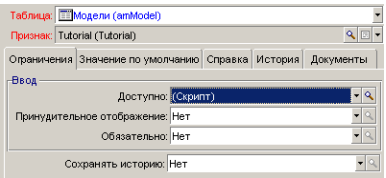
Выберите пункт меню навигации **Администрирование/ Признаки/ Признаки**. Нажмите кнопку **Создать** для создания нового признака. Заполните данный признак, как показано ниже:

Название поля	Значение
Заголовок (Имя SQL: TextLabel)	"Руководство"
Имя SQL (имя SQL: SQLName)	"Руководство"
Тип ввода (имя SQL: seDataType)	**Числовые**

Для создания признака нажмите **Создать** (клиент Windows) или **Сохранить** (веб-клиент).

Перейдите на вкладку **Параметры** и нажмите кнопку (клиент Windows) для изменения параметров признака "Руководство". Заполните вкладку **Ограничения**, как показано ниже.

> **Примечание.** В настоящее время изменение вкладки **Ограничения** в веб-клиенте невозможно.



Этап 2: открытие окна редактирования

На вкладке **Ограничения** установите для параметра **Доступно** (имя SQL: seAvailable) значение "(Script)". Щелкните кнопку увеличения . В Asset Manager открывается окно редактирования скрипта:

Этап 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 д
оступен, 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 .

Этап 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
```

Примечание. В скриптах регистр не учитывается

Для подтверждения скрипта нажмите кнопку **OK**.

Этап 5: проверка скрипта Basic

Это шаг позволяет убедиться, что скрипт функционирует правильно.

1. Откройте таблицу **Модели**, выбрав пункт меню навигации **Управление портфелем/Конфигурация активов/ Модели**. Нажмите кнопку **Создать** для создания новой модели.
2. Заполните только обязательные поля.

a. **Имя**

b. **Является подмоделью** (имя SQL: Parent) - "Компьютер/материнская плата".

c. **Вид** (имя SQL: Nature).

d. **Штрих-код** (имя SQL: BarCode).

3. Для создания новой модели нажмите **Создать** (клиент Windows) или **Сохранить** (веб-клиент).

4. Выберите вкладку **Признаки** и щелкните кнопку , чтобы добавить признак. На экране выбора показывается имя признака, для которого редактировался скрипт.

   Примечание. В настоящее время добавление признаков доступно только в клиенте Windows.

5. Измените значение поля **Является подмоделью** на "/Компьютер/" и подтвердите данное изменение, щелкнув **Изменить** (клиент Windows) или **Сохранить** (веб-клиент).

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

В приложении Asset Manager можно сохранять библиотеки скриптов, чтобы централизовать доступ к данным скриптам.

Библиотеку скриптов можно открыть с помощью пункта меню навигации **Администрирование/ Скрипты**.

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").

В этом разделе содержатся следующие сведения.

Концепции 242
Создание библиотеки скриптов 243
Вызов скрипта из библиотеки скриптов 243

Концепции

В Asset Manager с помощью скрипта определяется функция.

Создание библиотеки скриптов, таким образом, означает определение набора функций.

Создание библиотеки скриптов

Чтобы создать библиотеку скриптов:

1. Откройте список библиотек скриптов.
2. Щелкните **Создать**.
3. Занесите в поле **Имя** имя своей библиотеки.
4. В поле **Скрипт** введите свой скрипт.
5. Подтвердите скрипт с помощью кнопки **Создать** (клиент Windows) или **Сохранить** (веб-клиент).

   Примечание. Клиент Windows проверяет скрипт и выдает сообщение, если в скрипте содержатся синтаксические ошибки; веб-клиент сообщений об ошибках не выдает.

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)
   ```

3. Данным скриптом вызывается функция FullName из библиотеки lib и отображаются в диалогом окне имя и фамилия сотрудника.

4. Подтвердите скрипт с помощью кнопки **Создать** (клиент Windows) или **Сохранить** (веб-клиент).

   Примечание. Клиент Windows проверяет скрипт и выдает сообщение, если в скрипте содержатся синтаксические ошибки; веб-клиент сообщений об ошибках не выдает.

Примечание. Контекстный параметр, обычно используемый с интерфейсом API **amEvalSript**, 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 244
Формат констант "Дата+время" в скриптах 245
Формат констант "Продолжительность" в скриптах 245
Доступ для считывания и для записи к системному подробному списку 246
Виртуальная ссылка CurrentUser 247
Виртуальная ссылка "Старая" 248
Комментарии в скрипте на языке Basic 248
Инициирование сообщения об ошибке 248
Использование рекурсивных функций 249

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.

- В приложении Asset Manager широко используются программируемые функции. 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:

гггг/мм/дд чч:мм:сс

Пример:

```
RetVal="2001/07/12 13:05:00"
```

Примечание. 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).

Дата в Unix

Даты в Basic и в Unix выражаются по разному.

В Unix даты выражаются как целое число (32 бита), которое представляет собой число секунд, прошедших с полуночи 1 января 1870 года, независимое от часовых поясов (время по Гринвичу).

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.

Примечание. При выполнении преобразований в приложении 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

Приложение 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
```

Примечание. Не следует путать системный подробный список с определенным пользователем закрытым подробным списком.

Примечание. Полный список значений системных подробных списков включен в файл **database.txt**, который можно найти в подпапке **doc\infos**каталога 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>" ссылка возвращает значение поля для текущего пользователя.

Примечание. 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.lPersId]

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 **Executable**, введя в поле **Folder** (Папка), например, следующее действие: **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 приложении 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 ошибки>, [Hide system error])
```
```

- если для параметра **[Hide system error]** установлено значение '1', функция скрывает ненужное сообщение об ошибке (информация 'error in Line number').
- Если установлено значение '0', функция возвращает полную информацию.

> **Примечание.** 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.

Использование рекурсивных функций

В сценарии, когда рекурсивная функция в приложении Asset Manager использует локальные переменные, результат функции может быть неправильным. Известное ограничение: в данном сценарии скрипты Asset Manager поддерживают только рекурсию tail (также известную как вызов с префиксом tail).

Первый пример

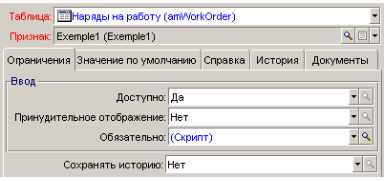
Этот раздел рассматривает гипотетическую проблему, которая может быть решена с помощью скрипта на языке Basic. Рекомендуем самостоятельно попытаться решить эту проблему, прежде чем обратиться к описываемому решению.

В этом разделе содержатся следующие сведения.

Описание проблемы 249
Этап 1: анализ и определение алгоритма 250
Этап 2: написание скрипта Basic 250
Этап 3: проверка скрипта Basic 251

Описание проблемы

При закрытии нарядов на работу необходимо заполнять признак с именем Example1, связанный с таблицей нарядов на работу. Данный признак является дополнительным для нарядов на работу, которые не закрыты. В остальной части примера предполагается, что признак уже создан, имеет произвольный тип ввода, связывается с таблицей нарядов на работу, доступен и отображается по умолчанию (принудительное отображение), как показано на записи экрана ниже:



Этап 1: анализ и определение алгоритма

Алгоритм должен выполнять следующие задачи:

- Установите для поля **Обязательный** (имя SQL: seMandatory) значение **Да**, если закрывается запись.
- Установите для поля **Обязательный** (имя SQL: seMandatory) значение **Нет** во всех остальных случаях.

Поэтому наш алгоритм будет таким:

```
Если закрывается наряд на работу,
то заполнение признака является обязательным,
иначе
заполнение признака необязательно
```

Наряд на работу закрывается, если значение поля **Статус** (имя SQL: seStatus) также **Закрытый**.

Поэтому значением поля **Статус** (имя SQL: seStatus) в таблице нарядов 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 нарядов 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
Закрыто	3

Поэтому интересующее значение в подробном списке:

- **Закрытый**, соответствующее значению "3".

Этап 2: написание скрипта Basic

```
If [seStatus]= 3 Then
 RetVal = 1
Else
```

```
 RetVal = 0
End If
```

Теперь необходимо написать алгоритм из шага 1 на языке Basic.

Для подтверждения скрипта нажмите кнопку **OK**.

Этап 3: проверка скрипта Basic

Это шаг позволяет убедиться, что скрипт функционирует правильно.

1. Откройте таблицу нарядов на работу, выбрав пункт меню **Жизненный цикл актива/ Наряды на работу/ Наряды на работу**. Выберите запись со значением **Закрытый** (или создайте наряд на работу с таким статусом, если нет ни одного).

2. Выберите вкладку **Признаки**. Приложение Asset Manager добавляет признак в соответствующий наряд на работу. Данный признак является обязательным.

3. Теперь выберите наряд на работу со статусом, отличным от **Закрытый**. Перейдите на вкладку **Признаки** данного наряда на работу. Признак "Example1" не отображается, но можно щелкнуть для добавления этого необязательного поля.

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.

Этап 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 значения полей **Фамилия** (имя SQL: Name) и **Имя** (имя SQL: FirstName) в таблице сотрудников определяют значение по умолчанию для поля **Field1** (имя SQL: Field1) в сведениях о сотруднике. Только два эти поля будут отображаться в нашем алгоритме.

Щелкните правой кнопкой мыши поле **Field1** (имя SQL: Field1) в сведениях о сотруднике и выберите **Настроить объект**.

Щелкните кнопку увеличения около поля **По умолчанию**, чтобы приступить к редактированию скрипта на языке Basic.

Этап 2: написание скрипта Basic

Теперь необходимо написать алгоритм из шага 1 на языке Basic.

