

Глава 7

Корпоративный портал

В этой главе

- Введение
- Архитектура Корпоративного портала
- Компоненты Корпоративного портала
- Разработка для Корпоративного портала
- Безопасность
- Интеграция с SharePoint

Введение

С помощью веб-клиента Корпоративного портала организации могут расширить аудиторию пользователей своей ERP-системы за счет включения в нее клиентов, поставщиков, деловых партнеров и сотрудников, предоставив им доступ к бизнес-приложениям и возможность работать совместно, где бы они ни находились.

Пользователи осуществляют доступ к Корпоративному portalу удаленно с помощью веб-браузера или из корпоративной сети в зависимости от того, как развернут и настроен Корпоративный портал. Он служит для пользователей отправной точкой при доступе к любым данным, структурированным или неструктурированным, таким как данные проводок, отчеты, диаграммы, ключевые показатели эффективности (KPI), документы и оповещения. Более подробную информацию об интерфейсе Корпоративного портала вы можете найти в главе 5.

Также Корпоративный портал служит веб-платформой. Он содержит набор веб-страниц по умолчанию и пользовательских ролей, которые вы можете использовать в исходном виде или же модифицировать для соответствия уникальным бизнес-требованиям. Кроме того, вы можете моди-

фицировать или создавать новые бизнес-приложения с веб-интерфейсом в Microsoft Dynamics AX, а также реализовать веб-интерфейс для уже существующих бизнес-приложений.

В Корпоративном портале в версии Microsoft Dynamics AX 2012 был добавлен целый ряд новых возможностей для ускорения разработки бизнес-приложений. Используя новый роле-ориентированный подход к разработке, вы можете создавать страницы списков, которые работают как в Корпоративном портале, так и в Windows-клиенте Microsoft Dynamics AX, сокращая таким образом время разработки. Новые шаблоны проектов и элементы управления в Microsoft Visual Studio способствуют быстрой разработке приложений. А новые свойства в метаданных позволяют моментально начать использовать общие шаблоны проектирования, для реализации которых прежде вам приходилось писать код. Более подробную информацию вы можете найти в разделе «What's New: Enterprise Portal for Developers in Microsoft Dynamics AX 2012» по адресу: <http://msdn.microsoft.com/en-us/library/gg845087.aspx>.

Архитектура Корпоративного портала

Корпоративный портал соединяет вместе все лучшее из технологий Microsoft Dynamics AX, ASP.NET и Microsoft SharePoint, чтобы предоставить пользователям бизнес-приложение с веб-интерфейсом и богатыми возможностями. Он сочетает обширный функционал SharePoint со структурированными бизнес-данными из Microsoft Dynamics AX.

Вы можете воспользоваться преимуществами богатой прикладной модели Microsoft Dynamics AX, определяя логику доступа к данным и бизнес-логику в MorphX. Определять же пользовательский веб-интерфейс и создавать пользовательские веб-элементы управления вы можете с помощью Visual Studio. Веб-элементы управления могут содержать компоненты Microsoft Dynamics AX, такие как *AxGridView*, а также стандартные элементы управления ASP.NET, такие как *TextBox*. Логика доступа к данным и бизнес-логика, определенная в Microsoft Dynamics AX, становятся доступными для пользовательских веб-элементов управления посредством привязки данных, программных интерфейсов для доступа к данным и метаданным приложения, а также прокси-классов.

Архитектура Корпоративного портала изображена на рис. 7-1.

