

Глава 2

Инструменты среды разработки MorphX

В этой главе

- Введение
- Дерево прикладных объектов
- Проекты
- Окно свойств
- Редактор кода X++
- Редактор меток
- Компилятор
- Понимание рекомендаций
- Отладчик
- Обратная разработка
- Обзорщик таблиц
- Поиск по АОТ
- Сравнение прикладных элементов
- Перекрестные ссылки
- Контроль версий

Введение

Microsoft Dynamics AX содержит набор инструментов для разработки, среди которых присутствует MorphX – среда разработки, которую используют для создания и изменения элементов бизнес логики Microsoft Dynamics AX. Набор элементов бизнес логики использует модель элементов приложения, описанную в главе 1 «Обзор архитектуры». MorphX позволяет создавать, просматривать, изменять и удалять прикладные элементы модели приложения, которые содержат метаданные, структуры (упорядоченных по иерархии элементов), свойства в виде пары ключ-значение и код X++. Например, прикладной элемент Таблица состоит из названия таблицы, набора свойств, полей, индексов, отношений, методов и т. д. В этой главе

описываются наиболее часто используемые инструменты и излагаются советы и приемы работы с ними.

Вы можете найти дополнительную информацию и обзор других инструментов MorphX в разделе **MorphX Development Tools** платформы **Microsoft Dynamics AX, Software Development Kit (SDK) 2012** на MSDN.



Совет. Для того чтобы перейти в рабочую область (Development Workspace) разработки **Microsoft Dynamics AX 2012**, которая содержит все инструменты разработки, нажмите **Ctrl+Shift+W**.

Табл. 2-1 содержит список инструментов MorphX, которые описываются в этой главе.

Табл. 2-1. Инструменты MorphX, используемые в разработке

Инструмент	Использование
Дерево прикладных объектов (АОТ)	Главная точка входа для разработки и выполнения большинства операций. Позволяет просматривать репозиторий всех прикладных элементов системы, которые входят в бизнес приложение. АОТ используется для просмотра свойств прикладного элемента, а также для модификации и создания новых элементов
Проекты	Группировка логически связанных прикладных элементов в проекте
Окно свойств	Просмотр и изменение свойств прикладного элемента. Окно свойств показывает свойства прикладных элементов как пары ключ-значение
Редактор кода X++	Просмотр и написание кода X++
Редактор меток	Просмотр и редактирование локализуемых строк, называемых метками
Компилятор	Компиляция кода X++ в исполняемый формат
Best Practices	Автоматический поиск дефектов в коде и определение прикладных элементов
Отладчик	Поиск ошибок в коде X++
Обратная разработка	Генерация диаграмм из выбранных прикладных элементов в Microsoft Visio Unified Modeling Language (UML) и Entity Relationship Diagrams (ERDs)

Табл. 2-1. Инструменты MorphX, используемые в разработке (окончание)

Инструмент	Использование
Обозреватель таблиц	Просмотр содержимого таблиц базы данных непосредственно с прикладных объектов таблиц
Обозреватель иерархии и контекста типа Иерархии	Просмотр иерархии и определение типа иерархии текущего активного прикладного элемента
Поиск	Поиск примеров исходного кода и метаданных в АОТ
Сравнение	Построчное сравнение двух версий кода одного и того же прикладного элемента
Перекрестные ссылки	Определение места использования прикладного элемента
Контроль версий	Отслеживание изменений прикладных элементов и просмотр истории изменений прикладных элементов

К этим инструментам можно получить доступ из:

- рабочей зоны разработчика меню Инструменты;
- контекстного меню прикладного элемента.

Можно изменить настройку некоторых инструментов MorphX, щелкнув Параметры в меню Сервис.

Рис. 2-1 показывает форму Параметры.

Дерево прикладных объектов

АОТ – это главная точка входа в среду разработки MorphX, а также обзор репозитория прикладных объектов. Открыть АОТ можно, щелкнув на пиктограмме АОТ в инструментах или нажав Ctrl+D. Пиктограмма АОТ выглядит так, как показано на следующем рисунке.



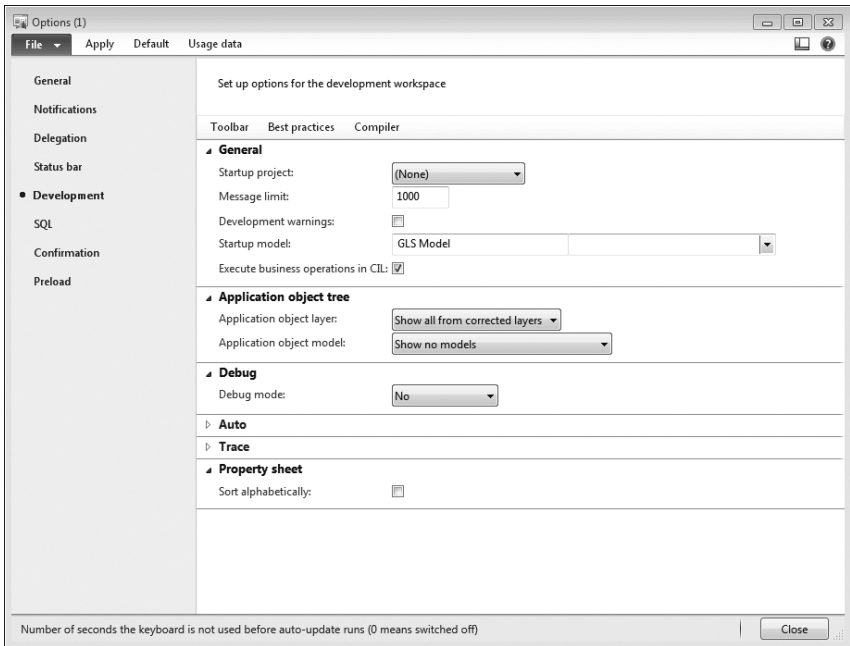


Рис. 2-1. Форма Параметры, где производятся настройки инструментов разработки

Навигация по дереву прикладных объектов

Дерево объектов приложения (АОТ) имеет структуру упорядоченной иерархии прикладных элементов. Корень дерева АОТ содержит узлы категорий прикладных элементов, такие как Classes (Классы), Tables (Таблицы) и Forms (Экранные формы). Некоторые элементы группируются в подкатегории для упрощения просмотра и перемещения по дереву репозитория. Например, Tables, Maps, Views и Extended Data Types располагаются под элементом Data Dictionary, а все веб-элементы, имеющие отношение к вебу, располагаются в узле Web. На рис. 2-2 показан АОТ.

По дереву объектов АОТ можно перемещаться с помощью стрелок на клавиатуре. Нажмите стрелку вправо, и узел, который содержит подчиненные прикладные элементы, раскроется.

Прикладные элементы отсортированы по алфавиту. У родительского прикладного элемента могут существовать тысячи подчиненных прикладных элементов. Относитесь внимательно к именованию объектов при-

ложения для того, чтобы перемещение по дереву объектов и поиск были эффективными.

Все названия элементов в АОТ используют следующий алгоритм именования:

<Бизнес часть названия элемента> + <Функциональная часть названия элемента> + <Функциональность, выполняемое действие или тип контента>

Согласно этому алгоритму именования, подобные элементы находятся рядом. Бизнес-область названия элемента часто называют *префикс*. Префиксы обычно используют для обозначения группы элементов.

Например, элемент *VendPaymReconciliationImport* имеет префикс *Vend*, который является сокращением названия бизнес-области (*Vendor*), *Paym-Reconciliation* реализует функциональную область (оплата по выверке), а списки *Import* предполагают действие (импорт). Название *CustPaymReconciliationImport* описывает функциональную область и действие для клиента.

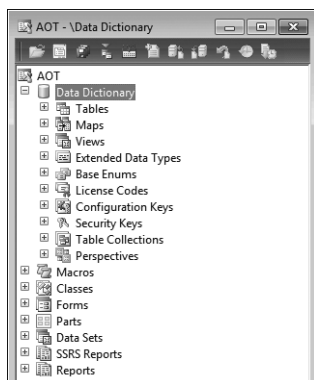


Рис. 2-2. АОТ



Совет. При создании прикладных элементов в дополнение к конвенции по именованию следует добавить еще один префикс, который уникально идентифицирует ваше решение. Этот дополнительный префикс поможет предотвратить конфликт имен, если ваше решение будет работать с решениями из других источников. Рассматривайте префикс как идентификатор компании и решения. Например, если компания с именем *MyCorp* разрабатывает

решение по начислению заработной платы, то следует использовать префикс *McPR* у всех новых элементов.

В табл. 2-2 содержатся наиболее распространенные префиксы и их описание.

Табл. 2-2. Префиксы

Префикс	Описание
<i>Ax</i>	Типизированный источник данных Microsoft Dynamics AX
<i>Axd</i>	Бизнес-документ Microsoft Dynamics AX
<i>Asset</i>	Основные средства
<i>BOM</i>	Спецификация
<i>COS</i>	Учет затрат
<i>Cust</i>	Покупатель
<i>Dir</i>	Глобальная адресная книга
<i>EcoRes</i>	Экономические ресурсы
<i>HRM/HCM</i>	Управление персоналом
<i>Invent</i>	Управление запасами
<i>JMG</i>	Управление цехом
<i>KM</i>	Управление знаниями
<i>Ledger</i>	Главная книга
<i>PBA</i>	Конфигуратор товаров
<i>Prod</i>	Производство
<i>Proj</i>	Проект
<i>Purch</i>	Закупки
<i>Req</i>	Потребности
<i>Sales</i>	Продажи
<i>SMA</i>	Сервисное обслуживание
<i>SMM</i>	Управление продажами и маркетингом, управление взаимоотношениями с клиентами (CRM)
<i>Sys</i>	Прикладные инфраструктуры и средства разработки
<i>Tax</i>	Налоги

Табл. 2-2. Префиксы (окончание)

Префикс	Описание
<i>Vend</i>	Поставщик
<i>Web</i>	Инфраструктура корпоративного портала
<i>WMS</i>	Управление складом



Совет. При создании нового элемента, проверяйте, что вы следуйте соглашению о наименовании. Дальнейшая разработка и поддержка будет проще.

Дизайнер проектов, который описывается далее в этой главе, предоставляет альтернативное представление прикладных элементов в АОТ.

Создание прикладного элемента в АОТ

Для создания нового элемента в АОТ в контекстном меню выберите Новый <Тип Элемента>, как показано на рис. 2-3.

При создании элемента ему автоматически присваивается имя по умолчанию. Однако заданное по умолчанию название следует заменить на новое, которое соответствует описанному выше соглашению имен.

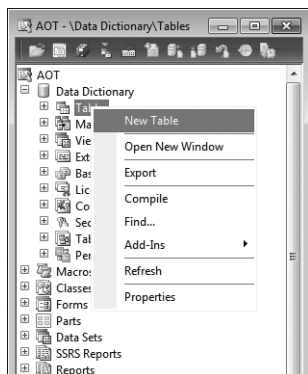


Рис. 2-3. Создание нового элемента в АОТ

Изменение прикладного элемента в АОТ

Прикладной элемент АОТ имеет набор свойств, характерных для данной категории элементов, а также либо подузлы, либо исходный код X++. Для

просмотра и редактирования свойств прикладного элемента используется Окно свойств прикладного элемента (рис. 2-9), а для просмотра и редактирования кода X++ используется редактор исходного кода X++ (рис. 2-11).

Порядок расположения дочерних узлов играет определенную роль в семантике прикладного элемента. К примеру, вкладки на экранной форме отображаются в том порядке, в котором они перечислены в АОТ. Для изменения порядка узлов в АОТ необходимо выбрать требуемый узел и, удерживая нажатой клавишу Alt, нажатиями клавиш со стрелкой вверх или вниз перемещать текущий узел вверх или вниз соответственно.

Вертикальная красная линия под измененным элементом будет показывать признак, что элемент изменен и не сохранен, так называемый черновик элемента (такие элементы называют «грязными»), как показано на рис. 2-4.

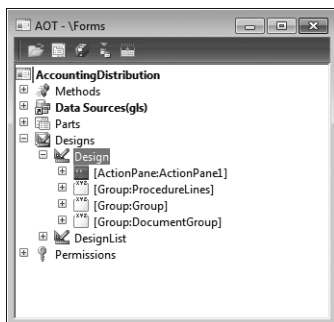


Рис. 2-4. «Грязный» прикладной элемент в АОТ, отмеченный красной вертикальной линией возле *AccountingDistribution*

Черновик элемента сохраняется в следующих ситуациях.

- Элемент запускается на выполнение.
- Разработчик явно указывает Сохранить или Сохранить все.
- Может происходить автоматическое сохранение. Этот определяется в форме Параметры меню Сервис.

Обновление прикладного элемента АОТ

Если несколько разработчиков одновременно редактируют один и тот же элемент в среде Microsoft Dynamics AX, то модификации каждого могут быть не синхронизированы с последней версией элемента. Чтобы локальные версии изменяемого элемента обновлялись, необходимо установить

признак автообновления в фон. Функция автообновления в конечном итоге применяет все изменения, но, возможно, вы захотите самостоятельно произвести обновление прикладного элемента. Для этого в контекстном меню выбранного элемента щелкните на Восстановить. Это действие обновляет как данные на диске, так и в оперативной памяти, версии элемента. Как правило, общая целостность того, что показывается в АОТ, управляется автоматически, но некоторые операции, такие как синхронизация с приложением базы данных или переустановка приложения, могут привести к несоответствию, которое требует ручного разрешения. Для того чтобы гарантированно использовать последние версии прикладных элементов, выполните следующие действия.

1. Закройте клиент Microsoft Dynamics AX, чтобы очистить элементы в памяти.
2. Остановите службу Application Object Server (AOS) Microsoft Dynamics на сервере, чтобы очистить элементы из памяти.
3. Удалите файлы кэша (*. AUC) элементов приложения в папке **Application Data**, расположенной в "%LocalAppData%", чтобы удалить элементы на диске.

Действия, доступные с прикладными элементами АОТ

Каждый узел в АОТ содержит набор доступных для этого узла действий. Вы можете получить доступ к этим действиям из контекстного меню, которое можно открыть, щелкнув правой кнопкой мыши на любом узле.

Вот два факта, которые следует помнить о действиях.

- Доступные действия зависят от типа выбранного узла АОТ.
- Можно выбрать несколько узлов и выполнять действия одновременно на всех выбранных узлах.

Часто используемое действие Открыть в новом окне, доступное для всех категорий узлов АОТ, приведет к открытию нового окна дерева АОТ с текущим узлом в качестве корневого. Это действие используется для захвата экрана. На рис. 2-4 показан элемент *AccountingDistribution*. После открытия нового окна АОТ вы можете перетаскивать элементы из одного окна в узлы другого окна, экономя время при разработке своих элементов.

Вы можете расширить список доступных действий контекстного меню. Возможно создание новых действий для любого прикладного эле-

мента в АОТ с использованием только встроенных возможностей MorphX. На самом деле все действия, перечисленные в подменю Настройки (Add-Ins), реализованы с помощью инструментов MorphX и языка X++.

Вы можете назначить класс в качестве новой надстройки, выполнив приведенные ниже шаги.

1. Создайте новый пункт меню и дайте ему содержательное имя, метку и текст справки.
2. Укажите свойство *Object Type* в значение **Class**.
3. Укажите свойству *Object* имя класса, который будет вызываться в оснастке Настройка.
4. Перетащите пункт меню в *SysContextMenu*.
5. Если создаваемая надстройка должна быть доступна только для определенных категорий узлов АОТ, то необходимо соответствующим образом изменить код метода *verifyItem* в классе *SysContextMenu*.

Слои и модели в АОТ

Когда вы модифицируете прикладной элемент, принадлежащий более низкому слою, копия элемента помещается в текущий слой текущей модели. Все элементы в текущем слое выделены жирным шрифтом (как показано на рис. 2-5), что делает легко распознаваемыми модифицированные прикладные элементы. За дополнительной информацией по технологии слоев обращайтесь к разделу «Слои» главы 21.