```
If [FirstName]= "" Then
 RetVal = [Name]
Else
 RetVal = [Name]+", "+[FirstName]
End If
```

Для подтверждения скрипта нажмите кнопку **ОК**.

Этап 3: проверка скрипта Basic

Это шаг позволяет убедиться, что скрипт функционирует правильно.

1. Откройте таблицу сотрудников и подразделений с помощью ссылки **Управление организацией/ Организация/Сотрудники и подразделения** на навигационной панели и создайте нового сотрудника.

2. Заполните поля **Фамилия** (имя SQL: Name) и **Имя** (имя SQL: FirstName), а затем нажмите кнопку **Создать** для подтверждения. В приложении Asset Manager отображается фамилия и имя сотрудника в поле **Field1** (имя SQL: Field1).

Скрипт правильно выполняет свою функцию.

Глава 11: Календари

Используйте пункт меню навигации **Управление организацией/ Операции/ Календарь рабочих дней** для доступа к списку календарей.

Глава включает следующее:

Обзор календарей 253

Влияние календарей на некоторые области функциональности 254

Методология, используемая для создания календаря 255

Описание создания календаря 255

Обзор календарей

В этом разделе содержатся следующие сведения.

Сведения о календаре 253

Использование календарей 253

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:

- записи службы поддержки через схемы эскалации;
- 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.

Пример:

1. Создается запись в четверг, 25 сентября 2000 г. в 17:55.
2. Данная запись связывается со схемой эскалации, содержащей оповещение, которое должно инициироваться, если запись оказывается неназначенной в течение 15 минут после открытия.
3. Схема эскалации связана с календарем, в котором указано, что периоды в четверг 25 сентября и пятницу 26 сентября с 09:00 до 18:00 являются рабочими.
4. Оповещение будет инициировано в пятницу 26 сентября 2000 года в 09:10, если запись не назначена в этот момент.

Если к схеме эскалации не прикреплены никакие календари, все часы считаются рабочими часами.

Внимание! Если изменяется календарь в базе данных, изменения учитываются на уровне полей, связанных с календарем, только после выхода и перезапуска приложения Asset Manager.

Например, если запись связана со схемой эскалации, связанной с модифицированным календарем, дата **Ожидаемое разреш.** (имя SQL: dtResolLimit) на вкладке **Отслеживание** сведений о записи обновляется при следующем повторном запуске Asset Manager.

Влияние календарей на некоторые области функциональности

Календари влияют на некоторые области функциональности приложения Asset Manager. Изменение календаря вызывает как прямые, так и косвенные изменения некоторых записей в базе данных. Объекты, которыми используются календари:

- Схемы эскалации. Ими устанавливаются моменты времени, в которые инициируются оповещения.
- ссылки операций на обработку записей службы поддержки;
- время разрешения записей службы поддержки (выраженное в рабочих часах);
- даты разрешения записей службы поддержки;
- время приостановки записей службы поддержки;

- 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:

Ввод общей информации 255
Заполнение вкладки "Расписание" 256
Заполнение вкладки "Исключения" 257
Предварительный просмотр календаря 259

Ввод общей информации

Перед вводом фактических рабочих часов и исключений необходимо определить календарь, заполнив поле **Имя** (имя SQL: Name) на экране сведений.

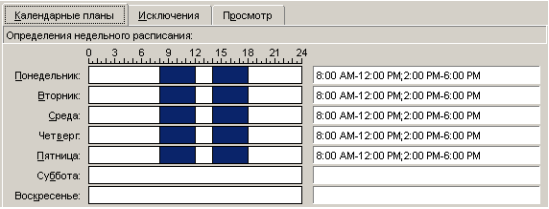
Можно связать календарь с часовым поясом, заполнив поле **Часовой пояс** (имя SQL: TimeZone).

Этим упрощается обработка записей, в которую вовлекаются активы или пользователи, расположенные в разных часовых поясах. При обработке таких записей соответствующие лимиты времени основываются на информации о часовых поясах.

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**.

Календарь - вкладка "Расписания"



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.
  a. Щелкните элемент управления в начале периода расписания.
  b. Расширьте выбор, перетащив курсор мыши к концу периода расписания. Приложение Asset Manager автоматически заполняет текстовое поле справа от градуированного элемента управления.
  c. 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, приложение Asset Manager считает, что используется 24-часовой формат.

Asset Manager автоматически заполняет градуированный элемент управления – ползунок, расположенный слева от текстового поля.

> **Примечание.** При использовании графического элемента управления значение устанавливается с точностью до получаса. При использовании текстового элемента управлении - с точностью до минуты.

Заполнение вкладки "Исключения"

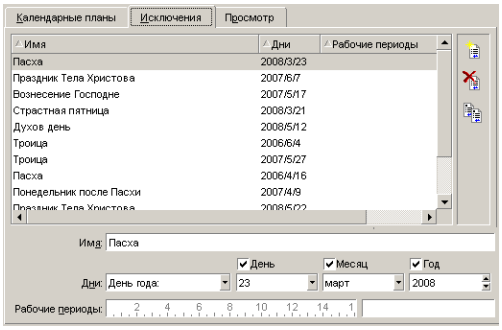
Вкладкой **Исключения** в сведениях о календаре определяются исключения в еженедельных периодах рабочего времени, определенных на вкладке **Расписания**.

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. Можно также определить период рабочего времени с исключением, используя поле **Рабочие периоды**. Это поле позволяет определить более точные исключения, например: "В последнюю пятницу каждого месяца группа работает с 08:30 до 10:30 и с 17:30 до 18:00".

Сведения вкладки "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

> **Примечание.** Столбец **Ранжирование** позволяет сортировать исключения по приоритету: он определяет, какое исключение будет приоритетным в случае конфликта. Приложение Asset Manager автоматически выполняет ранжирование исключений (от "P00" до "P15"). Чем меньше номер, тем более высокий приоритет у данного исключения. Такими образом, исключение с рейтингом "P06" имеет более высокий приоритет в сравнении с исключением "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.
"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 Taltek имеют следующие выходные дни:

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. Для завершения: Для поля **Дни** установите значение: "Первый".

Правило №2: В течение августа сотрудники Taltek работают только по утрам с 08:30 до 12:30.

1. Щелкните **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".

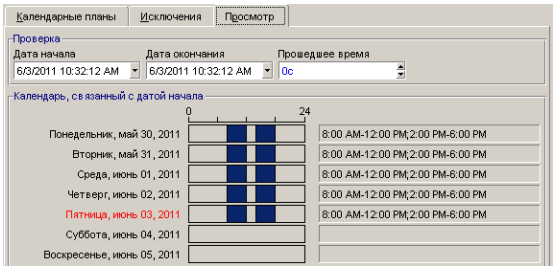
3. Так как исключение применяется только для выбора дней (все дни на протяжении месяца августа), для поля **Дни** необходимо установить значение "День года:".

4. В течение этого периода сотрудники работают с 08:30 до 12:30. Чтобы завершить ввод исключения, достаточно выбрать период с 08:30 до 12:30 в поле **Рабочие периоды**.

Предварительный просмотр 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**.

Календарь - вкладка "Предварительный просмотр"



- 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.

  Примечание. Можно ввести значение продолжительности в поле **Прошедшее время** для принудительного перерасчета поля **Дата окончания** в соответствии с

указанной **Датой начала**.

Глава 12: Часовые пояса

В этой главе объясняется, как использовать часовые пояса.

Примечание. Данная область функциональности доступна только для некоторых лицензионных соглашений Asset Manager.

Глава включает следующее:

Управление часовыми поясами 261

Внедрение часовых поясов 262

Создание часовых поясов 262

Управление часовыми поясами 263

Управление часовыми поясами в Asset Manager Automated Process Manager 269

Последствия для различных операций 270

Управление часовыми поясами

Так как клиентские компьютеры и сервер базы данных могут быть географически разнесены, приложение Asset Manager управляет часовыми поясами и разницей во времени по отношению ко времени по Гринвичу (GMT). В Asset Manager используются 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 наряда на работу на лондонском клиентском компьютере, если для срока разрешения наряда 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.
- На клиенте в Париже: 15 мая 2000 г. в 18:30.
- На клиенте в Лондоне: 15 мая 2000 г. в 17:30.

Внедрение часовых поясов

Чтобы часовые пояса правильно обрабатывались в приложении Asset Manager, необходимо выполнить следующие шаги:

1. Определите часовые пояса при создании базы данных в программе 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 приложении 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, приложение Asset Manager поставляется с файлом описания, содержащим основные часовые пояса. Данный файл можно импортировать с помощью следующей процедуры:

1. Выберите пункт меню **Файл/ Импорт** в клиенте Windows. В приложении Asset Manager откроется окно выбора импорта.
2. Выберите "Выполнить скрипт", щелкнув кнопку . В приложении Asset Manager откроется экран обновления базы данных. Щелкните , чтобы выбрать скрипт для

выполнения, в данном случае **tz.scr** в подпапке **datasys** каталога установки Asset Manager.

3. Нажмите кнопку **Импорт**. Приложение Asset Manager выполняет импорт в соответствии со скриптом.

Управление часовыми поясами

В данном разделе более подробно рассматривается поле **Летнее время** (имя SQL: memDaylightInfo).

В этом разделе содержатся следующие сведения.

Формат поля "Летнее время" 263
Значения аргумента <Year> 264
Значения аргумента <DaylightInfo> 264
Пример 267

Формат поля "Летнее время"

Поле **Летнее время** (имя SQL: memDaylightInfo) имеет следующую структуру (одной строкой):

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

До конца данного раздела используются следующие соглашения:

- <Year>=<DaylightInfo> вместе называются "параметром"
- <Year> и <DaylightInfo> по отдельности называются "аргументами"

В таблице ниже приведен обзор изменений летнего времени в соответствии со значениями аргументов <Year> и <DaylightInfo>.

	Аргумент <DaylightInfo> пуст	**Аргумент <DaylightInfo> имеет значение**
Аргумент <Year> пуст. ("<Year>=" не отображается)	Нет никаких изменений для летнего времени для часового пояса целиком.	Информация о летнем времени действительна для всех лет, кроме определяемых параметрами, имеющими аргумент <Year>.
Аргумент <Year> имеет значение	Неприменимо	Информация о летнем времени для данного часового пояса действительна для любого года из числа лет, указанных аргументом <Year>, вплоть до следующего аргумента <Year>.

Значения аргумента <Year>

Аргумент <Year>, 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> имеет следующий вид (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 <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.
<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.

Субаргумент	Описание	Возможные значения
<SDay>	День перехода с летнего времени на стандартное время.	"Понедельник" "Вторник" "Среда" "Четверг" "Пятница" "Суббота" "Воскресенье" Пусто (в таком случае, необходимо установить для <SDayPos> значение от 1 до 31, чтобы указать день перехода со стандартного времени на летнее время)
<SMonth>	Месяц перехода с летнего времени на стандартное время.	"Январь" "Февраль" "Март" "Апрель" и т.д. "Ноябрь" "Декабрь"
<SDayPos>	Положение в месяце дня перехода с летнего времени на стандартное время.	"Первый" "Второй" "Третий" "Четвертый" "Последний" "Предпоследний" (перед последним) Значение от 1 до 31, если субаргумент <SDay> пуст
<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.	"Понедельник" "Вторник" "Среда" "Четверг" "Пятница" "Суббота" "Воскресенье" Пусто (в таком случае, необходимо установить для <DDayPos> значение от 1 до 31, чтобы указать день перехода с летнего времени на стандартное время).
<DMonth>	Месяц перехода со стандартного времени на летнее время.	"Январь" "Февраль" "Март" "Апрель" и т.д. "Ноябрь" "Декабрь"
<DDayPos>	Положение в месяце дня перехода со стандартного времени на летнее время.	"Первый" "Второй" "Третий" "Четвертый" "Последний" "Предпоследний" (перед последним) Значение от 1 до 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 используемых параметров.

Первый параметр

- <Year> = 2000 означает, что следующие параметры <DaylightInfo> применимы с 2000 года и далее.
- <StdShift> = 0 означает, что в этом часовом поясе нет различия между часовым поясом и стандартным временем.
- <DltShift> = 60 означает, что разница между стандартным временем и летним временем составляет 60 минут, т.е. 1 час. Таким образом, летнее время равняется времени часового пояса плюс один час.
- <SDay> = Sunday означает, что переход на стандартное время осуществляется в воскресенье.
- <SMonth> = October означает, что переход на стандартное время осуществляется в октябре.
- <SDayPos> = Last обозначает 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> = Sunday означает, что переход на стандартное время осуществляется в воскресенье.
- <DMonth> = March означает, что переход на летнее время осуществляется в марте.
- <DDayPos> = Last обозначает 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 стандартное время осуществляется в 02:00.

Второй параметр

- Так как нет аргумента <Year>, 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 разница между стандартным временем и летним временем составляет 60 минут, т.е. 1 час. Таким образом, летнее время равняется времени часового пояса плюс один час.
- <SDay> = Sunday означает, что переход на стандартное время осуществляется в воскресенье.
- <SMonth> = September означает, что переход на стандартное время осуществляется в сентябре.
- <SDayPos> = Last обозначает 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> = Sunday означает, что переход на стандартное время осуществляется в воскресенье.
- <DMonth> = March означает, что переход на летнее время осуществляется в марте.
- <DDayPos> = Last обозначает 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 стандартное время осуществляется в 02:00.

В результате:

Примечание. 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

В приложении Asset Manager Automated Process Manager можно настраивать тесты для часовых поясов. Выберите пункт меню **Инструменты/Настройка модулей**.

В этом разделе содержатся следующие сведения.

Тесты, которые необходимо выполнить 269
Частота тестирования 270

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 установлена программа Asset Manager Automated Process Manager. Разница во времени выражается как [(n * 30 минут) + m], где m находится в диапазоне от -15 до +15 минут.

В обоих случаях

Если разница во времени превышает 5 минут, приложение Asset Manager Automated Process Manager предлагает обновить местное время на компьютере, где оно установлено.

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 установлена программа Asset Manager Automated Process Manager).

Конкретные аспекты параметра "Проверить часовой пояс сервера базы данных"

При необходимости можно обновить сведения о часовом поясе сервера в таблице параметров Asset Manager (если число (n * 30 минут) не соответствует часовому поясу сервера).

> **Примечание.** 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. При первом подключении программы Asset Manager Automated Process Manager к базе данных.
2. Затем регулярно, в соответствии с расписанием, указанным в программе Asset Manager Automated Process Manager (**Инструменты/Настройка модулей**).

Последствия для различных операций

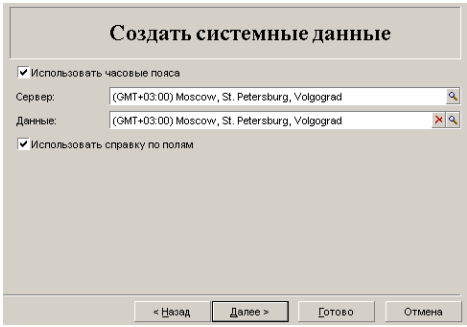
Способ, которым управляются часовые пояса,оказывает влияние на целый ряд операций:

В этом разделе содержатся следующие сведения.

Создание базы данных 270
Подключение к базе данных 271
Импорт и экспорт 272
Календари и схемы эскалации 272

Создание базы данных

При создании базы данных Asset Manager позволяет определять параметры часовых поясов. Выберите пункт меню **Действие/ Создать базу данных** в приложении 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.

> **Примечание.** 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 приложение Asset Manager выполняет в файле **am.ini** поиск записи LocalTimeZone, которой определяется часовой пояс клиентского компьютера.

Местоположение файла: См. руководство **Asset Manager — установка и обновление**, глава **Файлы .ini и .cfg**.

Если не удается найти эту информацию, приложение Asset Manager использует системный часовой пояс (определенный в Windows).

Далее Asset Manager ищет базу данных для часового пояса, соответствующего 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 приложение 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. Приложение Asset Manager рассчитывает часовой пояс клиентских часов и

определяет расхождение в показаниях часов между двумя компьютерами. Такой расчет выполняется следующим образом:

```
Расхождение = Модуль((Разность в минутах между временами двух рассматриваемых ко
мпьютеров)/30)
```

> **Примечание.** Модуль представляет собой остаток от деления.

Например, для следующих компьютеров:

- компьютер A находится в часовом поясе GMT и его часы показывают 18:02;
- компьютер B находится в часовом поясе GMT+1 и его часы показывают 18:19 (то есть 17:19 для компьютера A, 43 минуты разницы с компьютером A))

```
Расхождение = Модуль (43/30)= 13 минут
```

Если такая разница превышает пять минут (фиксированное значение), приложение Asset Manager предлагает откорректировать показания часов на уровне клиента.

В случае отказа пользователя происходит сбой подключения.

Приложение 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_lTimeZoneCheckInMns** в файле **am.ini**, в разделе [option].

Местоположение файла: см. руководство **Asset Manager — установка и обновление**, глава **Файлы .ini и .cfg**.

```
[option] g_lTimeZoneCheckInMns = 30
```

Для периодичности проверки расхождения в показаниях часов устанавливается значение 30 минут.

Данная периодичность может также настраиваться с помощью параметра **Проверка часового пояса сервера базы данных** в приложении Asset Manager.

> **Примечание.** 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.

Календари и схемы эскалации

Календари могут связываться с часовыми поясами.

В результате:

- значения времени выражаются в календаре в локальном времени по отношению к выбранному часовому поясу;
- пользователь может выбирать календарь, совместимый со схемами эскалации.

Глава 13: Вычисляемые поля

Доступ к экрану для создания вычисляемого поля возможен с помощью меню **Администрирование/ Система/ Вычисляемые поля**.

Глава включает следующее:

Определение вычисляемого поля 275

Полезность вычисляемых полей 275

Создание вычисляемых полей 276

Использование вычисляемых полей 281

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.

Примечание. 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 276
Методы создания 278

Введение

Каждый тип вычисляемых полей имеет разные свойства, которыми определяется, как он используется.

В следующей таблице собраны основные отличия между тремя типами:

Типы вычисляемых полей

Тип поля	Свойства полей данного типа			Поле рассчитывает	Характеристики языка, используемого формулой расчета	
	Может отображаться	Может сортироваться	Может использоваться в фильтрах		Преимущества	Недостатки

Типы вычисляемых полей, продолжение

Тип поля	Свойства полей данного типа			Поле рассчитывает	Характеристики языка, используемого формулой расчета	
AQL	Да	Да	Да	Сервер базы данных	Эффективен Встроенный редактор	Ограниченный по возможностям язык. Поля данного типа не могут использоваться в значениях по умолчанию
Вычисляемые строки	Да	Да	Нет	Клиент	Простые	Не очень эффективен (только простое объединение строк и значений полей и строк)
Basic	Да	Нет	Нет	Клиент	Много возможностей Гибок	Поля данного типа могут только отображаться

Примечание. 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` предложение

Свойство	Соответствующие аргументы AQL

Может сортироваться	`SELECT`
	`ORDER BY`
	`GROUP BY` предложения

Может использоваться в фильтрах	`SELECT`
	`ORDER BY`
	`GROUP BY`
	`WHERE`
	`HAVING` предложения

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 этом разделе 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:
  - Вычисляемые строки
  - AQL
  - Basic

Примечание. Всюду, где возможно, рекомендуется использовать наименее "затратное"

решение.

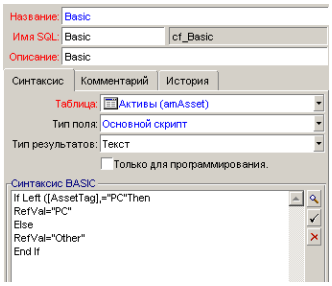
Например:

- 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/ Система/ Вычисляемые поля**. В приложении Asset Manager отобразится экран создания вычисляемого поля:



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;

  Примечание. 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.

> **Примечание.** 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 меню навигации **Администрирование/ Права/ Права пользователя**. В приложении Asset Manager отображается экран создания прав пользователя.

> **Примечание.** Вычисляемые поля доступны только для доступа к статусу.

1. Введите краткое описание прав пользователя в поле **Описание** (имя SQL: Description) (имя SQL: Description), а также, если требуется, комментарий в поле **Комментарий** (имя SQL: Comment).

2. Если используется клиент Windows, разверните древовидную структуру таблицы, связанной с вычисляемым полем. Ветвь, указываемая значком f(x)(Вычисляемые поля), включает полный список вычисляемых полей для рассматриваемой таблицы.
   Если используется веб-клиент, выберите таблицу, а затем выберите вычисляемое поле из списка категорий для компонентов 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), приложение Asset Manager упрощает задачу, отображая только поля, которые можно использовать.

В этом разделе содержатся следующие сведения.

Использование вычисляемых полей в конфигурациях списков 281
Фильтрация записей таблицы 281
Ссылки на вычисляемые поля 282

Использование вычисляемых полей в конфигурациях списков

Можно отобразить значение поля, рассчитанное для всех записей в таблице, с помощью команды **Настройка списка** в контекстном меню.

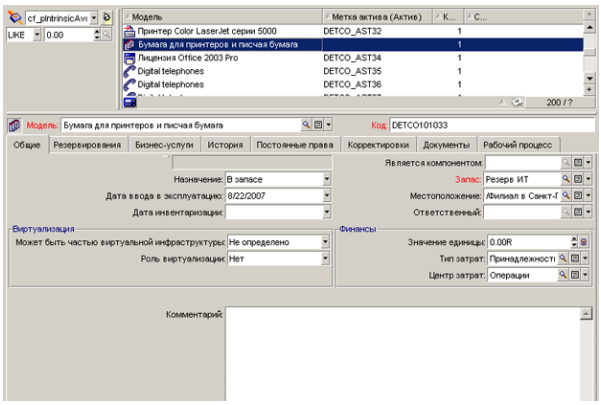
Внимание! Эта функция не работает, если в качестве СУБД используется Microsoft SQL Server (вычисляемое поле не отображается).

Фильтрация записей таблицы

Приложение Asset Manager позволяет фильтровать записи в таблице в соответствии со значением вычисляемого поля с типом AQL. Для этого выберите в контекстном меню команду **Простой фильтр** и перейдите к ветви **Вычисляемые поля**. Приложение Asset Manager отображает только поля с типом 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.

Глава 14: Мастера

Приложение Asset Manager позволяет создавать собственные мастера, а также адаптировать существующий мастер под собственные нужды. Мастера хранятся в виде текстовых полей (поле **Скрипт мастера** (имя SQL: WizardScript), вкладка **Мастер** сведений действия с типом **Помощник**). Создание мастера представляет собой ввод его кода непосредственно в этом поле или с помощью графического редактора. Выполнение этих действий предполагает знакомство со структурой мастеров, а также с языком скриптов, используемым для описания данной структуры.

Внимание! Если вы модифицируете действия в отношении скрипта или мастера, обязательно повторно отметьте веб-службы, если выполнено развертывание Asset Manager Web.

См. руководство **Tailoring**, глава **Customizing the database**, раздел **Development best practices/ Tag the Web services**.

Глава включает следующее:

Обзор 283

Соединение мастеров 289

Функции Basic 290

Свойства узла 291

Типы узлов 295

Типы элементов управления и связанные свойства 312

Использование графического редактора 349

Пример создания мастера 352

Пример программирования мастера 358

Часто задаваемые вопросы 364

Обзор

В этом разделе содержатся следующие сведения.

Соглашения о нотации 284

Определения 284

Структурный шаблон 286

Шаблон страницы мастера 287

Общее число пунктов 288

Общая структура и синтаксис 288

Соглашения о нотации

Ниже приводится нотация, используемая для описания структуры мастеров:

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 приложением Asset Manager.
; или //	В мастерах точка с запятой или две косые черты обозначают строку комментариев, которая не интерпретируется приложением 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.

Контроль

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.

Примечание. 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 т.д.) на странице;
- переменная;
- и т. д.

Родительский объект и дочерний объект

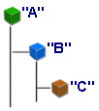
Если объект A содержит объект B:

- 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 приложении Asset Manager используются два типа переменных.

- "Переменные мастера", которые определяются в узлах с типом "LONG" или "STRING". Тип узла определяет тип переменной; переменная, определенная в узле "LONG", – это длинное целое, а переменная, определенная в узле "STRING", – это строка символов. Данные переменные по определению являются глобальными. То есть из любого узла в мастере невозможно создать ссылки на эти переменные с помощью их полного имени. Если необходимо, эти переменные пересчитываются автоматически приложением 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 приложением Asset Manager.

Переход

Переход - процесс смены одной страницы мастера на другую. Для данной страницы могут определяться несколько переходов. Каждый из них имеет свои собственные пользовательские условия, от которых зависит их действительность и которые должны выполняться, чтобы инициировать переход.

- Когда пользователь нажимает 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 получается в результате выполнения мастера. По определению структура мастера основывается на скрипте структуры. 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:

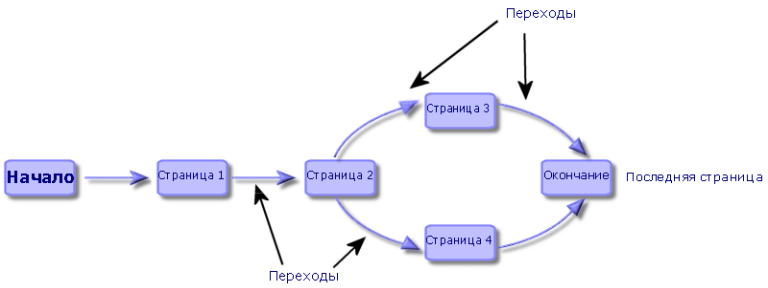
  - "ROOT" (Корневой узел). Данный узел является уникальным и заключает в себе все остальные.

- "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).

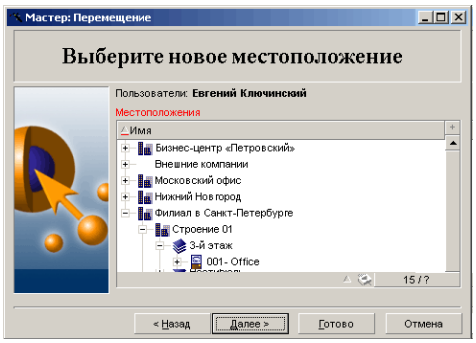
Мастера состоят из страниц, связанных переходами. Переход с одной страницы на другую определяется информацией, введенной сделавшим выбор пользователем. На диаграмме ниже проиллюстрирована структура мастера:

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 т.д.) в древовидной структуре мастера имеет неограниченное число подузлов и набор свойств.

Общая структура и синтаксис

Узлы имеют следующие структуру и синтаксис:

```
; Это комментарий вне скрипта
{ <тип узла> <имя узла>
<имя свойства> = <значение свойства>
' Это комментарий внутри скрипта
...
{ <имя свойства> =
                     ...
}
{ <тип узла> <имя узла>
                   <имя свойства> = <значение свойства>
                   ...
                   { <имя свойства> =
                                  ...
                    }
}
}
```

Для узла применяются следующие правила:

- Имена узлов являются необязательными. Если для узла не указано имени, приложение 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" данный главы.
- Строки, начинающиеся с точки с запятой (;) за пределами скрипта, и строки, начинающиеся с апострофа (') внутри скрипта, интерпретируются как комментарии и игнорируются.

Примечание. Внимание: должен соблюдаться пробел между скобкой ("{") и типом узла. В противном случае приложение Asset Manager не будет выполнять мастер.

Соединение мастеров

После выполнения мастер может использоваться для инициирования выполнения другого мастера и передавать параметры (переменные) в другой мастер. Это называется соединением мастеров.

В этом разделе содержатся следующие сведения.

Выполнение 289
Параметры 289

Выполнение

Для того, чтобы мастер A инициировал запуск мастера B, его конечный узел не должен иметь свойство CHAIN. Данное свойство должно иметь значение имени SQL действия типа **Помощник**, которое должно выполняться, в данном случае B.

Совет. 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.

Частное действие - это действие, для которого отмечено поле **Не общее** (bPrivate). В данном случае оно скрыто для всех пользователей, за исключением владельца (author).

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 B при помощи узла PARAMS мастера A. Эти параметры добавляются к параметрам узла PARAMS мастера B. Если один и тот же параметр определяется для узла PARAMS мастера A и узла PARAMS мастера B, параметр в мастере A преобладает над параметром в мастере B.

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

В этом разделе содержатся следующие сведения.

Значения, возвращаемые функциями 290
Сцепление строк в скриптах на языке Basic 291

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, "lUserId", {lEmplDeptId})
```

Правильным скриптом будет следующий:

```
Dim lValue as Long
lValue=AmGetFieldLongValue(hRecord, "lUserId", {lEmplDeptId})
```

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.

Свойства узла

Значения свойств могут определяться с помощью констант или скриптов. Константы могут быть числовые, булевы или текстовые.

> **Примечание.** Свойства, связанные с объектами, могут быть дополнительными или обязательными. Они могут быть или логическими (дополняющими определение объекта), или физическими (имеющими влияние на визуальное восприятие объекта).