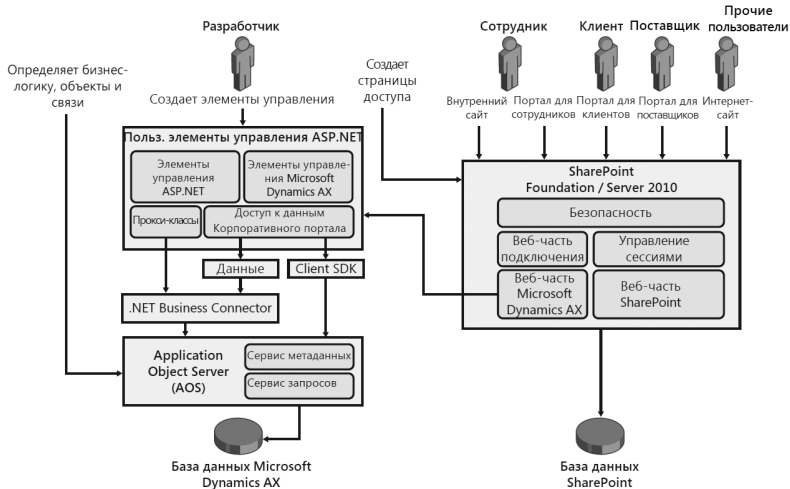


Рис. 7-1. Архитектура Корпоративного портала

Корпоративный портал использует инфраструктуру веб-частей SharePoint. Веб-части являются повторно используемыми компонентами SharePoint, которые генерируют HTML и предоставляют базис для модульного представления данных. Используя эту инфраструктуру, вы можете создавать веб-страницы, которые другим разработчикам будет просто модифицировать, а пользователям персонально настраивать для себя. Кроме того, инфраструктура веб-частей позволяет легко интегрировать контент, совместно работать и использовать сторонние приложения. Веб-страницы могут содержать как веб-части Microsoft Dynamics AX, так и веб-части SharePoint.

Веб-части Microsoft Dynamics AX представляют информацию и предоставляют доступ к функциональности Microsoft Dynamics AX. Веб-часть User control может содержать любой пользовательский веб-элемент управления ASP.NET, включая пользовательские веб-элементы управления Корпоративного портала. К Microsoft Dynamics AX она подключается с использованием инфраструктуры Корпоративного портала. Вы можете использовать веб-части SharePoint для реализации различных сценариев совместной работы и представления контента. Например, у вас может быть специально разработанная веб-часть, которая подключается к корпоративному сайту, забирает последние новости о вашей организации и отображает их. Или же у вас может быть веб-часть, отображающая данные с другого сайта SharePoint вашей организации.

Первое, что нужно для разработки нового или модификации существующего приложения на базе Корпоративного портала, – это понимать, как взаимодействуют браузер пользователя с клиентской стороны и Корпоративный портал с серверной стороны, когда пользователь осуществляет доступ к Корпоративному portalу. При осуществлении доступа к странице Корпоративного портала происходит следующая последовательность взаимодействий.

1. Пользователь открывает браузер на своем компьютере и переходит по ссылке к Корпоративному portalу.
2. Браузер устанавливает соединение с веб-сервером Internet Information Services (IIS).
3. IIS аутентифицирует пользователя в соответствии с используемым режимом аутентификации.
4. После аутентификации пользователя SharePoint проверяет, есть ли у пользователя права доступа к сайту.
5. В случае успешной авторизации пользователя запрос пересылается модулю SharePoint.
6. SharePoint извлекает данные о странице из базы данных SharePoint или из файловой системы. Эти данные содержат такую информацию, как шаблон страницы, разметка страницы, веб-части, которые должны на ней отображаться, и их свойства.
7. SharePoint обрабатывает страницу, создавая и инициализируя на ней веб-части, включая их свойства и данные пользовательской настройки страниц, если таковые присутствуют. Чтобы отобразить верхнюю навигационную панель, панель быстрого запуска и панель операций, поставщик навигации получает данные из Microsoft Dynamics AX (модули, меню, подменю и пункты меню).
8. При инициализации веб-частей Корпоративного портала иницируется веб-сеанс связи с инфраструктурой Корпоративного портала через .NET Business Connector и сервер приложения (AOS).
9. Инфраструктура Корпоративного портала проводит авторизацию пользователя в Microsoft Dynamics AX, **после чего вызывает соответствующие веб-обработчики объектов Корпоративного портала, на которые ссылаются отображаемые веб-части.**
10. Веб-часть User control запускает на выполнение тот веб-элемент управления, на который она ссылается, после чего данный веб-элемент под-

ключается через .NET Business Connector к Microsoft Dynamics AX, формирует HTML и возвращает его веб-части.