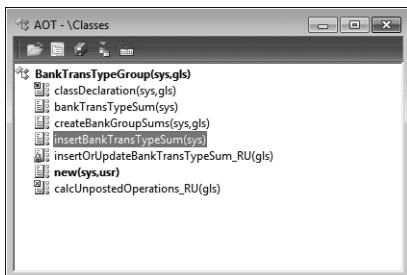


Рис. 2-5. Пример элемента в АОТ, который существует на нескольких слоях

Можно использовать настройки Слой Объекта Приложения или Модель Объекта Приложения в Параметрах меню Сервис для отображения более детальной информации по прикладным элементам АОТ (см. рис. 2-1). Рис. 2-5 показывает класс с включенной опцией Все слои. Видно, что каждый метод имеет суффикс с информацией о слое – это SYS, VAR и USR.

Если элемент присутствует в нескольких слоях, то для того чтобы получить доступ к версиям элемента с более низких слоев, необходимо щелкнуть на нем правой кнопкой мыши и выбрать пункт меню Слои.

Настоятельно рекомендуется включить опцию Все слои после обновления для визуального представления произведенных изменений в АОТ.

Проекты

Проекты используются для создания настраиваемого пользовательского представления прикладных элементов. В рамках проекта можно группировать и структурировать прикладные элементы согласно вашим потребностям. Проект – это мощная альтернатива работе с АОТ, поскольку все элементы, требуемые для создания некоторой функциональности, можно собрать в один проект.

Создание проекта

Проекты открываются по щелчку на значке проектов на панели инструментов. Рис. 2-6 показывает внешний вид окна Проекты, содержащий подзлы Частные (*Private*) и Общие (*Shared*).

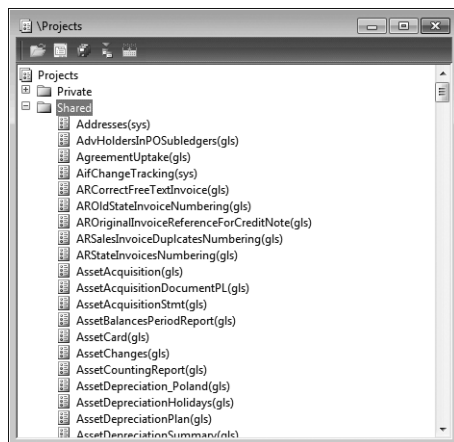


Рис. 2-6. Представлено окно Проекты со списком общих проектов

За исключением структуры, функция проекта аналогична функции АОТ. Каждый элемент проекта представлен в дереве АОТ. Когда вы создаете новый проект, вам необходимо решить, должен он быть частными или общими для всех разработчиков. Установить особые права доступа на общие проекты, ограничив доступ некоторым разработчикам, невозмож-

но. Вы можете сделать общий проект частным (и частный проект общим), перетащив его из общей категории в частную.



Примечание. Главной особенностью Microsoft Dynamics AX 2012 является агрегирование элементов в общих проектах с целью обеспечения возможности просмотра всех задействованных прикладных элементов. В стандартной поставке нет частных проектов. Можно определить проект, который будет открываться при запуске клиента Microsoft Dynamics AX, настройка определяется в форме Параметров каждого пользователя.

Автоматически сгенерированные проекты

Для упрощения работы с проектами существует функциональность автоматической генерации их содержимого, от использования групповых масок до создания новых специализированных типов проектов. В следующем разделе рассматриваются различные подходы по автоматическому генерированию проектов.

Маска по группе

Группы представляют собой папки в проекте. При создании группы вы можете наполнять ее содержимым автоматически, просто выставив свойство *ProjectGroupType* (Все – это опция) и выбрав регулярное выражение в свойстве *GroupMask*. Наполнение группы произойдет автоматически и будет обновляться при создании элементов, удалении или переименовании прикладных элементов. Использование группы Маски по группе гарантирует, что ваш проект будет всегда актуальным, даже если элементы создаются непосредственно в АОТ, а не через проект.

Рис. 2-7 показывает свойство *ProjectGroupType*, установленное в значение *Classes* и свойство *GroupMask*, установленное в значение *ReleaseUpdate* в группе проекта. Все классы с именем, содержащим *ReleaseUpdate* (префикс для скриптов, обновляющих данные), будут включены в группу проекта.

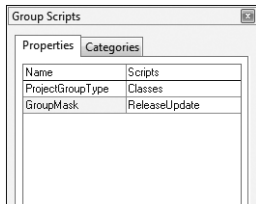


Рис. 2-7. Окно свойств с заданными значениями свойств *ProjectGroupType* и *GroupMask*

Рис. 2-8 показывает результирующий проект после применения настроек в проекте на рис. 2-7.

Фильтры

Вы можете также сгенерировать узлы проекта на основе запроса по АОТ. Так как все элементы АОТ хранятся в базе данных, вы можете использовать запрос для фильтрации элементов, а результаты будут представлены в проекте.

Чтобы создать проект на основании запроса по АОТ, щелкните кнопку Расширенный фильтр/сортировка на панели инструментов проекта. В зависимости от сложности запроса, генерация проекта может занять несколько минут. С помощью фильтрации можно создать проект, содержащий следующие категории элементов:

- прикладные элементы, созданные или измененные в течение последнего месяца;
- прикладные элементы, созданные или модифицированные определенным пользователем;
- прикладные элементы из определенного слоя.

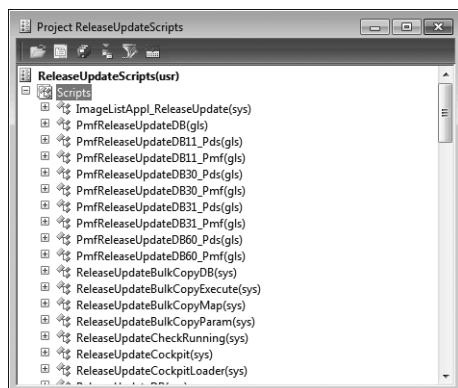


Рис. 2-8. Проект, созданный с помощью Маски по группе

Инструменты разработчика

Некоторые инструменты разработчика, такие как Мастер мастеров, формируют проекты, содержащие созданные мастером прикладные элементы. После запуска Мастера мастеров (Wizard Wizard) создается новый проект, содержащий экранную форму, класс и пункт меню – все элементы, которые нужны для запуска созданного мастера.

Вы можете также использовать ряд других мастеров, такие как AIF Document Service Wizard и Мастер классов, для создания проекта. Мастера доступны в меню Сервис Мастера.

Сравнение по слоям. Новый проект может также быть создан на основании сравнения всех прикладных элементов одного слоя с прикладными элементами другого слоя, который называется «ссылочным». Если прикладной элемент существует в обоих слоях и определения элемента различаются или элемент не существует в ссылочном слое, то он будет добавлен к результирующему проекту. Вы можете запустить функциональность сравнения слоев, нажав Сервис > Обновление кода > Сравнение по слоям.

Обновление кода. При обновлении приложения с одной версии системы Microsoft Dynamics AX на другую или установке пакета обновления вам придется иметь дело с новыми элементами, которые добавляются и существующими элементами, которые изменяются. Данные изменения могут конфликтовать с прикладными элементами, модифицированными на более высоком слое.

Проект Обновления делает трехстороннее сравнение для обнаружения любых конфликтов обновления кода с существующими прикладными элементами. Он сравнивает оригинальную версию с модифицированной и обновленной версиями. Если обнаруживается конфликт, то элемент добавляется в проект. Результирующий проект содержит список элементов обновления с обнаруженными конфликтами между версиями. Вы можете использовать инструмент для сравнения, описанный далее в этой главе, чтобы увидеть конфликты в каждом прикладном элементе. В совокупности эти функции обеспечивают экономически эффективный инструмент для обновления.

Для получения дополнительной информации об обновлении кода см.: «Microsoft Dynamics AX 2012 White Papers: Code Upgrade» по ссылке <http://www.microsoft.com/download/en/details.aspx?id=20864>.

Для создания проекта обновления выберите Сервис > Обновление кода > Обнаружение конфликтов обновления кода.

Типы проектов

Когда вы создаете новый проект, то можете указать его тип. До сих пор в этой главе мы обсуждали стандартный тип проектов. Тестовый проект, используемый для группировки набора классов для тестирования, является еще одним специальным типом проекта, существующим в Microsoft Dynamics

АХ. Для создания собственного типа проекта создайте новый класс, наследуемый от *ProjectNode*. В созданном специальном типе проекта вы можете управлять структурой, пиктограммами и другими доступными действиями.

Окно свойств

Свойства являются важной частью системы метаданных. Каждое свойство представляет собой пару типа ключ-значение. Вы можете использовать свойства для проверки и изменения свойства прикладных элементов.

При открытии рабочей области разработки оснастка свойств открывается по умолчанию. Если закрыть ее, то вы можете открыть ее снова в любое время, нажав **Alt + Enter** или кнопку Свойства на панели инструментов рабочей области Разработки. Окно свойств автоматически обновляется при выборе или переходе на другой прикладной элемент АОТ. Поэтому нет необходимости заново открывать окно свойств для каждого прикладного элемента; вместо этого можно просто оставить его открытым, и при выборе нового прикладного элемента его свойства будут отображены в окне свойств автоматически. Рис. 2-9 показывает лист свойств класса *TaxSpec*. Два столбца, отображаемые в окне свойств, представляют собой ключ и значение ключа для каждого свойства прикладного элемента.

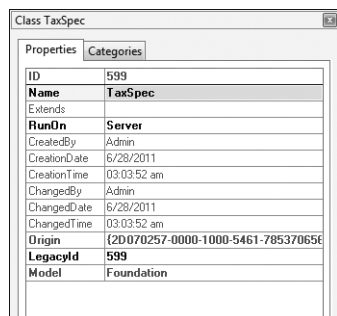


Рис. 2-9. Лист свойств элемента в дереве АОТ



Совет. Нажав клавишу **Esc** на листе свойств, вы установите фокус на предыдущем выбранном элементе.

Рис. 2-10 показывает вкладку *Categories* для класса, выбранного на рис. 2-9. Здесь выбраны логически связанные свойства прикладного элемента по категориям. Для элементов с большим количеством свойств данное представление облегчает поиск нужного свойства.

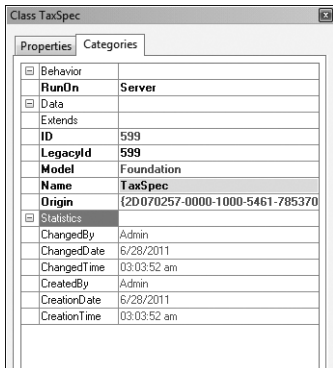


Рис. 2-10. Вкладка Categories окна свойств для прикладного элемента AOT

Свойства только для чтения выделены серым цветом. Так же как файлы в файловой системе, элементы содержат информацию создателя и времени изменения. Все элементы, пришедшие от Microsoft, имеют одно и то же время и одного и того же пользователя.

Порядок сортировки по умолчанию размещает связанные свойства друг за другом. Категории были введены в более раннюю версию Microsoft Dynamics AX для облегчения поиска свойств. Вы можете поменять порядок сортировки по алфавиту в форме Параметры.

Можно закрепить панель свойств по одной из сторон экрана, щелкнув правой кнопкой мыши в строке заголовка. Благодаря стыковке с краями экрана, окно свойств всегда располагается поверх остальных окон приложения.

Редактор кода X++

Весь код X++ пишется только в редакторе кода X++. Открыть редактор можно, выбрав определенный узел в дереве AOT и нажав клавишу Enter. Окно редактора состоит из двух частей. Левая панель показывает доступные методы, а правая – сам код X++, выбранный в левой части метода, как показано на рис. 2-11.

Редактор кода X++ – это обычный текстовый редактор, который поддерживает синтаксическую подсветку и *IntelliSense*.

Горячие клавиши

Для навигации и редактирования кода в редакторе X++ используются стандартные сочетания клавиш, описанные в табл. 2-3.



Рис. 2-11. Редактор кода X++

В редакции Microsoft Dynamics AX 2012 некоторые сочетания клавиш отличаются от более ранних версий для совместимости с интегрированной средой разработки (IDE) Microsoft Visual Studio.

Табл. 2-3. Горячие клавиши редактора кода X++

Действие	Горячая клавиша	Описание
Справка	F1	Открывает контекстно-зависимую Справку по типу или методу, выбранному в редакторе
Переход к следующему сообщению об ошибке	F4	Открывает редактор и позиционирует курсор на следующей ошибке компиляции на основе содержимого окна компилятора
Исполнение текущего элемента	F5	Запуск на выполнение текущей формы, задания или класса
Компиляция	F7	Компиляция кода текущего метода
Установка точки останова	F9	Установка или удаление точки останова на выделенной строке кода
Запустить сценарий редактора	Alt + R	Открывает список всех доступных действий и позволяет выбрать одно действие на исполнение (например, отправить выделенный фрагмент кода по электронной почте)
Открыть редактора меток	Ctrl+Alt+Spacebar	Открывает редактор меток и выполняет поиск и по выбранному тексту
Посмотреть определение (провалиться в метод)	F12	Переход к определению выбранного метода. Эта комбинация очень полезна для быстрой навигации

Табл. 2-3. Горячие клавиши редактора кода X++ (окончание)

Действие	Горячая клавиша	Описание
Переход к следующему методу	Ctrl+Tab	Устанавливает фокус на следующем методе в редакторе
Переход к предыдущему методу	Ctrl+Shift+Tab	Устанавливает фокус в предыдущем методе в редакторе
Выбор блока кода	Alt+Shift+клавиши со стрелками	Выберите код, который хотите выделить, нажав клавишу Alt и кнопку мыши, или удерживайте нажатой клавиши Alt и Shift и перемещайте курсор с помощью клавиш со стрелками
Отмена выделения	Esc	Отменяет текущее выделение
Удаление текущего выделения/строки	Ctrl+X	Удаляет текущее выделение или, если ничего не выбрано, удаляет текущую строку
Последовательный поиск	Ctrl+I	Начинает последовательный поиск по первому вхождению искомого текста при его вводе. Нажатие Ctrl+I приводит к следующему вхождению, Ctrl+Shift+I перемещает к предыдущему вхождению
Вставка XML-заголовка описания	///	Вставляет XML-комментарий в заголовок метода после ввода символов ///. При запуске в методе этот скрипт генерирует XML-шаблон описания с информацией, относящийся к данному классу или методу
Запуск заранее настроенного сценария редактора	<name of script>+Tab	Запуск заранее настроенного сценария редактора при вводе имени сценария редактора на пустой строке и нажатии клавиши Enter. Имена сценариев чувствительны к регистру
Закомментировать	Ctrl+E, C	Вставляет комментарии для выбранного выделения
Удаление комментариев для выбранного блока кода	Ctrl+E, U	Удаляет комментарии для выбранного выделения

Сценарии редактора

Редактор кода X++ содержит набор скриптов, которые можно вызвать, нажав на пиктограмму скрипта на панели инструментов редактора или нажав непосредственно в редакторе **<Название скрипта>+ТАВ**. Встроенный редактор сценариев обеспечивают такие функции, как:

- отправка выделенного кода по электронной почте;
- сохранение выделенного кода в файл;
- генерация кода по стандартным шаблонам, таким как *main*, *construct* и *parm*;
- открытие элемента-владельца текущего метода в отдельном окне АОТ.



Примечание. Генерация кода позволяет в течение нескольких минут создать новый класс с правильной структурой конструктора и инкапсуляцией членов-переменных класса с помощью *parm*-методов. *Parm*-методы (*parm* – сокращение от англ. *Parameter* [Параметр]) используются в качестве простого способа установки/получения значений переменных класса. Стоит упомянуть, что сгенерированный код соответствует всем рекомендациям Best Practice.



Совет. Для генерации метода *main* в классе нажмите **Ctrl+A**; чтобы выделить весь код в редакторе нового метода, напечатайте **main** и в конце нажмите клавишу Tab. Это приведет к тому, что весь текст будет замещен на стандартный шаблон статического метода *main*.