В этом разделе содержатся следующие сведения.

Декларативный шаблон 291
Определение константы как значения свойства 292
Ссылка на свойство 292
Определение скрипта как значения свойства 292
Методы, применимые в свойствах 293
Свойство типа "Таблица" 294
Использование глобальных переменных CurrentTable и CurrentSelection 295

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 свойства>}

Примечание. Булево значение TRUE эквивалентно числовому значению, отличному от 0. Значение "FALSE" эквивалентно числовому значению 0.

Ссылка на свойство

Для ссылки на свойство объекта (т.е. создание ссылки на содержимое данного свойства или объекта и, в частности, 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}="Выбрать сотрудника" THEN
RetVal="Сотрудник"
                ELSE
RetVal="Подразделение"
                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).

Примечание. 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:

```
<Colunm|Colunm|Colunm|...>=<код строки>, <Colunm|Colunm|Colunm|...>=<код строк
и>, ...
```

Значения данных свойств можно просмотреть в форме таблицы:

		Столбец 1	Столбец 2	Столбец 3
Номер строки: 1	**код строки (например, 18)**	Ячейка (1,1)	Ячейка (2,1)	Ячейка (3,1)
Номер строки: 2	**код строки (например, 29)**	Ячейка (1,2)	Ячейка (2,2)	Ячейка (3,2)
Номер строки: 3	**код строки (например, 78)**	Ячейка (1,3)	Ячейка (2,3)	Ячейка (3.3)
и т. д.	и т. д.	и т. д.	и т. д.	и т. д.

Примечание. Идентификатор - идентификатор текстового типа.

Пример

Допустим, что свойство "VALUES" узла "LISTBOX" получает свое значение в результате запроса к таблице сотрудников и подразделений. Данный запрос возвращает значения полей **Фамилия** (имя SQL: Name) и **Имя** (имя SQL: FirstName) для каждой записи в этой таблице. Предположим, что это 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**	Lübeck	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".	Эта переменная автоматически заполняется приложением Asset Manager. Пользователь не может принудительно присвоить значение.
CurrentSelection	Содержит список внутренних идентификаторов записей, выбранных во время запуска мастера, разделенных запятыми. Переменная типа "String".	Эта переменная автоматически заполняется приложением Asset Manager>. Она содержит пустую строку, если ничего не было выбрано или если не были обновлены никакие таблицы. Пользователь не может принудительно присвоить значение.

Типы узлов

В этом разделе содержатся следующие сведения.

Корневой узел 296
Узел "Страница" 304
Узел перехода 306
Свойства узла перехода 307
Конечный узел 308
Начальный узел 310
Узел таймера 310
Узлы "Длинный" и "Строка" 311

Узел элемента управления 312

Корневой узел

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:

```
' Block of general properties of the root node NAME=... IMAGE=... ... ' Definiti
on of sub-nodes of the root node { FINISH ... } { PAGE ... } { TRANSITION ... }
```

Свойства корневого узла

В приведенных ниже таблицах перечислены все логические и физические свойства, которые можно определить в корневом узле.

Логические свойства корневого узла

Имя свойства=Значение	Описание свойства	Пример	Комментарий
NAME="<Имя мастера>"	Определяет название окна мастера. Свойство типа "String".	NAME = "MoveWizard"	Необходимо задать значение для данного свойства. Длина имена мастера не должна превышать 22 символов. Данное свойство используется для сериализации мастера: данные, относящиеся к мастеру, хранятся под этим именем. Поэтому желательно для разных мастеров использовать разные имена.

Логические свойства корневого узла, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
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) предварительно 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}	

Логические свойства корневого узла, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
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.		

Логические свойства корневого узла, продолжение

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**) предварительно введенные значения для выполнения в следующий раз. Свойством NAME определяется, в каком разделе файла **.ini** будут храниться значения. Свойство типа "Boolean". Местоположение файлов **.ini**: см. руководство **Asset Manager — установка и обновление**, 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).		

Физические свойства корневого узла

Имя свойства=Значение	Описание свойства	Пример	Комментарий
IMAGE="<путь к файлу точечного рисунка> "IMAGE16="<путь к файлу точечного рисунка>"	Определяет графический файл точечного типа (.bmp) для отображения в мастере. Свойство типа "String".	IMAGE="C:\Images\Page1.bmp"	Если для данного свойства не определено никакого значения, то не будет отображаться никакого изображения. Путь к файлу графики зависит от папки Config в Asset Manager. Приложение Asset Manager сначала выполняет поиск изображения в базе данных. Если определить значение "IMAGE16", то это свойство будет использоваться вместе "IMAGE" при глубине цвета экрана равной 16.

Физические свойства корневого узла, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
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	
MINWIDTH=<MinWidth>	Определяет минимальную ширину окна мастера. Значение, выраженное в **твипах**.		
MINHEIGHT=<MinHeight>	Определяет максимальную высоту окна мастера. Значение, выраженное в **твипах**.		

Физические свойства корневого узла, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
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**.		

Физические свойства корневого узла, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
IMGBORDER=<ширина>	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.

Подузлы корневого узла

Тип узла	Описание
PAGE	Описывает страницу мастера.
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.

Узел "Страница"

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 pa
ge node IMAGE=... TITLE=... ' Определение подузлов узла "Страница" { TRANSITION
... } { <Control type> <Control name> ... } ... }
```

Свойства узла "Страница"

В приведенных ниже таблицах перечислены все свойства, которые можно определить в узле "Страница".

Логические свойства узла "Страница"

Имя свойства=Значение	Описание свойства	Пример	Комментарий
TITLE="<Название окна>"	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") }	

Физические свойства узла "Страница"

Имя свойства=Значение	Описание свойства	Пример	Комментарий
IMAGE="<путь к файлу точечного рисунка>" IMAGE16="<путь к файлу 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:

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.

Узел перехода

Определение узла перехода

Узлом перехода описывается переход между двумя страницами мастера. Она состоит исключительно из блока свойств.

Примечание. 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:

```
' Declaration of the transition { TRANSITION0 <Name of the transition> ' Block o
f properties of the transition node FROM=... TO=... CONDITION=... }
```

Свойства узла перехода

В приведенной ниже таблице перечислены все свойства, которые можно определить в узле перехода:

Логические свойства узла перехода

Имя свойства=Значение	Описание свойства	Пример	Комментарий
FROM="<имя исходной 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.

Логические свойства узла перехода, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
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.

Конечный узел

Конечный узел описывает последний переход: тот, который ведет на последнюю страницу мастера. Это особый тип узла перехода, который не имеет свойств FROM или 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.

Логические свойства конечного узла

Имя свойства=Значение	Описание свойства	Пример	Комментарий
CHAIN=<имя SQL мастера, подлежащего выполнению>	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.		

Примечание. Узел PARAMS позволяет передавать параметры следующему мастеру.

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.	

Физические свойства конечного узла, продолжение

Имя свойства=Значение	Описание свойства	Пример
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.	

Начальный узел

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.

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".	

Примечание. Если данный узел не существует, мастер запускается с первой страницы.

Узел таймера

Узел таймера позволяет выполнять задачу через регулярные интервалы.

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, зависимое от данного интервала, будет автоматически пересматриваться при каждом вхождении. Данное значение (tickcount) автоматически нарастает.	

Узлы "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 свойство, тип которого зависит от узла; свойство типа Long для узла "Длинный" или String – для узла "Строка". Данное свойство, VALUE, позволяет определить значение переменной.

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	

Примечание. Узлы "Длинный" и "Строка" могут определяться в любом узле мастера. У них нет подузлов.

Узел элемента управления

Определение узла элемента управления

Элементы управления страницы для взаимодействия с пользователем. Для данной страницы можно определить столько элементов управления, сколько необходимо. Приложение 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:

```
' Declaration of the control { <Control type> <Control name> ' Properties of the control ... }
```

Типы элементов управления и связанные свойства

Все элементы управления имеют общие свойства. Однако есть свойства, специфичные для конкретных элементов управления.

В этом разделе содержатся следующие сведения.

Общие свойства 313
Элемент управления CHECKBOX 320
Элемент управления COMBOBOX 320
Элемент управления OPTIONBUTTONS 321

Элемент управления LISTBOX 322
Элемент управления LABEL 329
Элемент управления PROGRESSBAR 330
Элемент управления COMMANDBUTTON 330
Элемент управления DBLISTBOX 331
Элемент управления DBQUERYBOX 335
Элемент управления DBEDIT 339
Элемент управления DBTABLE 341
Элемент управления DBPATH 341
Элемент управления LINKEDIT 341
Элемент управления TEXTBOX 343
Элемент управления CHART 343
Элемент управления FILEEDIT 346
Элемент управления TICKEDIT 347
Элемент управления CALENDAR 347
Элемент управления TIMESPANEDIT 347
Элемент управления NUMBOX 347
Элемент управления COMBOEDIT 348
Элемент управления DATETIMEEDIT 348

Общие свойства

В приведенной ниже таблице перечисляются дополнительные свойства, применимые для всех элементов управления.

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

Логические свойства, общие для всех элементов управления, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
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 т.д.).	• Для элемента управления CHECKBOX <значение> 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 использоватьс я только значение, определенное в первом положении свойства VALUE.

Логические свойства, общие для всех элементов управления, продолжение

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 активизирована (=TRUE), можно деактивизировать ее на уровне элемента управления.		По умолчанию, данное свойство имеет значение свойства SERIALIZE в корневом узле.

Логические свойства, общие для всех элементов управления, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
HELP = "help"	Данное свойство позволяет включать текст справки в формате HTML в элемент управления мастера. Данная справка вызывается нажатием сочетания клавиш SHIFT+F1.		

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	
ENABLED= <TRUE|FALSE>	Определяет, является ли элемент управления активным (=TRUE) или нет (=FALSE). Свойство типа "Boolean". Свойство типа "Boolean".	Choice1.Enabled=FALSE	

Физические свойства, общие для всех элементов управления, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
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".		

Физические свойства, общие для всех элементов управления, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
XOFFSET	Определяет пространство, резервируемое для метки элемента управления, если она помещается слева от элемента управления (с помощью свойства LABELLEFT). Свойство типа **Твип**.		
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 заголовок. Свойство типа **Твип**.		
YOFFSET	Определяет смещение до элемента управления и его заголовка. Свойство типа **Твип**.		

Физические свойства, общие для всех элементов управления, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
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:

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:

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 приложение 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:

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:

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 заголовок соответствующего столбца. Свойство типа "String".	COLTITLE = "Фамилия|Имя"	

Физические свойства элемента управления LISTBOX, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
COLWIDTH = "<ширина|ширина|ширина...>"	Определяет размер столбцов пропорционально полному размеру элемента управления. Свойство типа "String".	COLWIDTH = "50|50"	

Физические свойства элемента управления LISTBOX, продолжение

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, приписываемое элементу управления, если данная строка выбирается пользователем. Свойство типа "String".	VALUES="Таблица активов=актив, , Пользователь=пользователь,"	Если пропустить "Значение", то приложение 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.

Физические свойства элемента управления LISTBOX, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
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"	**Примечание.** Добавьте соответствующее имя SQL таблицы, если необходимо сделать определенные столбцы редактируемыми. Например, определите TABLE="amEmplDept", если элемент управления LISTBOX отображает столбцы из таблицы «Сотрудники и подразделения».

Физические свойства элемента управления LISTBOX, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
COLNAME="<имя|имя|имя>"	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 т.д.).	COLNAME="фамилия|имя|dtHire"	**Внимание!** Свойство COLNAME **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	

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)	
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).		

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, что используется в приложении Asset Manager для заполнения поля. Для определения заголовков столбцов приложение Asset Manager будет сначала учитывать значения свойства COLTITLE (если существует). Свойство типа "String".	COLNAME="фамилия|имя"

Элемент управления LABEL

Элементом управления LABEL просто определяется метка. Данный элемент управления имеет следующее свойство:

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:

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:

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

Физические свойства элемента управления COMMANDBUTTON, продолжение

Имя свойства=Значение	Описание свойства	Пример
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.

Примечание. 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:

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.

Физические свойства элемента управления DBLISTBOX, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
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 = "Фамилия|Имя"	
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.

Физические свойства элемента управления DBLISTBOX, продолжение

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"
MULTISEL = <TRUE|FALSE>	Указывает, поддерживает ли элемент управления множественный выбор (=TRUE) или нет (=FALSE). Свойство типа "Boolean".	MULTISEL=TRUE	
DBLCLICK = <TRUE|FALSE>	Если для данного свойства установлено значение TRUE, приложение Asset Manager будет имитировать нажатие кнопки **Далее** текущей страницы при каждом двойном щелчке по строке.	DBLCLICK=FALSE	
FILTER= "<условие>"	Определяет условие AQL WHERE для фильтрования записей, которые должны быть обработаны в запросе. Свойство типа "String".	FILTER= "User.lEmplDeptId='Colombo, Gerard' "	

Физические свойства элемента управления DBLISTBOX, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
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.)	

Физические свойства элемента управления DBLISTBOX, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
TABLE.LABEL ([iNameType])	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:

Физические свойства элемента управления DBQUERYBOX

Имя свойства=Значение	Описание свойства	Пример	Комментарий
QUERY="<полный запрос AQL>"	Определяет запрос AQL, который возвращает информацию, отображаемую 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 = "Фамилия|Имя"	
COLWIDTH="<ширина|ширина|...>"	Определяет размер столбцов в списке в процентах от полного размера элемента управления. Свойство типа "String".	COLWIDTH = "50|50"	

Физические свойства элемента управления DBQUERYBOX, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
LISTHEIGHT = <процент>	Определяет размер данного элемента управления DBQUERYBOX пропорционально к остальным элементам управления DBQUERYBOX всего мастера. Свойство типа "Long".	LISTHEIGHT=50	Например, если есть два элемента управления DBQUERYBOX со значениями 50 и 100 соответственно для данного свойства, второй элемент управления будет в два раза выше, чем первый. Обратите внимание, что любое значение меньше 50 будет заменяться значением 50.
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"

Физические свойства элемента управления DBQUERYBOX, продолжение

Имя свойства=Значение	Описание свойства	Пример	Комментарий
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, поддерживает ли элемент управления множественный выбор (=TRUE) или нет (=FALSE). Свойство типа "Boolean".	MULTISEL=TRUE	
DBLCLICK=<TRUE|FALSE>	Если для данного свойства установлено значение TRUE, приложение Asset Manager будет имитировать нажатие кнопки **Далее** на текущей странице. Свойство типа "Boolean".	DBLCLICK=FALSE	

Физические свойства элемента управления DBQUERYBOX, продолжение

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 **aamapi94.ini** на каждом из серверов Asset Manager Web Service.
2. В разделе **[Option]** модифицируйте или добавьте следующую строку:

   ```
   /ACWeb/WizQueryBoxMaxLoad=1|<максимальное число записей>
   ```

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 т.д.).

Примечание. 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:

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.	См. раздел Элемент управления DBQUERYBOX
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.

Данный элемент управления не имеет никаких дополнительных свойств.

Элемент управления DBPATH

Элементом управления DBPATH создается элемент управления для ввода пути к базе данных Asset Manager.

Свойства

Кроме дополнительных свойств, общих для всех элементов управления, элемент управления DBPATH должен иметь следующее обязательное свойство:

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 базе данных Asset Manager.

Свойства

Кроме дополнительных свойств, общих для всех элементов управления, элемент управления LINKEDIT имеет следующие свойства:

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=<предложение WHERE запроса AQL>	Определяет фильтр AQL. Свойство типа "String".	Данное свойство не является обязательным.

Логическое свойство элемента управления LINKEDIT, продолжение

Имя свойства=Значение	Описание свойства	Пример/Комментарий
LINK=<имя SQL ссылки>"	Имя SQL ссылки из таблицы, определенной в свойстве TABLE. Дополнительное свойство	LINK="POrLIne"
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.	См. раздел Элемент управления DBQUERYBOX

Логическое свойство элемента управления LINKEDIT, продолжение

Имя свойства=Значение	Описание свойства	Пример/Комментарий
LINK.LABEL([iNameType])	Данным методом возвращается 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:

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:

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"
FORMAT	Определяет тип данных: • Длинное целое значение (long) • Double • Число (number) • Процент (percent)	
SERIE	Число для выбранной строки. Свойство CHART должно быть в интерактивном режиме (=TRUE).	
INDEX	Число для выбранного столбца. Свойство CHART должно быть в интерактивном режиме (=TRUE).	
CLICK	При щелчке на графике вызывается скрипт данного свойства.	

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 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).	
BACKIMAGE	Определяет путь изображения, используемого как фон для графика. Для отображения фонового изображения должно быть утверждено свойство BACKGROUND (=TRUE).	

Физические свойства элемента управления CHART, продолжение

Имя свойства=Значение	Описание свойства	Пример
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.

Свойства

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.

Свойства

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.

Свойства

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, определенным на панели управления используемой операционной системы. • RAWLONG • DOUBLE в формате, определенным на панели управления используемой операционной системы.	

Элемент управления COMBOEDIT

Данный элемент управления позволяет вставить элемент управления типа раскрывающегося списка.

Свойства

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.

Свойства

Свойства элемента управления DATETIMEEDIT

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

Примечание. В настоящее время эта функциональность в веб-клиенте недоступна.

В приложении Asset Manager можно создавать мастера с помощью встроенного графического редактора. Редактор предназначен для того, чтобы упростить и ускорить процесс создания мастера. Редактор не заменяет язык написания скриптов, понимание которого крайне важно для проектирования мастеров.

- Обзор интерфейса
- Создание нового узла
- Редактирование свойств узла
- Компиляция, выполнение и отладка мастера

Примечание. Чтобы использовать графический редактор, создаваемое или изменяемое действие должно быть действием типа "Мастер" (имя SQL: seActionType).

В этом разделе содержатся следующие сведения.

Обзор интерфейса 350
Создание нового узла 351
Редактирование свойств узла 351
Компиляция, выполнение и отладка мастера 352

Обзор интерфейса

Для доступа к графическому редактору выберите пункт меню навигации **Администрирование/ Действия**. 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.

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 приложение Asset Manager перемещает выделение на найденное вхождение (клавиша F3 и сочетание клавиш Shift+F3 позволяют найти следующее и предыдущее вхождения соответственно).

> **Примечание.** 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, приложение 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.

> **Примечание.** 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.

Приложение 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.

Примечание. Скрипт всегда имеет приоритет над фиксированным значением. Если назначить свойству как скрипт, так и значение, приложение 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 т.д.), приложение 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.

Примечание. При выборе в контекстном меню пункта **Восстановить значение по умолчанию** (после щелчка свойства правой кнопкой мыши), приложение 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.

Примечание. 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 возникновения проблем.

Пример создания мастера 353
Шаг 1 - Анализ ваших потребностей 353
Шаг 2 - Определение того, как будет организован мастер 355

Пример создания мастера

Назначение данного мастера - перемещение активов из одного местоположения в другое. Для этого необходимо:

- 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.

  Примечание. 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, заданными в шаге 1, необходимо определить организацию мастера. Т.е.:

1. число страниц;
2. связи между различными страницами.
3. Содержимое каждой из этих страниц.

Примечание. 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:

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

Страница	Назначение данной страницы	Используемый элемент управления
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 включает написание скрипта мастера. Для этого используйте описания структуры каждого из узлов мастера. Ниже показан закомментированный исходный код мастера "Перемещение". Данный код представляет собой только один из способов написания мастера. Существует много других способов написания мастера для выполнения этой же задачи.