11. Веб-страница собирает все HTML-содержимое, сгенерированное всеми веб-частями на странице, после чего эта страница отображается в браузере пользователя.

Как видно из описания процесса обработки запроса веб-страницы, бизнес-логика и извлечение данных выполняются сервером приложения, элементы пользовательского интерфейса обрабатываются ASP.NET, а SharePoint отвечает за общий формат веб-страницы и ее пользовательскую настройку.

На рис. 7-2 графически изображена эта последовательность событий.

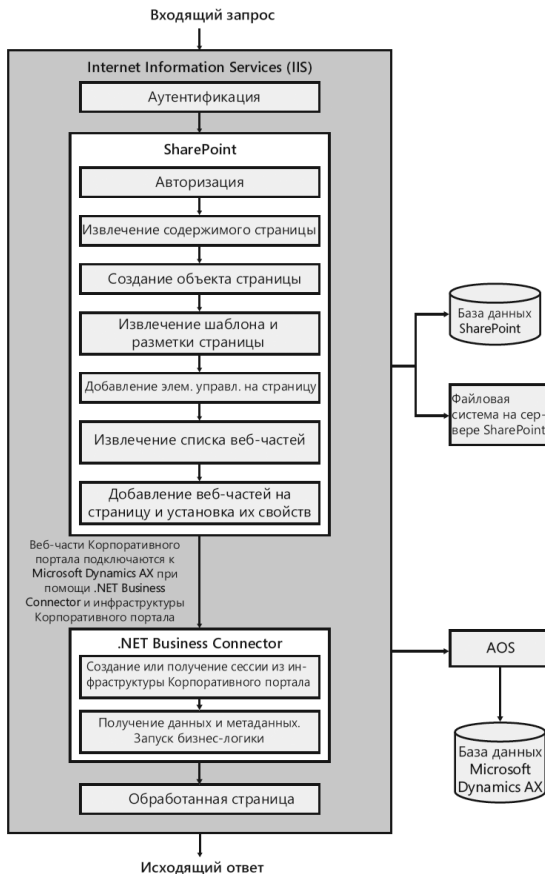


Рис. 7-2. Обработка страницы Корпоративного портала

Компоненты Корпоративного портала

В этом разделе описываются компоненты, составляющие страницу Корпоративного портала: веб-части, элементы репозитория прикладных объектов (АОТ), наборы данных (datasets) и элементы управления инфраструктуры Корпоративного портала.

Веб-части

Веб-части поддерживают модификацию и персонализацию и могут быть легко интегрированы в веб-страницу. Корпоративный портал включает стандартный набор веб-частей, показанных на рис. 7-3, которые открывают доступ к бизнес-данным Microsoft Dynamics AX.