Набор сценариев редактора может быть расширен путем добавления новых сценариев – методов в класс *EditorScripts*.

Редактор меток

Термин метка в Microsoft Dynamics AX обозначает локализуемый текстовый ресурс. Текстовые ресурсы используются в системе повсеместно в качестве сообщений пользователю, заголовков управляющих элементов экранных форм, справочного текста в строке состояния, текста на веб-формах и еще во множестве других прикладных элементов. Метки лока-

лизуемы, а это значит, что они могут быть переведены на большинство языков.

Поскольку требования к пространству для отображения текстовых ресурсов на экране обычно зависят от языка, возможно, вы опасаетесь, что сам интерфейс пользователя тоже должен быть локализован вручную. Однако с технологией *IntelliMorph* интерфейс пользователя формируется динамически и учитывает все требования пространства на экране, накладываемые локализацией.

Технология системы меток проста. Все текстовые ресурсы хранятся в файле меток в кодировке Юникод, который должен иметь трехбуквенный идентификатор. В Microsoft Dynamics AX 2012 метки располагаются в AOT и распространяются с помощью файлов модели. На рис. 2-12 показан узел *Label Files* в AOT с файлами меток для других языков и выделенным файлом меток для США.

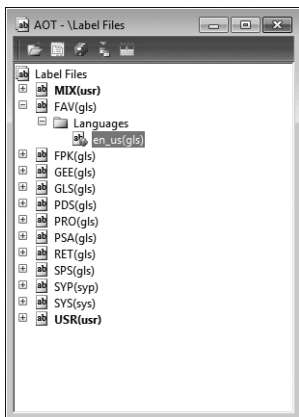


Рис. 2-12. Узел *Label Files* в AOT

Исходник представлен в виде текста следующего соглашения имен:

Ax< Идентификатор файла меток >< Местонахождение >.ALD

Ниже приведены два примера названий файлов меток, первый из которых соответствует файлу меток для Соединенных Штатов Америки, а второй – для Дании:

Axsysen-us.ALD

Axtstda.ALD

Каждый текстовый ресурс в файле меток имеет 32-битный целочисленный идентификатор, сам текст метки и ее описание, которое заполнять необязательно.

Структура файла меток очень проста: @<Идентификатор файла меток><Идентификатор метки> <Текст метки> [Описание метки]

На рис. 2-13 показан пример файла меток для одного из языков.

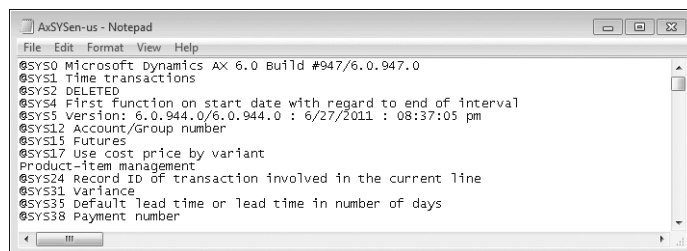


Рис. 2-13. Файл меток, открытый в программе Блокнот (Microsoft Notepad), в котором видно несколько меток для США

Такая простая структура файлов меток делает возможным локализацию без использования Microsoft Dynamics AX с применением сторонних продуктов.

Набор функций репозитория АОТ позволяет выполнять такие операции с файлами меток, как Экспорт в файл меток (эта функция выгружает файл меток для перевода внешними программами).

Новый файл меток можно создать при помощи Мастера файлов меток, доступного из контекстного меню узла *Label Files* репозитория АОТ или меню Сервис > Мастеры > Мастер файлов меток. Мастер проведет вас через весь процесс добавления нового файла меток или нового языка в существующий файл меток. После завершения работы Мастера файл меток готов к использованию. Если у вас имеется файл меток с расширением *.ald*, вы можете создать соответствующий элемент в репозитории АОТ (в контекстном меню узла *Label Files* это пункт Создать из файла).



Примечание. В названии файла меток может использоваться любая трехбуквенная комбинация, и любой файл меток может использоваться из любого прикладного слоя. Типичным заблуждением является то, что идентификатор файла меток должен совпадать с названием прикладного слоя, в котором этот файл

используется. Система Microsoft Dynamics AX имеет в слое *SYS* – файл меток *SYS*, в слое *SYP* – файл меток с идентификатором *SYP*. Такое наименование было выбрано в силу своей простоты, и следует помнить, что Microsoft Dynamics AX не накладывает никаких ограничений на идентификатор файлов меток.

Вот несколько советов по работе с файлами меток.

- При выборе идентификатора для файла меток следует остановить выбор на таком трехбуквенном идентификаторе, который с высокой степенью вероятности будет уникальным, например сокращение от названия вашей компании. Не стоит выбирать в качестве идентификатора файла имя прикладного слоя, такое как *VAR* или *USR*. Скорее всего, вам когда-то понадобится объединить две или более независимые разработки в одном приложении, что будет значительно труднее выполнить при совпадении идентификаторов файлов меток.
- Не стоит бояться использовать в своих разработках метки из меточных файлов, поставляемых компанией Microsoft, **но не следует вносить** изменения в существующие метки в данных файлах, поскольку они обновляются с каждой новой версией Microsoft Dynamics AX.

Создание новых меток

Для создания новых меток используется Редактор меток. Его можно запустить, используя любой из следующих способов.

- Нажать меню Сервис, далее выбрать Метка > Редактор меток.
- Нажать на кнопку Просмотр метки/текста на панели инструментов Редактора кода X++.
- Нажать на кнопку поиска в свойствах строкового типа в окне свойств.

Вы можете использовать Редактор меток и для поиска уже существующей метки (как показано на рис. 2-14). Всегда лучше использовать существующую метку, чем создавать новую. Новую метку можно создать путем нажатия комбинации клавиш *Ctrl+N* или нажатием на кнопку Создать на панели инструментов Редактора меток.

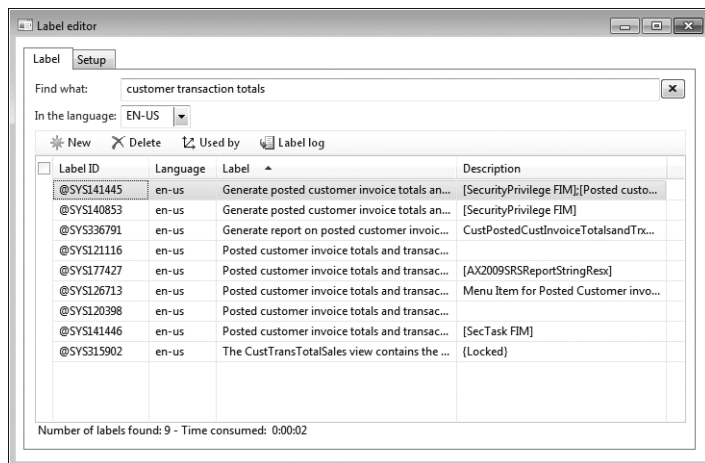


Рис. 2-14. Редактор меток

Помимо поиска существующих и создания новых меток, Редактор меток позволяет просмотреть, где выбранная метка используется в приложении. Также Редактор меток позволяет просмотреть журнал изменений меток, в который помещаются любые их изменения.

Вот несколько советов по созданию и повторному использованию меток.

- При повторном использовании метки вначале убедитесь, что метка имеет ожидаемое значение и на других языках. Некоторые слова имеют в определенных языках двойное значение, и при этом переводятся как два разных слова в других языках. Например, английское слово **can** может быть как глаголом, так и существительным. В поле Описание метки приводится описание значения метки, которое подразумевалось.
- При создании новых меток убедитесь, что используются только полные предложения или другие отдельно стоящие слова или фразы. Не конструируйте предложения объединением разных меток, состоящих из одного или нескольких слов, поскольку порядок слов в предложении различен от языка к языку.

Использование меток в коде X++

В среде разработки MorphX метки используются в формате *@<идентификатор файла меток><идентификатор метки>*. Если необходимо избежать автоматического преобразования метки в соответствующий ей текст, то для этого можно воспользоваться функцией *literalStr*. Для замещения текста значениями переменных можно использовать функцию *strFmt*, в которую передается строка, содержащая символы в формате *%n*, где *n* ≥ 1. Замещение текста также может использоваться и внутри меток. В следующем коде приведено несколько примеров.

```
// prints: Time transactions
print "@SYS1";

// prints: @SYS1
print literalStr("@SYS1");

// prints: Microsoft Dynamics is a Microsoft brand
print strFmt("%1 is a %2 brand", "Microsoft Dynamics", "Microsoft");
pause;
```

Вот несколько полезных советов по использованию меток в коде X++.

- При создании интерфейса пользователя следует использовать метки для всего выводимого на экран текста. При обращении к меткам из кода X++ следует использовать двойные кавычки.
- При создании системных текстов, таких как названия файлов, не стоит использовать метки. При обращении к системному тексту из кода X++ следует использовать одинарные кавычки. Для повторного использования системный текст можно поместить в макрос.

Использование одинарных и двойных кавычек позволяет отличать системный текст от текста интерфейса пользователя и с помощью инструмента рекомендаций Best Practice находить и сообщать о присутствии в коде жестко заданного текста интерфейса пользователя. Рекомендации разработки и соответствующий инструмент MorphX описываются далее в этой главе.

Компилятор

Каждый раз при изменении кода X++, как и в любом другом языке программирования, необходимо выполнить его компиляцию. Перекомпили-

ровать исходный код можно нажатием клавиши F7 в окне редактора кода X++. Код также будет автоматически откомпилирован при закрытии редактора кода или при сохранении прикладного элемента, в определение которого были внесены изменения.

В процессе компиляции кода также формируется список со следующей информацией.

- **Ошибки компиляции.** Они не позволяют успешно завершить компиляцию кода и должны быть исправлены как можно скорее.
- **Предупреждения компилятора.** Они обычно означают, что что-то в реализации кода неправильно. Список предупреждений, выдаваемых компилятором, приведен в табл. 2-4 далее в этом разделе. Предупреждения компилятора могут и должны быть исправлены. Попытки возврата кода, содержащего предупреждения компилятора, системой контроля версий отклоняются.
- **Задачи (также называемые to-do).** В процессе обработки кода компилятор отбирает комментарии в коде, которые начинаются с TODO. Такие комментарии могут быть полезны в процессе разработки для добавления напоминаний разработчику, но их следует использовать только в тех случаях, когда реализация кода не может быть завершена в данный момент. Использовать комментарий с **to-do можно, к примеру**, если вы ожидаете возврата кода от другого разработчика в системе контроля версий. Следует избегать использования to-do-комментариев для отсрочки завершения работы, которая может быть выполнена уже сейчас. Для разработчика нет ничего хуже, чем при отладке проблемы на приложении клиента обнаружить комментарий to-do, означающий, по сути, что проблема была известна, но проигнорирована.
- **Отступления от рекомендаций Best Practice.** Более сложные проверки кода выполняются с помощью инструмента Рекомендаций Best Practices. За дополнительной информацией обратитесь к разделу «Рекомендации разработки Best Practices» далее в этой главе.



Примечание. В отличие от других языков программирования, X++ требует компиляции только измененного кода. Это происходит вследствие того, что код на промежуточном языке, который формируется компилятором, существует параллельно с кодом X++ и метаданными. Некоторые изменения могут, конечно

же, потребовать перекомпиляции и другого кода, в случае если, к примеру, был переименован метод или перечень его параметров. Если код, в котором используется измененный метод, не будет перекомпилирован, то при его вызове возникнет ошибка времени выполнения. Это значит, что приложение может работать даже при наличии ошибок компиляции в коде, но только до тех пор, пока не будет вызван код с ошибками. После завершения всех запланированных модификаций кода следует всегда выполнять перекомпиляцию всего АОТ с исправлением ошибок компиляции, если таковые имеются. Если вы изменили объявление переменных в классе родителя, все классы наследники следует также перекомпилировать. Это можно сделать с помощью пункта **Compile Forward** в меню надстройки измененного узла класса.

Окно сообщений компилятора предоставляет доступ ко всем сообщениям, возникшим в процессе компиляции, как показано на рис. 2-15. Каждая категория сообщений может быть включена или выключена соответствующими кнопками в верхнем меню. На каждой вкладке содержатся однотипные сведения для каждой обнаруженной компилятором проблемы: описание проблемы и место в коде, где она была обнаружена.

Результаты компиляции могут быть экспортированы. Это будет полезно в том случае, если вы хотите предоставить список проблем другим членам команды.

Экспортированный файл будет сохранен в **HTML-формате**, и его можно будет просмотреть в браузере **Microsoft Internet Explorer** или импортировать в окно сообщений компилятора в другом сеансе **Microsoft Dynamics AX**.

Для того чтобы выбрать те типы проблем, о которых должен сообщать компилятор, в окне сообщений компилятора нажмите кнопку **Настройка\Компилятор**. Сообщения компилятора поделены на четыре уровня, как показано в табл. 2-4. Каждый уровень – это определенная глубина диагностики, определяющая содержимое сообщений, 1-й уровень – наиболее критичные сообщения, а в 4-й рекомендуемый уровень включаются правила хорошего тона программирования.

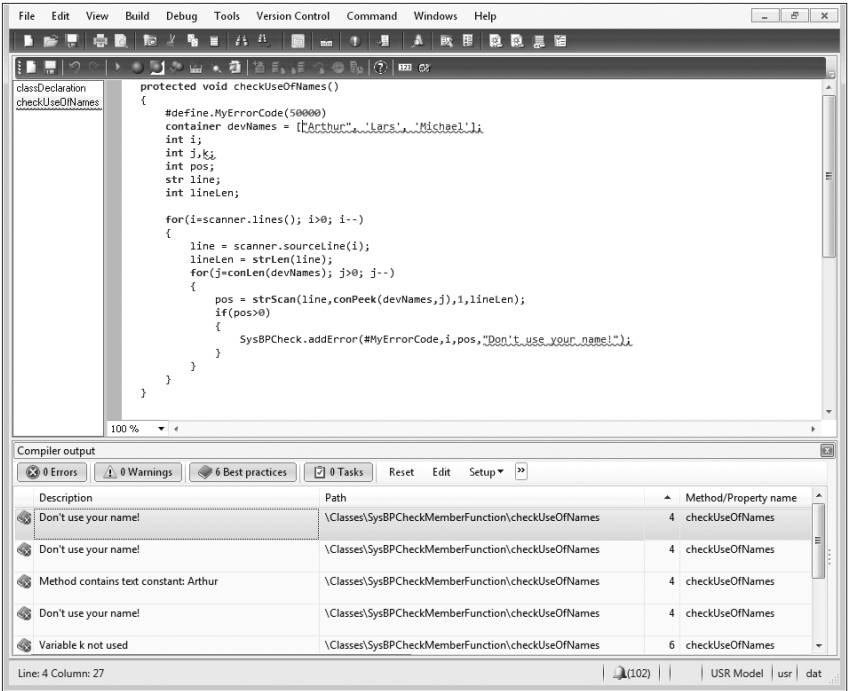


Рис. 2-15. Мощное сочетание Редактора кода X++ и компилятора

Табл. 2-4. Предупреждения компилятора

Сообщение компилятора	Уровень
Оператор BREAK был найден вне допустимого контекста	1
Новый метод производного класса не вызывает <i>super()</i>	1
Новый метод производного класса не может вызывать <i>super()</i>	1
Функция никогда не возвращает значение	1
Не все пути возвращают значение	1
Присвоение/Сравнение теряет точность	1
Недоступный код	2
Пустое составное утверждение	3
Имя класса должно начинаться с заглавной буквы	4
Наименования членов должны начинаться со строчной буквы	4

Рекомендации разработки Best Practices

Следуя рекомендациям разработки в системе Microsoft Dynamics AX, вы получаете следующие преимущества.