```
;=================================================================
NAME = "Move"
TITLE = "Move user"
VERSION = "699"
;=================================================================================
====
;Ask which user to move. By default, use selection in amEmplDept if context is o
n this table
;=================================================================================
====
{ PAGE pgUser
  TITLE = "Choose the persons who are moving"
  { DBLISTBOX Users
    COLNAME = "Name|FirstName"
    COLWIDTH = "50|50"
    DBLCLICK = 1
    LABEL = "Persons to move"
    MULTISEL = 1
    TABLE = "amEmplDept"
    { VALUE =
      if [CurrentTable] = "amEmplDept" then
        RetVal = [CurrentSelection]
      else
        RetVal = ""
      end if
    }
    VISIBLE = 1
```

```
    }
    { TRANSITION trPersonToNewLoc
      TO = "pgNewLoc"
    }
  }

  ;=================================================================================
  ====
  ;Ask for new location
  ;=================================================================================
  ====
  { PAGE pgNewLoc
    TITLE = "Choose the new location"
    { STRING UserName
      VALUE = AmDbGetString("SELECT FirstName + ' ' + Name FROM amEmplDept WHERE l
  EmplDeptId 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 Li
  keParam(amEmplDept_2:FullName)+'%' AND amEmplDept_2:lEmplDeptId IN(" &
```

```
{pgUser.Users} & ")","|",",", "=")
  }
}

;=================================================================================
====
;Finish
;=================================================================================
====
{ FINISH FINISH
  { DO =
    On Error Goto ErrHandler
    Dim lErr as long

    dim hRecord as Long

    dim iEmplCount as Integer
    iEmplCount  = {pgRecap.Users.VALUES.Count()}
    dim iMax as Long
    iMax = iEmplCount

    dim lLocaId as long
    lLocaId = {pgNewLoc.NewLocId}

    lErr = amStartTransaction()

    dim i as Integer
    For i = 1 To iEmplCount
      lErr = AmProgress((100 * i ) / iMax)
      lErr = AmLog("Moving the employee" + {pgRecap.Users.VALUES(i,1)})
      hRecord = AmGetRecordFromMainId("amEmplDept", {pgRecap.Users.VALUES(i,0)}
)
      If hRecord <> 0  then
        lErr = AmSetFieldLongValue( hRecord, "lLocaId", lLocaId)
        lErr = AmUpdateRecord(hRecord)
        lErr = AmReleaseHandle(hRecord)
      End If
    Next i

    lErr = amCommit()

    RetVal = 0
    Exit Function


    ErrHandler:
      On Error Goto 0
      AmLog(AmLastError() & " - " & AmLastErrorMsg())
      AmLog("The transaction has been cancelled")
```

```
      RetVal = 1
      Exit function
  }
  SUMMARY = 1
}
```

Пример программирования мастера

Избегайте использования локальных переменных для общего доступа к данным

При программировании мастера не рекомендуется использовать локальные переменные для общего доступа к данным. Вместо этого рекомендуется использовать следующие подходы.

- Для общего доступа к данным на разных страницах, используйте глобальные переменные.
- Чтобы передать параметры между мастерами в одной цепочке выполнения, используйте структуру Params:

  ```
  { Params Params
    { Long lSummary
      Value = 0
    }
  }
  ```

  Пример структуры Params можно найти в мастере "sysFinBudgetClassCreate" или "sysFinBudgetCreate".

Эти рекомендации связаны с ограничениями в веб-клиенте. В клиенте Windows выполняется вычисление зависимых свойств, представленных на последующих страницах мастера. При этом в веб-клиенте такие зависимые свойства не вычисляются.

Следующий пример скрипта позволяет понять это ограничение.

```
Name = "Wizard"
Version = "10172"
{ Long global_long
  Value = 200
}

{ Page PageA
  { Long A_long
    Value = 100
  }
  { Label A_Label_1
    Caption = {A_long} + 1000
  }
```

```
  { Label A_Label_2
    Caption = {global_long} + 1000
  }
  { Label A_Label_3
    Caption = {A_Label_1.Caption} + 10
  }
  { Transition Transition1
    To = "PageB"
  }
}

{ Page PageB
  { OnEnter =
    Dim lErr as long
    lErr = AmSetProperty("PageA.A_long.Value", 300)
    lErr = AmSetProperty("global_long", 400)
    lErr = AmSetProperty("PageC.C_long", 500)
    AmMsgBox({PageA.A_Label_1.Caption})
    AmMsgBox({PageA.A_Label_2.Caption})
    AmMsgBox({PageA.A_Label_3.Caption})
  }
  { Label B_Label_1
    Caption = {PageA.A_long} + 1000
  }
  { Label B_Label_2
    Caption = {PageC.C_Label_1} + 20
  }
  { Transition Transition1
    To = "PageC"
  }
}

{ Page PageC
  { Long C_long
    Value = 0
  }
  { Label C_Label_1
    Caption = {PageC.C_long} + 1000
  }
}

{ Finish Finish
  { Do =
    AmMsgBox({PageB.B_Label_1.Caption})
  }
}
```

В приведенном выше скрипте определены три переменные long:

- A_long: переменная в узле PageA
- global_long: глобальная переменная
- C_long: переменная в узле PagPageC

В следующей таблице показаны зависимости свойств.

Свойства	Зависит от
PageA.A_Label_1.Caption	A_long
PageB.B_Label_1.Caption	
PageA.A_Label_2.Caption	global_long
PageA.A_Label_3.Caption	PageA.A_Label_1.Caption
PageB.B_Label_2.Caption	PageC.C_Label_1.Caption
PageC.C_Label_1.Caption	C_long

В узле PageB событие OnEnter изменяет значение C_long.

В клиенте Windows в связи с этим происходят следующие изменения.

- Изменение переменной C_long инициализирует вычисление PageC.C_Label_1.Caption, которое является зависимым свойством, определяемым на следующей странице (PageC).
- Изменение C_Label_1.Caption инициализирует вычисление PageB.B_Label_2.Caption.

Однако в веб-клиенте из-за существующего ограничения изменение переменной C_long не инициализирует вычисление C_Label_1.Caption. Таким образом, вычисление PageB.B_Label_2.Caption не выполняется.

Если изменить скрипты в PageB.B_Label_2.Caption на:

```
PageC.C_long}+20
```

тогда значение PageB.B_Label_2.Caption может вычисляться.

При повторном вычислении свойств предыдущей страницы поведение веб-клиента совпадает с поведением клиента Windows: зависимые свойства на предыдущей странице вычисляются.

Как и в предыдущем скрипте, событие OnEnter на странице PageB изменяет значение A_long. Следующие действия произойдут как в клиенте Windows, так и веб-клиенте:

- Изменение A_long инициализирует вычисление PageA.A_Label_1.Caption и PageB.B_Label.Caption.

- Изменение PageA.A_Label_1.Caption инициализирует вычисление PageA.A_Label_3.Caption.

Назначьте константы параметру COMOBOX

Рекомендуется явным образом назначить константы значениям параметров COMOBOX.

В следующем примере иллюстрируется эта рекомендация.

В мастере "sysSamEntitlement" элемент управления COMOBOX "cbType" в "pgComputers" имеет скрипты, связанные с его свойством "Значение":

```
Retval="No filter," & ComputerType()
```

Функция "ComputerType()" выглядит следующим образом:

```
  Function ComputerType() as String
  Dim strBuf as String
  Dim strTmp as String
  Dim strStep as String
  Dim strValues as String
  Dim strNature(20) As String
  Dim i as Long
  strBuf = AmGetFieldFormat(AmGetFieldFromName(AmGetTableFromName("amNature"), "
seComputerType"))
  strStep=""
  i=0
  Do While strBuf<>""
       i=i+1
       strNature(i) = ExtractValue(strBuf, "|")
       strTmp       = ExtractValue(strBuf, "|")
       strValues    = strValues & strStep & strNature(i)
       if strValues<>"" Then strStep = ","
       loop
  ComputerType=strValues
  End Function
```

Функция "seComputerType" разбирается до строки вида "xxx|Domain|Virtual Machine|Mobile Device|xxx"; а значение COMOBOX представляет собой строку вида "No Filter|xxx|Domain|Virtual Machine|Mobile Device|xxx".

На самом деле "No Filter" — это только текст подсказки, который не является одним из параметров в COMOBOX; другие части строки — это параметры проверки для пользователей, но они не имеют соответствующих констант.

Константы очень важны, так как они указывают системе мастера, какой параметр выбирается пользователем для дальнейших процессов.

Код C++ автоматически генерирует идентификаторы для параметров COMOBOX, а также идентификатор для "No Filter". Однако текст подсказки не должен иметь идентификатор проверки, поскольку он не является параметром проверки COMOBOX. Кроме того, автоматическое генерирование идентификаторов приводит к их дублированию. Например,

если скрипты мастера назначают для COMOBOX значение "No Filter|xxx=1|Domain=2|xxx=3". Для "No Filter" автоматически генерируется идентификатор, и получается значение "No Filter=1|xxx=1|Domain=2|xxx=3". В данном случае система мастера не может определить, какой параметр выбран, если пользователь выбирает 1.

Не сравнивайте CHECKBOX с целым числом

Не рекомендуется сравнивать значение элемента управления CHECKBOX с целым числом, поскольку для CHECKBOX используется логический тип данных.

Безопаснее использовать следующий скрипт:

```
if {CheckBoxTest.Value} = false then AmMsgBox("CheckBoxTest OK")
```

Тем не менее, если требуется выполнить сравнение с целым числом, не рекомендуется использовать следующий скрипт:

```
if {CheckBoxTest.Value} = 1 then AmMsgBox("CheckBoxTest equals to 1 OK")
```

Рекомендуется использовать следующий скрипт:

```
if {CheckBoxTest.Value} <> 0 then AmMsgBox("CheckBoxTest equals to !0 OK")
```

Данная рекомендация объясняется следующими причинами:

Целое значение CHECKBOX может быть 0 или 1; однако инфраструктура скриптов Asset Manager может иметь другие варианты использования выражения true/false, например:

```
Dim a as integer
a = ( x = y)
```

В этом случае "a" может иметь значение 0 или -1 .

Связанный мастер не запускается на уровне строки кода

Параметр "amWizChain" не вызывает связанный мастер немедленно. Связанный мастер вызывается только после выполнения всего скрипта.

Например, если выбрать мастер **Жизненный цикл актива > Управление активами ПО > Действие пользователя > Создание новой лицензии на ПО**, цепочка мастера запускается в узле "START". Скрипты в узле "START" выглядят следующим образом:

```
Dim lErr as Long,hq As long
lErr=amSetProperty("lCntrId",CLng(AmGetUserEnvSessionItem ("SAM", "SAMCurrentCon
tract")))
If {lCntrId} = 0 Then
  lErr=amWizChain("sysSamGetEnv")
  lErr=amSetProperty("lCntrId",CLng(AmGetUserEnvSessionItem ("SAM", "SAMCurrentC
ontract")))
End if
if {lCntrId} > 0 Then
  hq=amQueryCreate()
```

```
  lErr = AmQueryExec(hq, "SELECT cf_Self, lCostId, lCostCatId FROM amContract WH
ERE lCntrId = " & {lCntrId})
  lErr=amSetProperty("strCntrSelf",AmGetFieldStrValue (hq, 0))
  lErr=amSetProperty("pgFinanceInfo.lkeCostCenter",AmGetFieldLongValue(hq, 1))
  lErr=amSetProperty("pgFinanceInfo.lkeCostCategory",AmGetFieldLongValue(hq, 2))
  AmReleaseHandle(hq)


  lErr=amSetProperty("strPossibleModels",amDbGetListEx("SELECT lModelId FROM amA
sset WHERE lLicCntrId=" & {lCntrId} , "", ",", ""))
End if
RetVal="pgSelection"
```

В этом примере связанный мастер фактически запускается после события "START".

Веб-клиент не совместим с автоматическим обновлением

Не ссылайтесь явно на свойство "CAPTION".

Например, на странице "pgFinanceInfo" мастера **Создание новой лицензии на ПО** ("sysSamCreateLicpfi") объект "tr" в разделе "TRANSITION" имеет условие, содержащее следующие скрипты:

```
IF {lkeMaintContract.Value} > 0 AND ({lkeModel} > 0 OR {dtStart} <> "" OR {dtEn
d} <> "" OR {numCost} > 0) THEN
  RetVal = AmSetProperty("pgFinanceInfo.lbError.Caption","<P><HR><CENTER><FONT C
OLOR='RED'><B>" & FormatResString("Please either select an existing contract $1o
r$2 create a new one","<U>","</U>") & "<HR>")
  AmMsgBox(FormatResString("Please either select an existing contract $1or$2 cre
ate a new one","",""))
  RetVal = FALSE
ELSE
  RetVal = AmSetProperty("pgFinanceInfo.lbError","")
  RetVal = TRUE
END IF
```

Следующий скрипт не вызывает ошибок или предупреждений во время выполнения, однако скрипт является недопустимым в C++:

```
RetVal = AmSetProperty("pgFinanceInfo.lbError","")
```

Целью скриптов является установка для параметра pgFinanceInfo.lbError.caption строки NULL. C++ выполняет следующее:

1. Создает и инициализирует строку для "pgFinanceInfo.lbError.Caption" со строкой NULL.
2. Пытается найти свойство "pgFinanceInfo.lbError" и задать для него строку NULL.

Во втором шаге C++ не может найти свойство с именем "pgFinanceInfo.lbError" и готов отобразить ошибку. Однако код обнаруживает, что новое значение равно NULL и старое (исходное) значение также равно NULL; поэтому C++ не выдает ошибку.

Если изменить скрипт на:

```
RetVal = AmSetProperty("pgFinanceInfo.lbError","text")
```

будет получена ошибка примерно следующего содержания:

```
Error (12,001): Property 'lbError' unknown in object 'pgFinanceInfo' of the wiza
rd. ('Line 6 of script 'pgFinanceInfo.tr.Condition, line 389'')
```

```
Unable to complete operation in current state. ('Line 6 of script 'pgFinanceInfo
.tr.Condition, line 389'')
```

Чтобы избежать этой ошибки, измените скрипт, чтобы он явно ссылался на свойство "CAPTION":

```
RetVal = AmSetProperty("pgFinanceInfo.lbError.caption","text")
```

Примечание. Для мастера недопустимо явно ссылаться на свойство CAPTION, тем не менее, допустимо явно ссылаться на свойство "VALUE" в элементах управления (например, "DBLISTBOX").

Не используйте двойной источник для свойства VALUE

Не применяйте несколько источников для свойства VALUE. Применение нескольких источников для свойства VALUE может привести к следующим проблемам.

- Если свойство VALUE может быть затронуто в нескольких местах, невозможно предсказать порядок влияния на свойство.
- Автоматическое обновление, которое предусмотрено в клиенте Windows, в мастере веб-клиента не работает.

Часто задаваемые вопросы

Назначение данной главы - дать ответы на вопросы, которые, скорее всего, возникнут при создании мастера.

Не работает {lbxMyListBox.Values.Count} 365
Не работает {lbxMyListBox.Line(lRow)} 365
Не работает {lbxMyListBox.Values.Line({lbxTmp})} 365
Не работает назначение фиксированного значения свойству 365
Выполнение мастера, создающего актив в базе данных, приводит к ошибке 366
При выполнении мастера отображается неполное сообщение об ошибке 366

Если на странице мастера используется управляющий элемент DBLISTBOX, наблюдается снижение производительности 366

Как разрешить или запретить изменение некоторых столбцов в элементе управления LISTBOX? 367

Как сделать так, чтобы мастером открывался подробный экран? 367

В чем разница между свойствами COLNAME и COLTITLE элемента управления LISTBOX? 367

Слово "query" нельзя использовать в элементах мастера 368

Не работает {lbxMyListBox.Values.Count}

Ответ

Необходимо ввести в синтаксис метода открывающую и закрывающую круглые скобки. Вот исправленный код:

```
{lbxMyListBox.Values.Count()}
```

Не работает {lbxMyListBox.Line(lRow)}

Ответ

Метод LINE связан со свойством VALUES элемента управления LISTBOX. Вот исправленный код:

```
{lbxMyListBox.Values.Line(lRow)}
```

Не работает {lbxMyListBox.Values.Line({lbxTmp})}

Ответ

В методе нельзя использовать ссылочное свойство. В данном случае необходимо записать:

```
Dim lRow As Long lRow = {lbxTmp} {lbxMyListBox.Values.Line(lRow)}
```

Не работает назначение фиксированного значения свойству

Например:

```
{Property} = 123
```

Ответ

Чтобы присвоить свойству какое-либо значение, необходимо использовать функцию AmSetProperty(), как показано ниже:

```
Dim irc as Integer irc= AmSetProperty("Proprerty", 123)
```

Примечание. Необходимо обязательно восстановить возвращаемый код (в данном примере - irc), даже если его не нужно использовать.

Выполнение мастера, создающего актив в базе данных, приводит к ошибке

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).

При выполнении мастера отображается неполное сообщение об ошибке

Ответ

Нажмите сочетание клавиш SHIFT+F9, чтобы отобразить отладчик. Сообщения об ошибке в окне истории часто более подробны.

Если на странице мастера используется управляющий элемент DBLISTBOX, наблюдается снижение производительности

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.

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"
```

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)")
```

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.

Слово "query" нельзя использовать в элементах мастера

Ответ

При использовании слова "query" в элемента мастера происходит непредвиденная ошибка. Поскольку слово "query" зарезервировано для использования в скриптах мастера, использование этого слова в любых элементах мастера, например, в заголовке мастера, приведет к ошибке.

Глава 15: 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.

Глава включает следующее:

Определение позиции новостей 369

Обзор новостей 369

Важность позиций новостей 370

Сообщение для рассылки 370

Список рассылки новостей 370

Отображение новостей 371

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 2011 г."

Обзор новостей

В этом разделе содержатся следующие сведения.

Создание позиции новостей 369

Чтение новостей 370

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.

> **Примечание.** Любой администратор службы поддержки, руководитель группы сотрудников и администратор Asset Manager будут видеть позиции новостей при условии, что они зарегистрированы в списке рассылки новостей.

В этом разделе содержатся следующие сведения.

Параметр Все группы сотрудников (SQL: bAllGroups) 370
Параметр Включить подгруппы (SLQ bChildGrps) 371

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 (SLQ 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.

См. раздел "Активизация бегущей строки новостей" ниже.

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.

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.

Благодарим за отзыв!

Если у вас есть комментарии к данному документу, обратитесь в отдел документации по электронной почте. Если на вашем компьютере настроен клиент электронной почты, при нажатии на ссылку выше откроется окно нового сообщения, в теме которого будет указана следующая информация:

Отзыв о Asset Manager, 9.40 Расширенное использование

Напишите в сообщении свой отзыв и отправьте его нам.

Если клиент электронной почты не настроен, скопируйте приведенную выше информацию в окно нового сообщения вручную, а затем отправьте свой отзыв по адресу ovdoc-ITSM@hp.com.