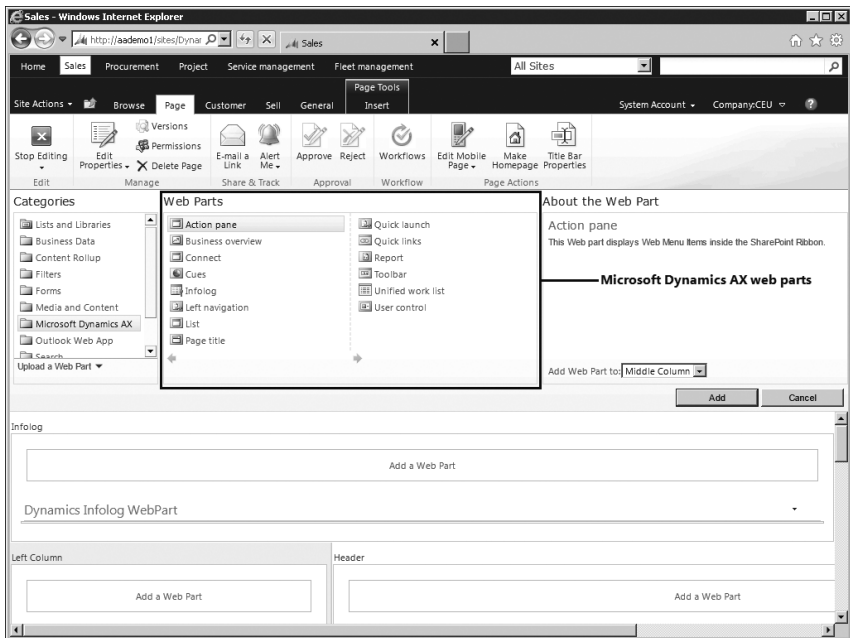


Рис. 7-3. Добавление веб-частей Microsoft Dynamics AX на страницу

С Корпоративным порталом поставляются следующие веб-части Microsoft Dynamics AX.

- **Action pane.** Используется для отображения панели операций, схожей с веб-частью SharePoint ribbon. Панель операций указывает на

веб-меню в АОТ и отображает кнопки во вкладках и группах, чтобы упростить нахождение нужных команд. В качестве альтернативы этой веб-части вы можете использовать элемент управления *AxActionPane* в рамках веб-элемента управления.

- **Business overview.** Используется для отображения в ролевых центрах информации бизнес-аналитики (BI), такой как ключевые показатели эффективности (KPI), и других аналитических данных. Более подробную информацию вы можете найти в главе 10.
- **Connect.** Используется для отображения ссылок на информацию из сообщества Microsoft Dynamics AX. Обычно эта веб-часть используется на страницах ролевых центров.
- **Cues.** Используется для визуального – в виде стопки бумаг – отображения числовой информации, такой как количество активных возможных сделок, новых интересов и т.д. Эта веб-часть обычно добавляется на страницы ролевых центров и указывает на группу Cue в АОТ. Более подробную информацию вы можете найти в главе 5.
- **Infolog.** Используется для отображения на веб-странице сообщений Infolog Microsoft Dynamics AX. Когда вы создаете новую страницу с веб-частями, используя шаблоны страниц Корпоративного портала, веб-часть Infolog автоматически добавляется в верхнюю часть страницы, в зону веб-части Infolog. Любые ошибки, предупреждения или информационные сообщения, которые формируются в Microsoft Dynamics AX, будут автоматически отображены с помощью этой веб-части. Если нужно отобразить в веб-части Infolog какую-либо информацию из вашего веб-элемента управления, то нужно отправить это сообщение через прокси-класс C# для объекта X++ *Infolog*.
- **Left navigation.** Используется для отображения навигации, специфичной для страницы – в противоположность навигации по модулю. Вы можете использовать эту веб-часть как альтернативу веб-части Quick launch, которая отображает элементы навигации по модулю. Данная веб-часть указывает на веб-меню в АОТ.
- **List.** Используется для отображения содержимого управляемой моделями страницы списка. Когда вы разворачиваете управляемую моделями страницу списка на Корпоративном портале, шаблон страницы автоматически добавляет веб-часть List в зону средней колонки стра-

ницы. Эта веб-часть указывает на пункт меню типа **display** для управляемой моделями формы страницы списка.