- Вы выявляете неочевидные ошибки. Следуя рекомендациям, вы можете обнаружить множество причин, ведущих к ошибкам, например недочеты в пограничных условиях, которые тяжело предусмотреть и на устранение которых уходит много времени. Рекомендации содержат наработанные советы специалистов по разработке в Microsoft Dynamics AX.
- Ваше обучение будет менее затратным. При выполнении похожих задач стандартным образом вы чувствуете себя более уверенным в незнакомой области разработки. Как следствие, добавление новых исполнителей в проект будет менее затратным, и те разработчики, которым придется модифицировать ваш код, смогут делать это быстрее.
- Вы делаете долгосрочные инвестиции. Код, который соответствует стандартам, менее требователен к доработкам при обновлении до версии Microsoft Dynamics AX 2012, после установки пакетов обновления или обновления до следующих релизов.
- Скорее всего, вы будете успевать в срок. Большая часть проблем, с которыми вы сталкиваетесь при разработке решения в Microsoft Dynamics AX, уже решены кем-то, по крайней мере, один раз. Выбор проверенного решения приводит к более быстрой реализации решения и меньшему количеству повторного тестирования. Вы можете найти решения известных задач и в Справке разработчика, и в самом коде приложения.

Пакет SDK Microsoft Dynamics AX 2012 содержит важные обсуждения рекомендаций разработки в Microsoft Dynamics AX. **Построение кода**, соответствующего проверенным стандартам и паттернам, не может гарантировать успех проекта, но сводит к минимуму риск задержки проекта, дорогостоящего исследования и снижает долгосрочные расходы на техническое обслуживание. Пакет SDK Microsoft Dynamics AX 2012 доступен по адресу: <http://msdn.microsoft.com/en-us/library/aa496079.aspx>.

Инструмент Best Practices – это мощное дополнение к обсуждению рекомендаций разработки в пакете SDK. Этот инструмент – версия инструмента анализа статического кода MorphX, похожий на FxCop для Microsoft

.NET Framework. Инструмент Best Practices встроен в компилятор, а результаты выводятся в окно компилятора, так же как и другие сообщения после процесса компиляции.

Цель статического анализа кода – это автоматическое определение дефектов и риска использования неверного шаблона в коде.

Чем дольше существует дефект, тем дороже становится поиск и исправление ошибки в фазе использования (design phase), гораздо дешевле исправить ошибку в поставляемом коде, работающем на нескольких клиентских сайтах. Инструмент Best Practices позволяет любому разработчику запускать анализ собственного кода и модели приложения, чтобы убедиться в том, что он соответствует набору предопределенных правил. Разработчики могут запускать анализ во время разработки, и они всегда должны это делать перед поставкой и тестированием. Поскольку приложение в Microsoft Dynamics AX больше, чем просто код, инструмент Best Practices также выполняет статический анализ метаданных – свойств, структур и связей, организованных в AOT. **Инструмент Best Practices показывает отклонение от правил Best Practices, как показано на рис. 2-15.**

Двойным щелчком по строчке в вкладке **Best Practices** открывается редактор кода X++ на строчке кода с нарушением, или если нарушение Best Practices было в метаданных, то будет открыт элемент в окне AOT.

Понимание рекомендаций

Инструмент рекомендаций Best Practices включает около 400 правил, но это лишь небольшая часть тех рекомендаций, которые описаны в руководстве разработчика Microsoft Dynamics AX SDK. Вы можете определить правила рекомендации, которые нужны вам, открыв окно параметров рекомендаций: в меню Сервис щелкните Настройка > Разработка, а затем Рекомендации.



Примечание. Для того чтобы отклонения от заданных правил разработки выводились в окно сообщений компилятора, уровень ошибок компилятора необходимо установить в значение 4. Для отключения проверки кода на соответствие рекомендациям нажмите кнопку Сервис > Параметры > Разработка, а затем Компилятор, и установите уровень диагностики в значение, меньшее чем уровень 4.

Рекомендации разработки Best Practices поделены на категории. По умолчанию все категории включены, как показано на рис. 2-16.

Рекомендации разработки делятся на три уровня в зависимости от серьезности проблемы.

- **Ошибки.** Большинство рекомендаций разработки сфокусированы на поиске ошибок. Любая попытка возвращения кода, содержащего ошибки рекомендаций, отклоняется. Все ошибки следует рассматривать как серьезные и исправлять при первой возможности.
- **Предупреждения.** По отношению к предупреждениям должно применяться правило 95/5, которое гласит, что 95% всех предупреждений должны рассматриваться как ошибки, а оставшиеся 5% составляют исключения из правил. В проектную документацию должны включаться объяснения по всем неисправленным предупреждениям.
- **Информация.** В некоторых ситуациях ваша реализация может производить побочный эффект, который не очевиден вам или пользователю (к примеру, при присваивании значения переменной, которая больше в коде не используется). Это, как правило, информационные сообщения.

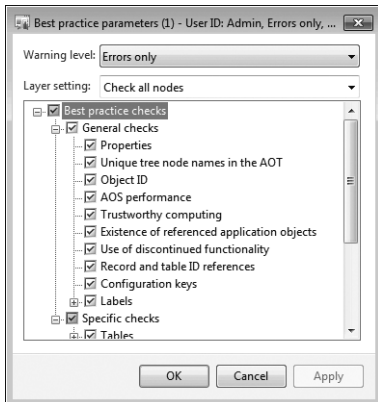


Рис. 2-16. Окно параметров рекомендаций Best Practice

Подавление ошибок и предупреждений

Microsoft Dynamics AX 2009 позволяет подавлять ошибки и предупреждения Best Practices. О подавленном отклонении от рекомендаций Best Practices сообщается в виде информационного сообщения. Это дает вам возможность отделить просмотренные отклонения от уже принятых. Для того чтобы подавить ошибку или предупреждение Best Practices, помести-

те строку, содержащую следующий текст, прямо перед отклонением от рекомендаций:

```
//BP Deviation Documented
```

Только небольшое подмножество рекомендаций может быть подавлено. Для принятия решения о подавлении рекомендации необходимо придерживаться следующих советов.

- **Опасные исключения API.** Там, где существуют исключения, которые невозможно определить автоматически, вам следует проверить каждую ошибку вручную, чтобы убедиться в правильности реализации кода. Опасные интерфейсы API являются типичным примером данного случая. Опасный прикладной интерфейс – это API, который при неправильном использовании может скомпрометировать безопасность системы. Если используется опасный API, система рекомендаций будет выдавать сообщение об ошибке, которое может быть подавлено. Использовать опасные API разрешается, когда приняты конкретные меры предосторожности, например обеспечение безопасности доступа к коду с использованием специальных классов. Вы можете подавить ошибку после принятия соответствующих мер защиты.
- **Ложные срабатывания.** Около 5% всех предупреждений являются ошибочными и могут быть подавлены. Обратите внимание, что могут быть подавлены только предупреждения, найденные в коде, а не в метаданных.

После настройки рекомендаций компилятор будет автоматически запускать проверку при каждой компиляции прикладных элементов. Результаты проверки отображаются на вкладке Рекомендации в окне Сообщения компилятора.

Некоторые нарушения рекомендаций по метаданным также могут быть подавлены, но процесс их подавления отличается. Вместо добавления комментария в исходном коде нарушение добавляется в глобальный список игнорируемых нарушений. Данный список ведется в макросе *Sys-BPCheckIgnore*. Это позволяет централизованно просматривать число подавлений, которое следует свести к минимуму.

Добавление собственных рекомендаций

Инструмент рекомендаций **Best Practices** позволяет создавать собственные наборы правил. Классы, которые используются для провер-

ки прикладных элементов на соответствие рекомендациям, называются *SysBPCheck<Element type>*. Для каждого узла в АОТ при компиляции элемента однократно вызываются методы *init*, *check* и *dispose*.

Один из самых интересных классов – это класс *SysBPCheckMemberFunction*, который вызывается для каждого участка кода X++ вне зависимости от того, является ли он методом класса, методом формы, макросом или другим методом. Например, для запрета разработчикам включать свои имена в исходный код можно реализовать соответствующую проверку, создав для этого следующий метод в классе *SysBPCheckMemberFunction*.

```
protected void checkUseOfNames()
{
    #Define.MyErrorCode(50000)
    container devNames = ['Arthur', 'Lars', 'Michael'];
    int i;
    int j,k;
    int pos;
    str line;
    int lineLen;

    for (i=scanner.lines(); i>0; i--)
    {
        line = scanner.sourceLine(i);
        lineLen = strLen(line);
        for (j=conLen(devNames); j>0; j--)
        {
            pos = strScan(line, conPeek(devNames, j), 1, lineLen);
            if (pos)
            {
                sysBPCheck.addError(#MyErrorCode, i, pos,
                    "Don't use your name!");
            }
        }
    }
}
```

Для добавления нового правила к списку проверяемых убедитесь, что вышеуказанный метод вызывается из метода *check*. Компиляция приведенного выше кода приведет к выводу ошибок рекомендаций **Best Practices**, приведенных в табл. 2-5.

Табл. 2-5. Ошибки рекомендаций Best Practices в методе *checkUseOfNames*

Сообщение	Строка	Столбец
Не используйте ваше имя!	4	28
Не используйте ваше имя!	4	38
Не используйте ваше имя!	4	46
Переменная k не используется	6	11
Метод содержит текстовую константу: 'Don't use your name!'	20	59

В рабочей реализации такого кода имена разработчиков, скорее всего, читались бы из файла. При этом нужно убедиться в том, что имена кэшируются при считывании из файла во избежание обращения к жесткому диску для считывания имен разработчиков при компиляции любого метода.



Примечание. Проверка на соответствие рекомендациям также обнаружила, что код содержит переменную *k*, которая была объявлена, но на нее нет ссылок. Эта одна из ценных проверок, гарантирующая, что код легко поддерживать в актуальном состоянии, что в свою очередь помогает избежать ошибок. В данном случае *k* была предназначена для специфических целей и может быть удалена.

Отладчик

Как и большинство сред разработки, в MorphX присутствует отладчик кода. Отладчик – это автономное приложение, не являющееся частью оболочки Microsoft Dynamics AX, как все остальные инструменты, упомянутые в данной главе. Будучи отдельным приложением, отладчик позволяет отлаживать код X++ в любых компонентах Microsoft Dynamics AX из следующего списка:

- Толстый клиент Microsoft Dynamics AX;
- Сервер приложения AOS;
- Business Connector (BC.NET).

Подробности о других сценариях отладки, таких как web-сервис, отчеты Microsoft SQL Server Reporting Services (SSRS) и Корпоративный портал, смотрите в главе 3.

Использование отладчика

Для запуска отладчика в процессе исполнения X++-кода необходимо, чтобы в коде, который выполняется, встретилась точка останова. Точки останова устанавливаются в редакторе кода X++ в рабочей области разработчика Microsoft Dynamics AX. Отладчик запускается автоматически, когда в любом компоненте происходит переход на точку останова.

Для работы отладчика его нужно включить для каждого компонента, как описано ниже.

- В рабочей области разработчика в меню Сервис нажмите Параметры > Разработка > Отладка, затем в списке Режим отладки выберите значение В точке останова.
- Для сервера приложения AOS откройте конфигурационную утилиту сервера Microsoft Dynamics AX 2009 Server Configuration из меню Пуск > Программы > Администрирование. Создайте новую конфигурацию (если необходимо) и установите флажок Enable breakpoints to debug X++ code running on this server.
- Для отладки интерпретируемого кода X++ в Корпоративном Портале используйте контекст BCSPROXY, в конфигурационной утилите Microsoft Dynamics AX Server создайте новую конфигурацию и, если необходимо, установите признак Enable Global Breakpoints.

Проверьте, что вы входите в локальную группу безопасности Microsoft Dynamics AX Debugging Users. Обычно это автоматически обеспечивается во время развертывания, но если вы установили Microsoft Dynamics AX не для текущего пользователя, то должны сделать это вручную через пункт меню Edit Local Users and Groups на панели управления. Это необходимо, чтобы запретить несанкционированную отладку, которая может привести к потере конфиденциальных данных, риску потере безопасности или внеплановой остановке служб.



Внимание. В рабочем приложении включение отладочных возможностей не рекомендуется. При переходе на точку останова исполнение кода будет приостановлено, а для пользователя это будет выглядеть так, как будто программа зависла.

Отладчик позволяет устанавливать и удалять точки останова при помощи клавиши F9.

Точку останова можно поставить на любую строку, однако если она устанавливается на строку, не содержащую оператор `X++`, то в действительности установится на следующий оператор в выбранном методе. Переход к точке останова на последней фигурной скобке никогда не выполняется.

Отладчик позволяет устанавливать и удалять точки останова при помощи клавиши `F9`.

Точки останова хранятся в таблицах *SysBreakpoints* и *SysBreakpointLists* базы данных. У каждого разработчика есть свой набор точек останова. Это означает, что точки останова не удаляются при закрытии клиента Microsoft Dynamics AX, а также то, что другие компоненты Microsoft Dynamics AX могут получить к ним доступ в отладочных целях.

Пользовательский интерфейс Отладчика

В главном окне Отладчика изначально показывается строка кода, точка останова в которой и вызвала отладчик. Далее вы пошагово управляете исполнением кода, просматривая при этом значения переменных и другие аспекты исполняемого кода. На рис. 2-17 представлен Отладчик со всеми окнами отладки.

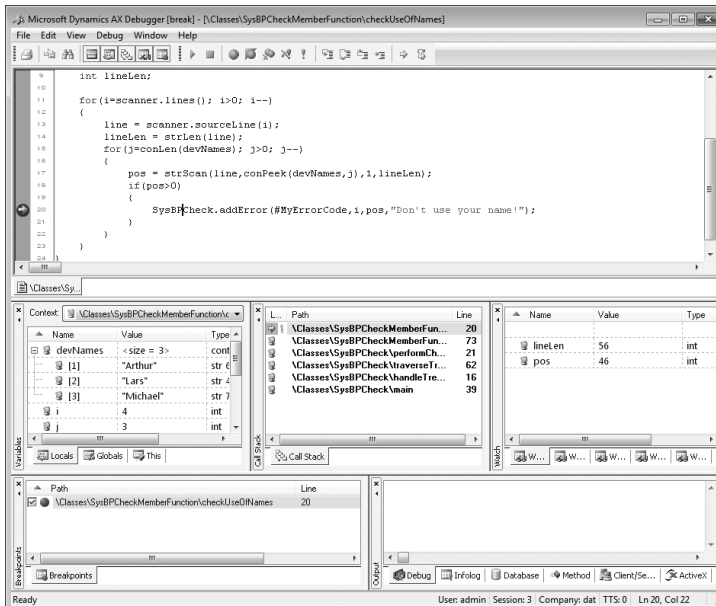


Рис. 2-17. Отладчик со всеми доступными окнами

В табл. 2-6 описываются различные окна отладки и некоторые из возможностей Отладчика.

Табл. 2-6. Элементы интерфейса (UI) Отладчика

Элемент Отладчика	Описание
Главное окно	<p>В главном окне отладчика отображается X++-код, исполняемый в данный момент.</p> <p>У каждой переменной есть всплывающая экранная подсказка, которая показывает значение этой переменной. Вы можете перетаскивать указатель, который расположен с левой стороны окна, на другую строку кода. Это может быть особенно полезно, если исполнение происходит не по тому пути, который вам нужен, или вы хотите повторить определенный шаг</p>
Окно Variables	<p>В данном окне можно просматривать локальные и глобальные переменные, а также переменные текущего объекта, с детализацией по имени, значению и типу.</p> <p>Областью действия локальных переменных является активный метод. Глобальными переменными являются глобальные классы, экземпляры которых всегда инициализированы: <i>Appl</i>, <i>Infolog</i>, <i>ClassFactory</i> и <i>VersionControl</i>.</p> <p>Переменные активного объекта имеют смысл только в классах и показывают переменные активного класса.</p> <p>Если в процессе пошагового исполнения значение переменной меняется, то она отмечается красным цветом в окне Variables. Каждая переменная отображается с иконкой сервера или клиента в зависимости от того, где находится объект этой переменной. Вы можете изменить значение переменной из данного окна. Для этого дважды щелкните на строке с нужной переменной и введите новое значение</p>
Окно Call Stack	<p>Окно Call Stack показывает путь выполнения кода до текущей точки исполнения. Щелчок мышью на строке в окне Call Stack откроет код выбранного метода в главном окне и обновит отображаемые локальные переменные в окне Variables. Иконка клиента или сервера указывает уровень, на котором исполняется код</p>

Табл. 2-6. Элементы интерфейса (UI) Отладчика (окончание)

Элемент Отладчика	Описание
Окно Watch	<p>В окне Watch отображаются имена, значения и типы переменных. Доступны пять окон Watch. Эту возможность можно использовать для группировки просматриваемых переменных с учетом предпочтений разработчика.</p> <p>В окне Watch можно просматривать значения переменных без ограничений области видимости переменных, накладываемых окном Variables. В это окно переменные можно перетаскивать как из главного окна кода, так и из окна Variables</p>
Окно Break-points	<p>В окне Breakpoints перечисляются все ваши точки останова. Точки останова можно удалять, включать или отключать прямо из этого окна</p>
Окно Output	<p>В окне Output отображаются активные сведения трассировки приложения, а также вывод в инфраструктуру Infolog, которая описана в главе 5. Окно Output включает в себя следующие страницы.</p> <ul style="list-style-type: none"> ■ Debug. Для вывода в данное окно трассировочной информации кода X++ используется статический метод <i>printDebug</i> класса <i>Debug</i>. ■ Infolog. На этой странице содержатся сообщения из очереди журнала сообщений Infolog. ■ Database, Client/Server, and ActiveX. Любые трассировочные сведения, включенные на вкладку Разработка в окне Сервис\Параметры, будут отображаться на данных страницах
Окно строки состояния	<p>Строка состояния в нижней части окна отладчика предлагает следующую важную контекстную информацию.</p> <ul style="list-style-type: none"> ■ Активный пользователь. Идентификатор текущего пользователя системы. Эта информация особо полезна при отладке входящих веб-запросов. ■ Активный сеанс. Идентификатор текущего сеанса на сервере приложения AOS. ■ Активная компания Идентификатор текущей компании. ■ Текущий уровень транзакции Текущий уровень транзакции. При достижении нулевого уровня транзакции она фиксируется в базе данных



Совет. Как разработчик, вы, возможно, захотите предоставить больше сведений об объекте в поле значения переменной, чем предусмотрено по умолчанию. Для класса значениями по умолчанию являются *New* и *Null*. Эти значения по умолчанию можно изменить путем перекрытия метода *toString*. Если выбранный класс не наследуется явно от класса *object* (базовый класс для всех классов), то для реализации данной функции нужно создать в классе новый метод *toString*, который возвращает базовый тип *str* и не принимает параметров.

Горячие клавиши отладчика

В табл. 2-7 приводится перечень наиболее важных горячих клавиш отладчика.

Табл. 2-7. Горячие клавиши отладчика

Действие	Горячая клавиша	Описание
Выполнить	F5	Продолжить выполнение кода
Прекратить отладку	Shift+F5	Прервать исполнение кода
Перейти к следующему оператору	F10	Выполнить текущий оператор кода и перейти к следующему
Перейти к курсору	Ctrl+F10	Выполнить текущий оператор кода и продолжить выполнение до строки, на которой установлен курсор
Перейти к реализации текущего оператора	F11	Перейти к следующему оператору или к первому оператору реализации текущего метода
Выйти из текущего метода	Shift+F11	Выполнить оставшийся код текущего метода и выйти из него
Включить/выключить точку останова	Shift+F9	Установить или удалить точку останова на текущей строке
Открыть окно Variables	Ctrl+Alt+V	Открыть или закрыть окно Variables

Табл. 2-7. Горячие клавиши отладчика

Действие	Горячая клавиша	Описание
Открыть окно Call Stack	Ctrl+Alt+C	Открыть или закрыть окно Call Stack
Открыть окно Watch	Ctrl+Alt+W	Открыть или закрыть окно Watch
Открыть окно Breakpoints	Ctrl+Alt+B	Открыть или закрыть окно Breakpoints
Открыть окно Output	Ctrl+Alt+O	Открыть или закрыть окно Output

Обратная разработка

Вы можете генерировать Visio-модели из существующих метаданных. Учитывая объем доступных в Microsoft Dynamics AX 2012 метаданных (более 50 тысяч прикладных элементов и более 18 миллионов строк текста при экспорте), при помощи АОТ практически невозможно получить ясное представление о том, как элементы связаны друг с другом. Обратная разработка (**Visio Reverse Endineering**) **оказывает огромную помощь в визуализации взаимосвязей метаданных.**



Примечание. Для использования инструмента Обратная разработка необходимо наличие установленной программы Visio 2007 или более поздней версии.

Обратная разработка может генерировать либо **UML-диаграмму модели данных**, либо **UML-диаграмму модели объектов**, включая все элементы из частного или общего проекта.

Для того чтобы открыть инструмент обратной разработки, щелкните правой кнопкой мыши на Проекте или Ракурсе, выберите Надстройки, затем Обратная разработка. Этот инструмент также доступен из меню Сервис.

В диалоговом окне, представленном на рис. 2-18, вы должны указать имя файла и тип генерируемой модели.

При нажатии кнопки ОК инструмент использует метаданные всех элементов в проекте для генерации документа Microsoft Office Visio, который автоматически открывается этой программой. Сгенерированные элемен-

ты можно перетаскивать из Visio Model Explorer на поверхность для рисования, которая изначально пуста. Любая взаимосвязь между элементами отображается автоматически.

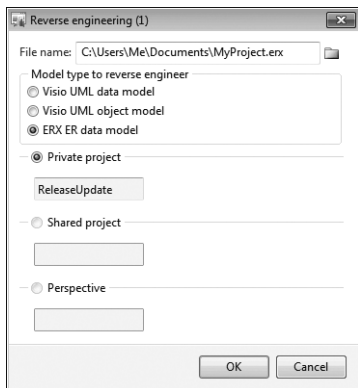


Рис. 2-18. Диалоговое окно Обратной разработки Microsoft Office Visio

UML-диаграмма модели данных

При генерации модели данных инструмент Обратная разработка ищет все таблицы в выбранном проекте. UML-диаграмма модели будет содержать класс для каждой таблицы в проекте, а также ее атрибуты и связи. На рис. 2-19 представлена диаграмма классов с таблицами *CustTable* (Клиенты), *InventTable* (Номенклатуры), *SalesTable* (Заказы на продажу) и *SalesLine* (Строки заказа). Для упрощения диаграммы некоторые атрибуты были удалены.

UML-диаграмма модели также содержит все таблицы, расширенные типы данных, перечислимые и базовые типы данных X++, на которые существуют ссылки в выбранном проекте. Это позволяет включать данные элементы в диаграммы без перезапуска инструмента Обратной разработки.

Поля в Microsoft Dynamics AX генерируются как UML-атрибуты в диаграмме. Все атрибуты помечаются как *public* в соответствии с принципом доступа к полям таблиц в Microsoft Dynamics AX. Атрибут, соответствующий полю первичного ключа, подчеркивается. Если поле является частью одного или более индексов, то названия индексов добавляются как префиксы к названию поля; если индекс уникален, то название индекса заключается в квадратные скобки.

Отношения в Microsoft Dynamics AX генерируются как UML-ассоциации. Свойство агрегации отношения устанавливается на основе двух условий в метаданных.

- Если отношение проверяется при вводе данных (свойство *Validate* установлено в значение Yes), то свойство агрегации устанавливается в значение *shared*, которое отображается белым ромбом.
- Если между двумя таблицами существует действие каскадного удаления, то к модели добавляется составная ассоциация, которая обозначается черным ромбом.

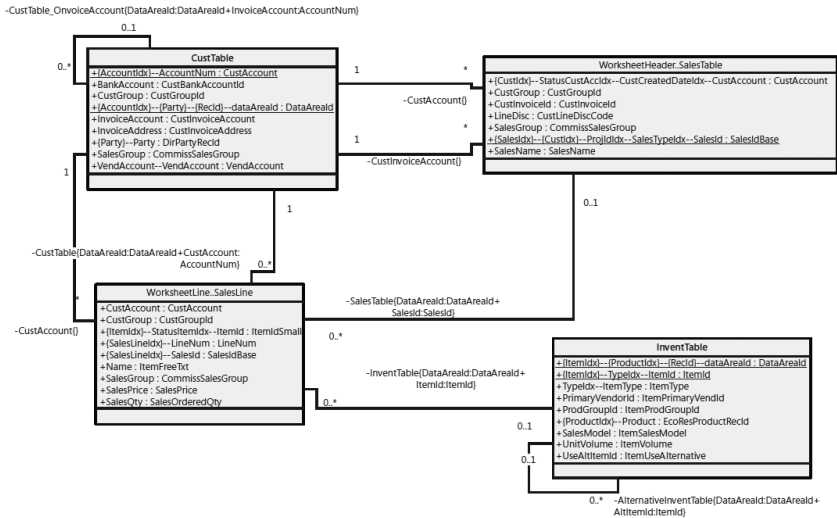


Рис. 2-19. UML-диаграмма модели данных

Конечное название ассоциаций – это название отношения в Microsoft Dynamics AX.

Названия и типы всех полей в отношении заключаются в квадратные скобки.

UML-диаграмма объектной модели

При генерации объектной модели инструмент Обратная разработка ищет в проекте Microsoft Dynamics AX классы, таблицы и интерфейсы. UML-диаграмма будет содержать класс для каждой таблицы и каждого класса из проекта Microsoft Dynamics AX и интерфейс для каждого интерфейса из проекта Microsoft Dynamics AX. UML-диаграмма модели также будет содержать атрибуты и операции, включая возвращаемые типы, параметры и типы параметров. На рис. 2-20 представлена объектная модель наиболее

важных в Microsoft Dynamics AX классов и интерфейсов *RunBase* и *Batch*. Для упрощения диаграммы некоторые атрибуты и операции, а также параметры операций, были удалены.

UML-диаграмма модели также содержит таблицы, классы, расширенные типы данных, перечислимые типы и базовые типы данных X++, на которые ссылаются объекты из выбранного проекта. Это позволяет включать данные прикладные элементы в диаграммы без необходимости повторного запуска инструмента Обратная разработка.

Поля таблиц и свойства классов Microsoft Dynamics AX генерируются в UML как атрибуты. Все поля генерируются как *public*-атрибуты, а свойства классов генерируются как *protected*-атрибуты. Для каждого атрибута также указывается тип. Методы генерируются как **UML-операции**, включая возвращаемые типы, параметры и типы параметров.

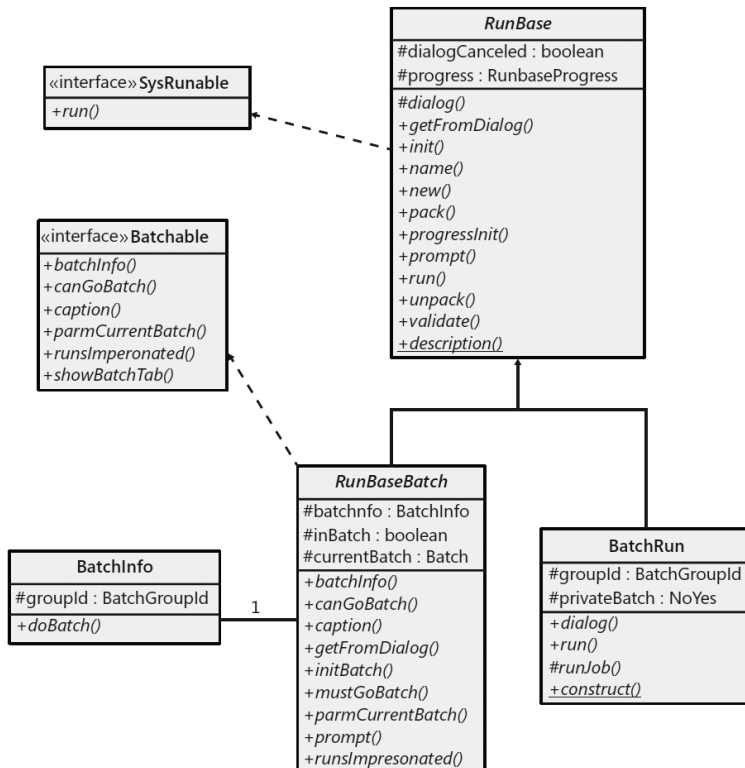


Рис. 2-20. UML-диаграмма модели объектов

Инструмент Обратная разработка также собирает информацию об обобщениях (классы, которые наследуются от других классов), реализациях (классы, которые реализуют определенные интерфейсы) и ассоциациях (классы, которые используют друг друга). Ассоциации ограничиваются ссылками в свойствах класса.



Примечание. Для получения перечня названий параметров операций необходимо включить режим отладки перед запуском инструмента Обратной разработки. Только в режиме отладки названия параметров считываются из метаданных и размещаются в стек. Для включения режима отладки выберите значение В точке останова в элементе управления Режим отладки, на вкладке Разработка формы Параметров.

Модель данных сущностей и отношений

При генерации модели данных сущностей и отношений инструмент Обратная разработка ищет в проекте Microsoft Dynamics AX **таблицы и представления**. Схема сущностей и отношений содержит тип сущности для каждой таблицы АОТ проекта и использует атрибуты для отображения полей таблиц. На рис. 2-21 показана схема сущностей и отношений (Entity Relationship Diagram, ERD) для таблиц *HcmBenefit* (Льгота), *HcmBenefitOption* (Параметр льготы), *HcmBenefitType* (Тип льготы) и *HcmBenefitPlan* (План льготы).

Поля таблиц Microsoft Dynamics AX генерируются как столбцы схемы сущностей и отношений. Столбцы могут выступать в роли внешнего ключа (*foreign key*, FK), альтернативного ключа (*alternate key*, AK), инверсионного входа (*inversion entry*, IE) и необязательного атрибута (*optional*, O). Столбец внешнего ключа используется для ссылки на определенную запись другой таблицы, альтернативный ключ уникально идентифицирует запись в текущей таблице, инверсионный вход идентифицирует ноль или более записей текущей таблицы (обычно по полям неуникальных индексов), а необязательные атрибуты не требуют обязательного ввода значения. Отношения Microsoft Dynamics AX генерируются как отношения между сущностями в ER-диаграмме. Свойство *EntityRelationshipRole* на отношениях в Microsoft Dynamics AX используется как описание роли внешнего ключа в схеме сущностей и отношений.

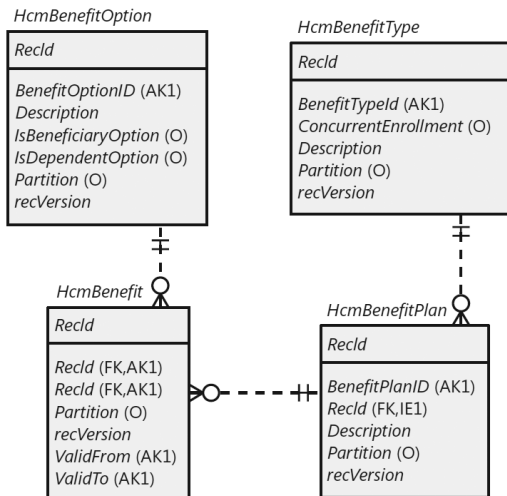


Рис. 2-21. ERD-диаграмма с использованием нотации IDEF1X



Примечание. Инструмент Обратная разработка генерирует файл с расширением *.ERX*. Для работы со сгенерированным файлом в Microsoft Office Visio создайте новую диаграмму модели данных, выберите базу данных, после чего импортируйте файл Erwin ERX. После этого можно с помощью мыши перетаскивать нужные таблицы из области таблиц и представлений (доступно из меню База данных) на рабочую поверхность диаграммы.

Обозреватель таблиц

Сценариев использования этого небольшого, но чрезвычайно полезного инструмента множество. Обозреватель таблиц позволяет просматривать записи в таблице напрямую, без необходимости создания какого-либо пользовательского интерфейса. Это может быть чрезвычайно полезно при отладке кода, проверке модели данных, модификации или чистке данных, а также во множестве других ситуаций.

Обозреватель таблиц доступен из подменю Настройки контекстного меню следующих элементов АОТ:

- таблиц;

- таблиц, используемых в качестве источников данных в формах, отчетах и источниках данных (*data sets*);
- системных таблиц, перечисленных в AOT в узле **System Documentation\Tables**.



Примечание. Обзорщик таблиц написан исключительно на языке X++. В AOT его можно найти по названию *SysTableBrowser*. Этот класс является хорошим примером динамического связывания источника данных с таблицей во время выполнения.

На рис. 2-22 показана форма Обзорщика таблиц, открытая для таблицы *CustTrans*. В дополнение к возможностям выборки, сортировки и фильтрации данных, которые обеспечивает элемент управления *grid*, Обзорщик таблиц позволяет вводить операторы SQL напрямую в форму с использованием синтаксиса языка X++, а также просматривать результаты этого запроса. Это прекрасная возможность для проверки сложных операторов SQL. Синтаксис поддерживает группировку, сортировку, агрегацию данных и списки полей.

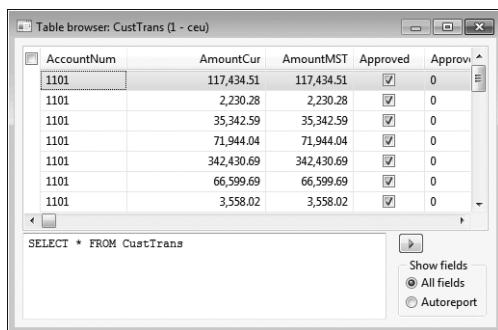


Рис. 2-22. Обзорщик таблиц, показывающий демо-данные таблицы *CustTrans*

Обзорщик таблиц также позволяет выбрать просмотр полей только из группы *AutoReport*. Эти поля будут распечатаны в отчете при нажатии пользователем кнопки Печать в форме, с которой данная таблица связана в качестве источника данных. Обычно эти поля содержат самую интересную информацию. Такой вариант может облегчить поиск значений для таблиц, содержащих большое количество полей.



Примечание. Обзорщик таблиц – это обычная экранная форма, которая так же как и другие использует технологию Intelli-Morph. Это означает, что в обзорщике не могут быть показаны поля, для которых свойство *visible* установлено в значение No, а также поля, для которых у пользователя не достаточно прав доступа.

Поиск по АОТ

Поиск – это все, размер приложения **Microsoft Dynamics AX** требует мощного и эффективного инструмента поиска.



Совет. Поиск по АОТ может использоваться для поиска примеров реального использования определенного API в дополнение к примерам из руководства разработчика.

Окно Поиска, показанное на рис. 2-23, может быть запущено из любого узла в АОТ путем нажатия комбинации клавиш Ctrl+F или выбора пункта Найти... в контекстном меню. Обратите внимание, что средство поиска по АОТ поддерживает множественное выделение прикладных элементов в АОТ.

Вкладка Наименование и местоположение определяет то, что вы ищете, и то, где производится поиск.

- В управляющем элементе Поиск выбирается одно из двух доступных значений: Методы и Все узлы. При выборе поиска по всем узлам отображается дополнительная вкладка Свойства.
- Значение, введенное в управляющий элемент По имени, ограничивает поиск только по введенным узлам. Данный параметр используется редко.
- В поле Содержащийся текст вводится текст для поиска, с поддержкой регулярных выражений.
- При выборе флажка Показать код источника в результаты поиска включаются фрагменты исходного кода, содержащего искомую строку. Это делает просмотр результатов поиска более удобным.

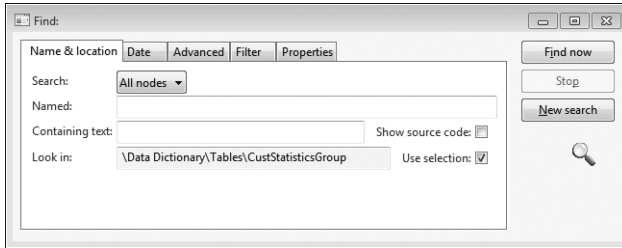


Рис. 2-23. Поиск по AOT

По умолчанию поиск выполняется по узлу AOT (и подчиненным узлам), выбранному при открытии окна поиска. Если при открытом окне поиска изменяется фокус в AOT, то автоматически обновляется и содержимое поля Искать в. Это просто незаменимо при поиске в нескольких узлах AOT по одним и тем же критериям. Однако данное поведение можно отключить, сняв флажок По выбору.

На вкладке Дата вы можете указать дополнительные фильтры для поиска, такие как Дата и Автор модификации.

На вкладке Дополнительно вы можете указать дополнительные параметры поиска, такие как прикладной слой для поиска, размер элемента, тип элемента и уровень, на котором выполняется элемент.

На вкладке Фильтр, показанной на рис. 2-24, вы можете написать сложный запрос с использованием языка X++ и библиотек типов. Код, написанный в поле Источник, будет выступать телом метода со следующим определением:

```
boolean FilterMethod(str _treeNodeName,  
                    str _treeNodeSource,  
                    XRefPath _path,  
                    ClassRunMode _runMode)
```

Пример на рис. 2-24 использует класс *SysScannerClass* для поиска любого вхождения ключевого слова языка X++ `TTSAbort`. Сканнер главным образом используется для передачи лексем в синтаксический анализатор в процессе компиляции. Однако в данном случае он используется для поиска специального ключевого слова. Это более точный (но и более медленный) метод, чем поиск при помощи регулярных выражений, поскольку комментарии в языке X++ не являются лексемами.

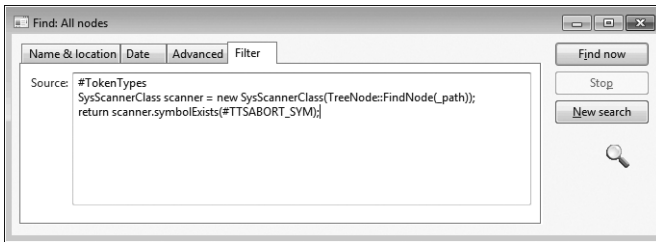


Рис. 2-24. Фильтрация результатов поиска

Вкладка Свойства появляется, только когда выбрано значение Все узлы в поле Поиск на первой вкладке. Искомое значение может быть задано для любого свойства. Пустое значение для поиска – это мощный инструмент при просмотре свойств, он приводит к поиску выбранного свойства во всех узлах, как показано на рис. 2-25. Процесс поиска начинается с нажатия на кнопку Найти. Результаты отображаются внизу окна по мере нахождения совпадений.

Двойное нажатие на любой строке набора результатов откроет редактор X++ с курсором на совпавшем фрагменте кода. При нажатии правой кнопкой мыши на строках в наборе результатов открывается контекстное меню, содержащее подменю Настройки.

Сравнение прикладных элементов

Обычно в системе существуют несколько версий одного и того же прикладного элемента. Данные версии могут происходить из различных прикладных слоев или различных версий элемента в системе контроля версий, или же они могут быть модифицированными версиями объектов, существующими в памяти. Microsoft Dynamics AX содержит встроенный инструмент Сравнения прикладных элементов, который показывает любые отличия между двумя версиями прикладного элемента.

Сравнение показывает изменения элементов, которые могут происходить тремя способами.

- В результате изменения свойства в метаданных.
- В результате изменения кода X++.
- В результате изменения порядка подузлов элемента, к примеру порядка вкладок на экранной форме.

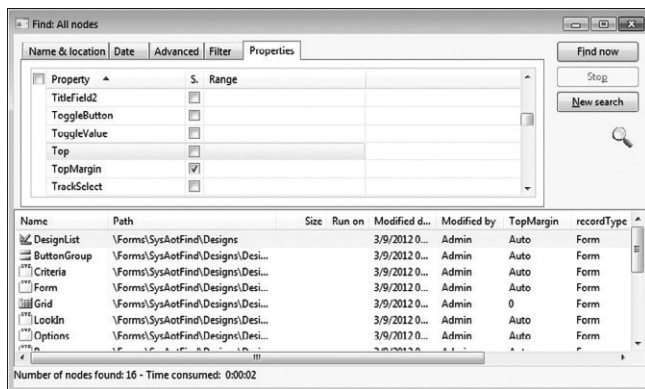


Рис. 2-25. Результаты сравнения в инструменте Сравнение прикладных элементов

Запуск инструмента сравнения

Запустить сравнение разных версий прикладного элемента можно из контекстного меню прикладного элемента, выбрав пункт Сравнение... Открывшееся диалоговое окно позволяет выбрать версии элементов для сравнения, как показано на рис. 2-26.

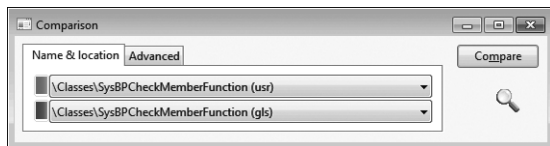


Рис. 2-26. Диалоговое окно сравнения

Версии элемента, из которых можно выбирать, поступают из многих источников. Вот список всех возможных источников для версий прикладных элементов.

- Версии на основе стандартных прикладных слоев (*SYS, SYP, GLS, GLP, FPK, FPP, SLN, SLP, ISV, ISP, VAR, VAP, CUS, CUP, USR* и *USP*).
- Версии на основе старых прикладных слоев (*old SYS, old SYP, and so on*). Если присутствует базовая модель данных, то элементы из этих файлов будут доступны в данном перечне. Это позволяет сравнивать более старую версию прикладного элемента с последней/текущей его версией. За дополнительной информацией о прикладных слоях и базовой модели данных обратитесь к главе 21.

- **Версии элементов из системы контроля версий (Version 1, Version 2, и т.д.).** Можно получить и использовать для сравнения любую из версий прикладного элемента в системе контроля версий. Система контроля версий описывается далее в этой главе.
- **Очищенная версия, следующая рекомендациям разработки (Washed).** Несколько простых проблем, которые противоречат рекомендациям разработки, могут быть разрешены автоматически посредством очистки. Сравнение с очищенной версией показывает, как текущая реализация отличается от принятых рекомендаций разработки. Для получения максимальной пользы от такого сравнения следует установить флажок С учетом регистра на вкладке Дополнительно диалогового окна Сравнения.
- **Файл экспорта/импорта (XPO).** Перед импортом прикладных элементов их можно сравнить с существующими в системе элементами (которые будут перезаписаны в процессе импорта). Сравнение в процессе импорта запускается путем выбора флажка Отобразить подробности в диалоговом окне Импорт и нажатия правой кнопкой мыши на любом элементе, который выделен полужирным шрифтом, что означает, что этот элемент уже существует в приложении. Новые элементы полужирным шрифтом, соответственно, не выделены.
- **Обновленная версия (Upgraded).** MorphX может автоматически составить план обновления класса. Необходимость обновления класса возникает в процессе обновления версии приложения. Шаг «Создать проект обновления» в контрольном списке обновления автоматически определяет модифицированные классы, конфликтующие с новыми версиями этих классов. Конфликт возникает в том случае, если вы изменили оригинальную версию класса, а оригинальная версия, в свою очередь, была изменена в новой версии продукта. MorphX создает план обновления, выполняя слияние ваших изменений и изменений в новой версии класса. MorphX требует наличия всех трех версий класса — оригинальной версии в базовой модели, версии с вашими изменениями в текущем слое в базовой модели и версии с изменениями новой версии продукта в том же слое, что и оригинальная версия.

Программа установки автоматически проверяет, что правильные версии доступны в требуемых для обновления местах в процессе обновления. Пример разрешения конфликта показан на рис. 2-27.



Рис. 2-27. Как создается план обновления версии прикладного элемента



Примечание. Также можно сравнивать и два различных элемента. Для этого выберите два элемента в АОТ, щелкните правой кнопкой мыши и выберите Сравнение... в открывшемся контекстном меню.

На рис. 2-28 показана вкладка Дополнительно, на которой указываются различные настройки сравнения.

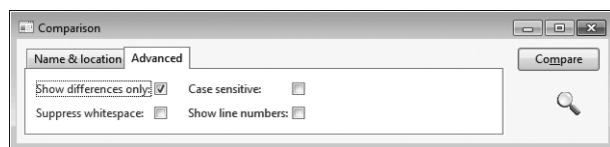


Рис. 2-28. Настройки сравнения прикладных элементов на вкладке Дополнительно

Настройки сравнения, показанные на рис. 2-28, описаны в следующем перечне.

- **Показывать только различия.** Все одинаковые узлы исключаются из просмотра, что упрощает поиск измененных узлов. Данный параметр выбран по умолчанию.
- **Пропускать пробелы.** Символы-разделители, такие как пробелы и символы табуляции, преобразуются в одиночный пробел при сравнении. При сравнении множество последовательных пробельных сим-

волов игнорируется таким же образом, как это делает компилятор. Данный параметр выбран по умолчанию.

- **С учетом регистра.** Поскольку язык X++ не чувствителен к регистру символов, средство сравнения также по умолчанию не различает регистр символов. Однако в определенных случаях, например при сравнении с очищенными версиями элементов, требуется чувствительность к регистру символов. По умолчанию данный параметр не выбран.
- **Показывать номера строк.** При сравнении в результат могут быть добавлены номера строк в коде X++. **Это может быть полезным в процессе обновления больших участков кода.**

Использование инструмента сравнения

После выбора элементов и установки параметров, вы можете начать сравнение прикладных элементов, нажав на кнопку Сравнение. Результаты отображаются в трехпанельном виде, как показано на рис. 2-29. На верхней панели отображаются сравниваемые элементы, левая панель представляет собой структуру, аналогичную АОТ, а на правой панели показываются детали выбранного в дереве элемента.

Иконки в дереве указывают на то, как изменился каждый из узлов. Красная или синяя галочка означает, что узел существует только в красной или только в синей версии элемента. Красный цвет в данном примере соответствует слою *sys*, а синий соответствует слою *old sys*. Серая галочка указывает, что узлы идентичны, но различаются один или более подузлов. Символ неравенства (≠) на красно-синем фоне означает, что узлы различаются в двух выбранных версиях прикладных элементов.



Примечание. Каждый узел в дереве на левой панели содержит контекстное меню, которое предоставляет доступ к подменю Надстройки и пункту меню Новое окно. Пункт меню Новое окно открывает выбранный узел, включая элементы старого (*old*) слоя, в АОТ.

Детальные результаты сравнения показываются на правой панели. Для выделения различий на данной панели также используется цветовое кодирование. Если элемент является редактируемым, то на панели отображаются и небольшие иконки действия. Эти иконки позволяют внести из-

менения в исходный код, метаданные и узлы, которые могут сэкономить ваше время при выполнении обновления прикладного элемента. Правая или левая стрелка удалит или добавит отличие, а наклонная стрелка переместит отличие в новую позицию. Стрелки всегда используются парами, поэтому вы сможете увидеть, куда отличие будет перемещено и откуда. Элемент является редактируемым, если он находится на текущем слое и извлечен из системы контроля версий при ее использовании.

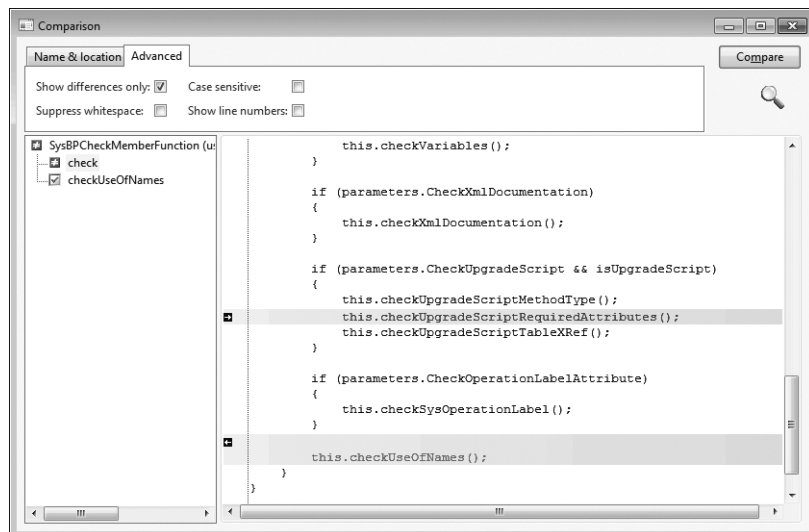


Рис. 2–29. Результаты сравнения прикладных элементов

Программные интерфейсы сравнения элементов

Несмотря на то что Microsoft Dynamics AX использует функциональность сравнения только для целей разработки, общая функциональность сравнения может использоваться значительно шире. Доступные интерфейсы программирования позволяют сравнивать любые типы сущностей и представлять отличия в форме дерева или в виде текста. Класс *Tutorial_CompareContextProvider* показывает, как просто можно сравнивать бизнес-данные с использованием данного API и представляя результат в стандартной форме Сравнение. Пример состоит из двух следующих частей.

- ***Tutorial_Comparable***. Данный класс реализует интерфейс *SysComparable*. По большому счету он просто создает текстовое представление выбранного клиента.
- ***Tutorial_CompareContextProvider***. Данный класс реализует интерфейс *SysCompareContextProvider*. Он обеспечивает контекст для сравнения. Например, он создает экземпляр класса *tutorial_Comparable* для каждого клиента, устанавливает параметры сравнения по умолчанию и обрабатывает контекстные меню.

На рис. 2-30 показан результат сравнения записей двух выбранных клиентов, выполненный в примере, который описан выше.

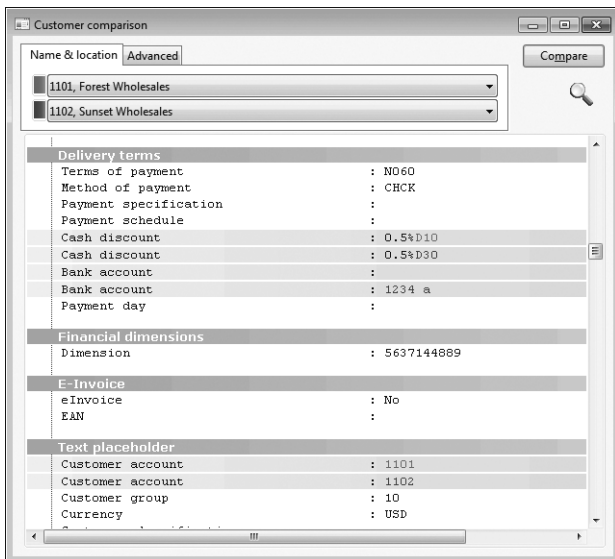


Рис. 2-30. Результат сравнения двух клиентов с использованием программного интерфейса сравнения

Функциональность построчного сравнения может также использоваться прямо в X++.

Статический метод *run* класса *SysCompareText*, показанный в следующем блоке кода, принимает в качестве параметров две строки и возвращает контейнер, в котором содержатся отличия двух строк. Также для управления сравнением можно устанавливать дополнительные параметры.

```
public static container run(str _t1,  
    str _t2,  
    boolean _caseSensitive = false,  
    boolean _suppressWhiteSpace = true,  
    boolean _lineNumbers = false,  
    boolean _singleLine = false,  
    boolean _alternateLines = false)
```

Перекрестные ссылки

Принцип работы перекрестных ссылок в Microsoft Dynamics AX прост. Если элемент использует другой элемент, то эта ссылка регистрируется в системе. Перекрестные ссылки позволяют определить, какие элементы использует конкретный прикладной элемент, а также какие прикладные элементы используются другими прикладными элементами. Microsoft Dynamics AX предоставляет инструмент Перекрестные ссылки для доступа и управления сведениями о перекрестных ссылках.

Вот несколько типичных сценариев использования инструмента перекрестных ссылок.

- Вы хотите найти примеры использования определенного прикладного элемента. Если вы не нашли ответ в документации к продукту, то можете воспользоваться перекрестными ссылками для поиска реальных примеров реализации.
- Вам нужно выполнить анализ влияния прикладного элемента. Если вы изменяете прикладной элемент, то хотите знать, на какие другие элементы окажет влияние ваше изменение.

Для сохранения точности результатов перекрестные ссылки необходимо регулярно обновлять. Обновление обычно длится несколько часов. Требуемое место в вашей базе данных для хранения информации о перекрестных ссылках – около 1 Гб для стандартного приложения.

Перекрестные ссылки можно обновить, выбрав из меню Сервис > Перекрестные ссылки > Периодические операции > Обновить. Обновление перекрестных ссылок также компилирует весь АОТ, поскольку сведения о перекрестных ссылках извлекаются компилятором.



Совет. Регулярное обновление перекрестных ссылок очень важно, если вы хотите полагаться на данные сведения при принятии решений. Если вы работаете в среде разработки совместно с другими разработчиками, то используете сведения о перекрестных ссылках совместно с другими членами вашей команды. Для такой среды хорошим подходом является обновление перекрестных ссылок каждую ночь. Если вы работаете в локальной среде разработки, то можете включить обновление перекрестных ссылок при компиляции.

Но следует помнить, что это замедлит процесс компиляции. Другой вариант – вручную обновлять перекрестные ссылки для элементов в текущем проекте. Это можно сделать путем выбора пункта Настройки > Перекрестные ссылки > Обновить в контекстном меню проекта.

В дополнение к основной информации о перекрестных ссылках, существуют еще две более мелкие подсистемы перекрестных ссылок.

- **Модель данных.** В данной подсистеме перекрестных ссылок хранится информация об отношениях между таблицами. Она преимущественно используется формой запросов и инструментом Обратная разработка.
- **Иерархия типов.** В этой подсистеме перекрестных ссылок хранится информация о наследовании классов и типов данных.

Для получения дополнительной информации об этих подсистемах и инструментах, которые зависят от них, обратитесь к Microsoft Dynamics AX 2012 SDK (<http://msdn.microsoft.com/en-us/library/aa496079.aspx>).

Дальнейшее обсуждение данных средств выходит за рамки книги. Вы можете найти полный список с перекрестными ссылками элементов, он представлен согласно узлу перечислимого типа System Documentation\Enums\xRefKind в AOT.

При обновлении средство Перекрестные ссылки сканирует все метаданные и X++-код в поиске ссылок на элементы тех видов, которые перечислены выше.



Совет. Хорошей идеей является использование встроенных функций замещения при ссылке на прикладные элементы в коде X++. Функция замещения возвращает либо название прикладно-

го элемента, либо его идентификатор. Функции замещения называются `<ElementKind>Str` или `<ElementKind>Num` соответственно. Использование функций замещения дает два преимущества: во время компиляции происходит проверка существования прикладного элемента, на который указана ссылка в коде, а также эта ссылка учитывается при обновлении перекрестных ссылок. Вот пример использования встроенных функций замещения:

```
// Prints ID of MyClass, such as 50001
print classNum(myClass);
```

```
// Prints "MyClass"
print classStr(myClass);
```

```
// No compile check or cross-reference
print "MyClass";
```

За дополнительной информацией о встроенных функциях замещения обратитесь к главе 20.

Для доступа к сведениям об использовании прикладного элемента нажмите правой кнопкой мыши на любом элементе в АОТ, перейдите в **Настройки**, затем в **Перекрестные ссылки** и выберите пункт **Чем используется**. Если такой пункт меню не присутствует в перечне, то элемент нигде не используется или перекрестные ссылки не были обновлены.

На рис. 2-31 показано, где используется метод `prompt` класса `RunBaseBatch`.

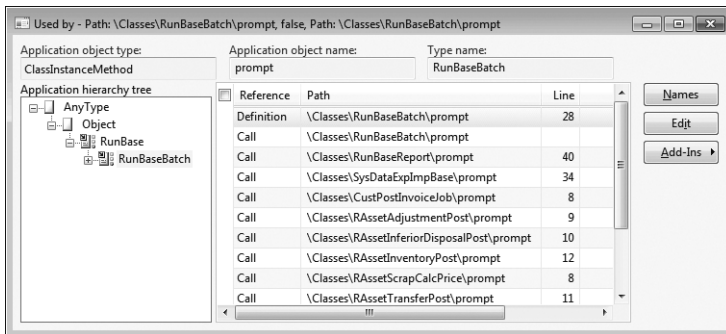


Рис. 2-31. Использование метода `RunBaseBatch.prompt` в форме перекрестных ссылок

При просмотре перекрестных ссылок для метода класса также доступна иерархия объектов, позволяющая вам увидеть, используется ли тот же метод в классе-родителе или потомке. Для типов, которые не поддерживают наследование (таблицы, методы таблиц и поля таблиц), иерархия типов не отображается.

Контроль версий

Контроль версий – это инструмент MorphX, который позволяет использовать систему контроля версий, такую как Microsoft Visual SourceSafe (VSS) или Microsoft Visual Studio Team (TFS) Foundation Server, для отслеживания изменений прикладных элементов в AOT. Инструмент доступен из нескольких мест: из Управления версиями в рабочей области разработчика, из панели инструментов управления AOT и на редакторе кода X++, а также из контекстного меню прикладных элементов в AOT.

Преимущества использования системы контроля версий состоят в следующем.

- **История изменений прикладных элементов.** Все изменения сохраняются вместе с их описанием, что позволяет просматривать журнал изменений и получать старые версии элемента.
- **Принудительная проверка качества кода.** Реализация контроля версий в Microsoft Dynamics AX предоставляет полностью настраиваемый порог качества для всех возвратов кода в систему. С установленным порогом качества все изменения проверяются на соответствие рекомендациям Best Practice. Если изменения в прикладном элементе не удовлетворяют установленным критериям, то они отклоняются системой.
- **Локальная среда разработки.** Каждый разработчик может иметь локально установленную среду разработки, в которой все изменения будут выполняться без влияния на других разработчиков. Когда модификации готовы, они могут быть возвращены в систему для использования другими разработчиками этой сборки. Это позволяет разработчику вносить изменения в фундаментальные области системы без создания проблем нестабильности приложения для других. Это также позволяет разработчикам не зависеть от централизованного сервера разработки в случае его сбоя.

Невзирая на то, что использование системы контроля версий при разработке необязательно, настоятельно рекомендуется использовать ее для

проектов любой величины. Microsoft Dynamics AX 2012 обеспечена поддержка трех систем контроля версий: VSS 6.0 и TFS, которые следует использовать для больших проектов разработки, а также MorphX VCS. MorphX VCS следует использовать на более мелких проектах, которые ранее не могли оправдать издержки, связанные с дополнительными накладными расходами по использованию сервера системы контроля версий. В табл. 2-8 приводится сравнение доступных систем контроля версий.

Табл. 2-8. Сравнение систем контроля версий

	Без системы Контроля версий	MorphX VCS	VSS	TFS
Требуемое кол-во серверов приложения AOS	1	1	1 на каждого разработчика	1 на каждого разработчика
Требуемое кол-во серверов базы данных	1	1	1 на каждого разработчика	1 на каждого разработчика
Требуется процесс сборки	Нет	Нет	Да	Да
Формат Мастер данных	Модель данных	Модель данных	XPOs	XPOs
Возможность локальной разработки	Нет	Нет	Да	Да
Множественные извлечения	N/A	Нет	Настраивается	Настраивается
Описание изменений	Нет	Да	Да	Да
История изменений	Нет	Да	Да	Да
Поддержка Списков изменений (автоматический возврат набора файлов)	N/A	Нет	Нет	Да
Использование порога качества кода	Нет	Настраивается	Настраивается	Настраивается

Прикладные элементы, существующие на сервере контроля версий, являются представлениями элементов АОТ в виде файлов. В качестве формата файла используется стандартный формат экспорта Microsoft Dynamics AX (.хро). Каждый .хро-файл содержит только один элемент.

Использование MorphX VCS не накладывает дополнительных требований на инфраструктуру, что идеально подходит партнерам с большим количеством проектов. В такой архитектуре разработчик часто работает параллельно над несколькими проектами, переключаясь между ними и часто возвращаясь к старым проектам. В таких ситуациях очевидны преимущества использования истории изменений прикладных элементов. Всего парой щелчков мыши можно настроить MorphX VCS на сохранение изменений в базу данных бизнес-приложения. Несмотря на то что MorphX VCS предоставляет многие из возможностей нормальных систем контроля версий, у этой системы, конечно же, есть ограничения. К примеру, MorphX VCS не предоставляет инструментарий для технического обслуживания, такого как резервное копирование, архивация и маркирование.

Microsoft Visual SourceSafe (VSS) и Microsoft Team Foundation Server (TFS), напротив, разработаны для больших проектов разработки, в которых много разработчиков работают совместно над одним проектом длительное время (к примеру, независимый поставщик программных продуктов работает над вертикальным решением).

На рис. 2-32 представлена типичная топология среды разработки с использованием Microsoft Visual SourceSafe или Microsoft Team Foundation Server, в которой каждый разработчик локально устанавливает сервер приложения (АОС) и базу данных. Каждому разработчику также требуется копия всех .хро-файлов. Файлы .хро передаются при взаимодействии разработчика с сервером контроля версий.



Рис. 2-32. Типичная топология среды разработки с использованием системы контроля версий



Примечание. В более ранних версиях Microsoft Dynamics AX, Team Server должен был назначать уникальные идентификаторы созданным элементам. Microsoft Dynamics AX 2012 использует новую схему распределения ID, которая устраняет необходимость в Team Server. За дополнительной информацией обращайтесь к главе 21.

Жизненный цикл прикладного элемента

На рис. 2-33 представлен жизненный цикл прикладного элемента в системе контроля версий. Когда элемент отмечен зеленым цветом, то его можно редактировать, в противном случае его можно только просматривать без права изменения.

Новый прикладной элемент можно создать двумя способами.

- Создать полностью новый элемент.
- Изменить существующий элемент, что приведет к созданию перекрытой версии элемента на текущем прикладном слое. Поскольку элементы в системе контроля версий хранятся для каждого слоя, то модификация прикладного элемента равносильна созданию нового элемента.

После создания прикладного элемента он должен быть добавлен в систему контроля версий. Сначала назначьте прикладному элементу корректное имя в соответствии с соглашениями имен, затем выберите пункт Создать в контекстном меню. После создания элемента необходимо выполнить его возврат в систему контроля версий.



Рис. 2-33. Жизненный цикл прикладного элемента

Возвращенный элемент может быть переименован. Переименование прикладного элемента удаляет элемент со старым названием и добавляет элемент с новым названием.

Проверка качества

Перед возвратом в систему контроля версий прикладные элементы проверяются на соответствие критериям качества кода. Критерии качества задаются при настройке системы контроля версий. Поддерживаются следующие проверки качества:

- ошибки компиляции;
- предупреждения компиляции;
- задания компилятора;
- ошибки рекомендаций Best Practice.

Если проверка качества кода включена, то она выполняется при возврате любого элемента. Если проверка завершается неудачно, то возврат элемента прекращается. В данном случае необходимо устранить проблему и заново выполнить возврат элемента.

Смена регистра в исходном коде

Проверка на соответствие правилам использования регистров символов в названиях переменных, объявлениях параметров и ссылках, доступная из подменю Надстройки контекстного меню прикладного элемента, выполняется при помощи инструмента Смена регистра в исходном коде. Проверка может быть выполнена перед возвратом элемента. Параметр для запуска данного инструмента указывается при настройке системы контроля версий путем выбора флажка Запуск обновления Title Case (заглавные буквы).

Общие задачи системы контроля версий

В табл. 2-9 описываются некоторые из задач, которые обычно выполняются в системе контроля версий. В следующих разделах описываются дополнительные задачи, которые можно выполнять при помощи системы контроля версий в Microsoft Dynamics AX.

Табл. 2-9. Задачи, выполняемые в системе контроля версий

Действие	Описание
Извлечение элемента	<p>Для изменения элемента его необходимо извлечь из системы контроля версий.</p> <p>Извлечение элемента блокирует его в системе, чтобы другие разработчики не смогли изменить его, пока вы над ним работаете. Просмотреть элементы, которые вами извлечены, можно через использование меню Microsoft Dynamics AX, щелкнув Управление версиями > Извлечь. Элементы, которые вы извлекли (или элементы, которые вы создали, но еще не вернули), отображаются в АОТ синим, а не черным, цветом</p>
Отмена извлечения элемента	<p>Если вы решили не изменять извлеченный вами элемент, вы можете отменить извлечение. Это снимет блокировку элемента и импортирует серверную версию элемента для отмены ваших локальных изменений</p>
Возврат элемента	<p>После завершения изменений прикладные элементы должны быть возвращены в систему контроля версий, чтобы стать частью следующей сборки. При выборе пункта Возврат в контекстном меню появится диалоговое окно, представленное на рис. 2-34, в котором отображены все извлеченные вами на данный момент элементы. Диалоговое окно Вернуть по умолчанию показывает все открытые элементы. Любые элементы, возврат которых не требуется, могут быть удалены из списка путем нажатия комбинации клавиш Alt+F9.</p> <p>Вот рекомендуемая процедура возврата прикладных элементов.</p> <ul style="list-style-type: none"> ■ Выполните синхронизацию для обновления определений всех прикладных элементов в вашей локальной среде разработки до последней версии. ■ Проверьте работоспособность всей функциональности. Простой компиляции кода недостаточно! ■ Выполните возврат элементов

Табл. 2-9. Задачи, выполняемые в системе контроля версий (*продолжение*)

Действие	Описание
Создание элемента	<p>При использовании контроля версий новые элементы создаются таким же образом, как и в среде MorphX без системы контроля версий. Данные элементы не будут частью вашего возврата до тех пор, пока вы не выберете пункт Создать в контекстном меню. Можно создавать элементы всех типов за исключением тех, которые перечислены в параметрах системных настроек (в рабочей области разработчика меню Управление версиями > Параметры системы). По умолчанию в систему контроля версий не принимаются задания и частные проекты.</p> <p>Новые элементы должны соответствовать соглашениям имен Microsoft Dynamics AX. По умолчанию задействованы соглашения имен из рекомендаций Best Practices, поэтому вы не можете вернуть элементы с названиями, такими как <i>aaaElement</i>, <i>Del_Element</i>, <i>element1</i> или <i>element2</i>. (Разрешены только те DEL_ -прикладные элементы, которые требуются в целях обновления версий.) Требования к наименованию прикладных элементов могут быть изменены в параметрах системных настроек</p>
Переименование элемента	<p>Для переименования элемент должен находиться в возвращенном состоянии. Поскольку все ссылки в .хро-файлах основываются строго на именах (а не на идентификаторах), то все ссылки на переименованные элементы тоже должны быть обновлены. Например, если изменяется имя поля таблицы, то также должны обновиться все формы или отчеты, которые используют данное поле. Большинство ссылок в метаданных в АОТ основываются на идентификаторах, поэтому на них не оказывает никакого воздействия переименование элементов. В большинстве случаев достаточно просто извлечь форму или отчет и включить их в возврат для обновления .хро-файла. Для определения ссылок может использоваться функциональность перекрестных ссылок. Ссылки в X++-коде основываются на именах. Вы можете использовать компилятор для поиска затрагиваемых ссылок. История изменений прикладного элемента при переименовании остается без изменений. В результате переименования не происходит никакой утери сведений</p>
Удаление элемента	<p>Элементы удаляются общепринятым в Microsoft Dynamics AX способом. Для удаления прикладного элемента он должен находиться в возвращенном состоянии. Вы можете просмотреть ожидаемые удаления в экранной форме Объекты в обработке</p>

Табл. 2-9. Задачи, выполняемые в системе контроля версий (окончание)

Действие	Описание
Получение последней версии элемента	Если кто-либо из разработчиков возвратил новую версию элемента, то вы можете использовать пункт Последняя версия из контекстного меню для получения самой последней версии определения элемента. Данный пункт недоступен, если элемент извлечен вами. Получение последней версии не используется в MorphX VCS

Рис. 2-34 показывает диалоговое окно возврата в систему контроля версий.

Метки

Работа с метками очень похожа на работу с прикладными элементами. Для изменения, удаления или добавления метки необходимо извлечь файл меток, содержащий нужную метку. Файл меток можно извлечь из формы Редактор меток.

Главное отличие между извлечением прикладных элементов и извлечением файлов меток заключается в том, что допускаются одновременные извлечения файлов меток несколькими разработчиками. Это означает, что другие разработчики могут изменять метки в то же время, что и вы.

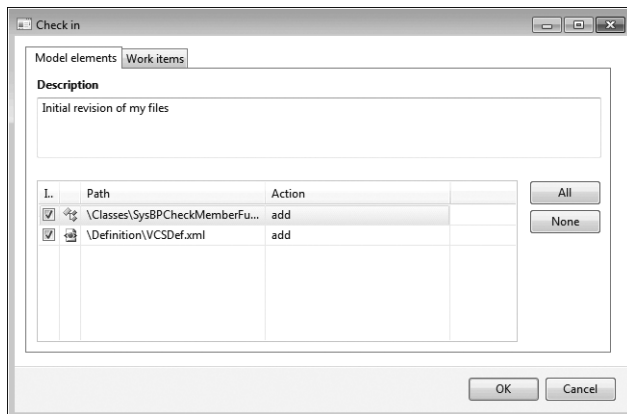


Рис. 2-34. Диалоговое окно возврата в систему контроля версий

При создании новой метки с использованием системы контроля версий назначается временный идентификатор метки (например, @\$AA0007

по сравнению с @USR1921). При возврате файла меток ваши изменения автоматически объединяются с последней версией файла и назначенный временный идентификатор метки обновляется. Все ссылки в коде автоматически обновляются новым идентификатором метки. Временные идентификаторы устранили необходимость в централизованном управлении Team Server, который был необходим в предыдущей версии Microsoft Dynamics AX 2009, так как идентификаторы больше не назначаются при создании меток. Если изменить или удалить метку, которая была изменена или удалена другим разработчиком, то ваши изменения теряются. Утерянные изменения отображаются в Infolog после завершения возврата.

Синхронизация

Синхронизация позволяет получить последние версии всех элементов. Это обязательный шаг, который должен выполняться перед возвратом элементов. Его можно выполнить из рабочей области разработчика (меню Управление версиями > Синхронизировать).

Синхронизация делится на три операции, которые выполняются автоматически в следующей последовательности.

1. Копирование последних версий файлов с сервера контроля версий на локальный диск.
2. Импорт файлов в АОТ.
3. Компиляция импортированных файлов.

Синхронизация должна использоваться для поддержания актуальности вашей системы. Синхронизация не затронет созданные вами элементы и прикладные элементы, которые вы извлекли из системы контроля версий до выполнения синхронизации.

На рис. 2-35 показано диалоговое окно Синхронизация.

Выбор флажка Инициировать позволяет получить последние версии всех файлов вне зависимости от того, изменялись они или нет, а затем импортирует каждый из файлов.

При использовании VSS можно выполнять синхронизацию с определенной отметкой в VSS. Таким способом можно с легкостью выполнить синхронизацию с определенной версией сборки приложения.

Синхронизация не используется в MorphX VCS.

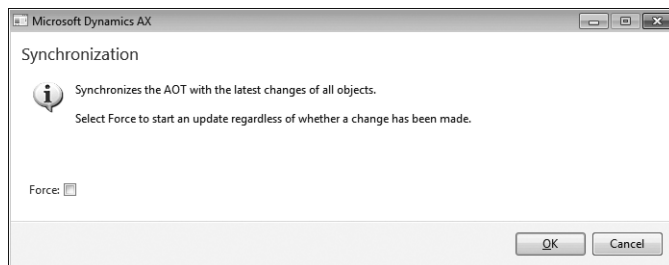


Рис. 2–35. Диалоговое окно Синхронизация

Журнал синхронизации

То, как вы отслеживаете версии прикладных элементов на клиенте, зависит от используемой системы контроля версий. VSS требует, чтобы Microsoft Dynamics AX отслеживала версии прикладных элементов самостоятельно.

Когда вы получаете последнюю версию прикладного элемента из системы контроля версий, она копируется в локальную папку хранилища. Для того чтобы содержимое файла было отражено в АОТ, его необходимо импортировать в Microsoft Dynamics AX.

Для минимизации риска частичной синхронизации, для каждого файла создается запись в журнале синхронизации. Когда все файлы копируются локально, журнал обрабатывается и файлы автоматически импортируются в Microsoft Dynamics AX.

Ошибки синхронизации приводят к тому, что система находится в состоянии частичной синхронизации. Для завершения синхронизации необходимо перезапустить клиент Microsoft Dynamics AX и вновь выполнить импорт. Для перезапуска импорта используется журнал синхронизации, доступный из рабочей области разработчика в меню Управление версиями > Журнал синхронизации.

В экранной форме Журнала синхронизации, представленной на рис. 2-36, отображается каждый пакет импортируемых файлов. Если флажок Обработано не выбран, то импорт завершился неудачно и должен быть выполнен повторно.

Журнал синхронизации не доступен при использовании MorphX VCS.

История изменений

Одним из самых больших преимуществ системы контроля версий является возможность отслеживать изменения прикладных элементов. Выбор

пункта История из контекстного меню прикладного элемента отображает список всех изменений элемента, как показано на рис. 2-37.

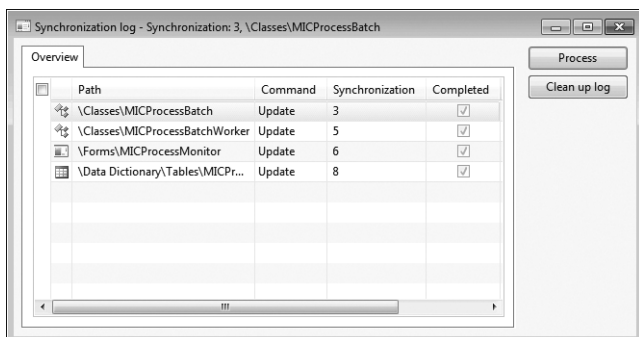


Рис. 2-36. Экранная форма Журнал синхронизации

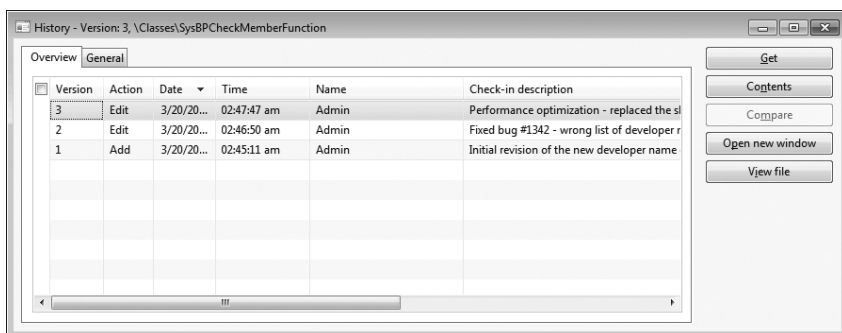


Рис. 2-37. История изменений прикладного элемента

В экранной форме Истории изменений показываются номер версии, выполненное действие, время выполнения действия и автор действия. Вы также можете просмотреть номер изменения и его описание.

Набор кнопок в экранной форме Изменения позволяет проводить дальнейшее исследование каждой версии элемента. Нажатие на кнопку Содержание открывает форму, в которой отображаются все элементы, изменение которых произошло вместе с текущим изменением. Нажатие на кнопку Сравнение открывает форму Сравнение, которая позволяет выполнить построчное сравнение двух версий элемента. Кнопка Новое окно открывает окно АОТ, в котором показывается выбранная версия элемента, что может быть полезным для исследования свойств прикладного элемента, поскольку позволяет вам использовать стандартный набор инстру-

ментов MorphX. Нажатие на кнопку Просмотр файла открывает *.xro*-файл для выбранной версии элемента в программе Блокнот.

Сравнение версий прикладных элементов

Сравнение является ключевой функцией для получения максимума преимуществ от использования системы контроля версий. Сравнение элементов можно запустить из нескольких мест, включая пункт Сравнение в подменю Настройки.

На рис. 2-38 показана форма Сравнение, в которой выбраны две версии таблицы *CustTable*.

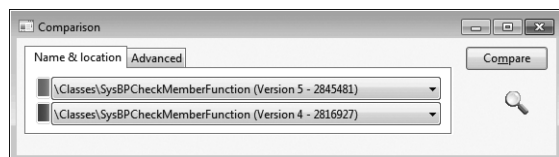


Рис. 2-38. Сравнение версий прикладных элементов

Обратите внимание, что форма Сравнение в дополнение к перечню слоев прикладного элемента содержит еще и список всех возвращенных версий этого элемента.

Объекты в обработке

При работе над проектом можно легко забыть, какие элементы были открыты для редактирования. Экранная форма Объекты в обработке, представленная на рис. 2-39, показывает список прикладных элементов, которые на данный момент извлечены из системы контроля версий. Обратите внимание на столбец, содержащий действие, выполненное над прикладным элементом. Удаленные элементы доступны только из данной формы, так как они больше не отображаются в АОТ.

Открыть форму Объекты в обработке можно из меню Управление версиями > Ожидающие объекты рабочей области разработчика.

Сборка приложения

Поскольку система контроля версий содержит файлы *.xro*, а не файл модели, для генерации файла модели из *.xro*-файлов требуется процесс сборки.

Ниже приведен обобщенный обзор процесса сборки.

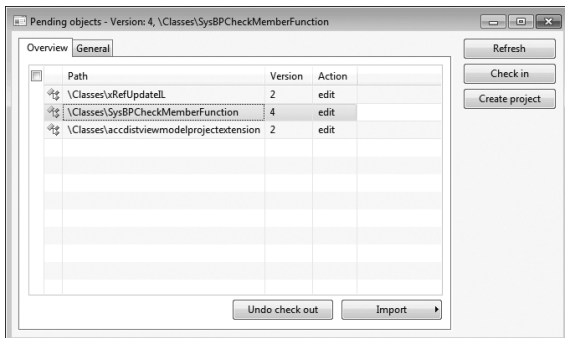


Рис. 2-39. Объекты в обработке

1. Используйте утилиту командной строки *CombineXPOs* для создания одного *.xpo*-файла путем объединения всех *.xpo*-файлов. Назначение данного шага – создать единственный *.xpo*-файл, подходящий для Microsoft Dynamics AX. Для поддержки ссылочной целостности в процессе импорта Microsoft Dynamics AX требует, чтобы все ссылаемые элементы были представлены в *.xpo*-файле или уже существовали в АОТ.
2. Импортируйте созданный *.xpo*-файл с использованием параметра командной строки `AOTIMPORTFILE=<FileName.xpo> -MODEL=<Model Name>` в `Ax32.exe`. На данном шаге выполняется импорт *.xpo*-файла и полная компиляция приложения. После завершения компиляции новая модель в хранилище моделей готова.
3. Экспортируйте модель в файл с помощью утилиты командной строки `axutil: axutil export /model:<model name> /file:<model file name>`.
4. Данные инструкции должны выполняться для каждого прикладного слоя и каждой модели, подлежащей сборке. Процесс сборки не используется в MorphX VCS.

Интеграция с другими системами контроля версий

Поддержка систем контроля версий в Microsoft Dynamics AX реализована так, чтобы Microsoft Dynamics AX могла быть интегрирована с любой системой контроля версий.

Для интеграции с новой системой контроля версий требуется создать новый класс, реализующий интерфейс *SysVersionControlFilebasedBackEnd*. Реализация данного класса отвечает за обеспечение связи приложения с сервером системы контроля версий.