- **Page title.** Используется для отображения заголовка страницы. Когда вы создаете новую страницу с веб-частями, данная веб-часть автоматически добавляется в зону строки заголовка. По умолчанию веб-часть **Page title** отображает текст, указанный в свойстве *PageTitle* узла AOT *Page Definition* (определение страницы). Если такого узла нет, то отображается название страницы. Вы можете изменить это поведение и получать текст заголовка из другой веб-части на странице за счет использования связи веб-частей. Например, если вы разрабатываете страницу списка и хотите отображать в качестве заголовка информацию из записи, такую как код клиента, то можете связать веб-часть **User control**, отображающую элемент управления *grid*, с веб-частью **Page title**. При выборе пользователем другой записи в списке клиентов заголовок страницы будет меняться и отображать код выбранного клиента и его название.
- **Quick launch.** Используется для отображения в левой части страницы ссылок для навигации по текущему модулю. Когда вы создаете новую страницу с веб-частями, данная веб-часть автоматически добавляется в зону левой колонки страницы, если в шаблоне, который вы выбрали, есть такая зона. Веб-часть **Quick launch** отображает веб-меню, которое указано в свойстве *QuickLaunch* соответствующего веб-модуля в AOT. Все страницы одного веб-модуля (дочерний сайт) отображают на левой панели одни и те же ссылки.
- **Quick links.** Используется для отображения ссылок на часто используемые пункты меню и внешние веб-сайты. Обычно эту веб-часть добавляют на страницы ролевых центров.
- **Report.** Используется для отображения отчетов Microsoft SQL Server Reporting Services (SSRS), созданных для Microsoft Dynamics AX.
- **Toolbar.** Используется для отображения на странице панели инструментов, расположенной в том месте, где вам необходимо. К примеру, вы можете использовать веб-часть **Toolbar** для размещения прямо над элементом управления *grid* связанных с ним кнопок **Добавить**, **Правка** и **Удалить**. Веб-часть **Toolbar** указывает на веб-меню в AOT. В качестве альтернативы этой веб-части вы можете использовать элемент управления *AxToolbar* в пользовательском веб-элементе управления.

- **Unified work list.** Используется для отображения действий, оповещений и видов деятельности для документооборота. Обычно эту веб-часть добавляют на страницы ролевых центров. Более подробную информацию вы можете найти в главе 8.
- **User control.** Используется для размещения любых элементов управления ASP.NET, включая веб-элементы управления Microsoft Dynamics AX, разработанные вами. Эта веб-часть указывает на управляемый элемент веб-контента, который идентифицирует пользовательский веб-элемент управления. Веб-часть **User control** может как предоставлять другим веб-частям информацию о контексте записи (*record context*), так и использовать эту информацию, получаемую от них. Для управления установите роль данной веб-части в значение *Provider*, *Consumer* или *Both* и затем соедините ее с другими веб-частями. Веб-части такого типа автоматически используют AJAX, за счет чего они могут обновлять отображаемое ими содержимое без необходимости обновлять всю страницу целиком.

Элементы АОТ

В дополнение к таким программным элементам, как формы, классы и таблицы, АОТ содержит несколько других типов элементов, специфичных для Корпоративного портала. Более подробную информацию об элементах, доступных для создания страниц Корпоративного портала, вы можете найти в главе 1.

Наборы данных

Наборы данных (*datasets*) используются для определения логики доступа к данным. Набор данных – это совокупность данных, обычно представляемых в табличной форме. Они сочетают знакомую модель программирования и доступа к данным из форм Microsoft Dynamics AX с механизмами привязки к данным из ASP.NET. Более того, программная модель наборов данных позволяет писать на языке X++ логику проверки и манипуляции данными при создании, чтении, редактировании или удалении данных в Корпоративном портале. С помощью элемента управления *AxDataSource* вы можете использовать наборы данных для отображения и манипуляции данными из любого элемента управления ASP.NET, который поддерживает привязку к данным.

Набор данных создается в MorphX и может содержать один и более источников данных, каждый из которых ссылается на таблицу или представление Microsoft Dynamics AX. Источники данных могут объединять данные из нескольких таблиц в одном запросе. Для этого используются внутренние и внешние объединения таблиц. Для отображения данных вида родитель–потомок используются активные объединения записей. Чтобы показать данные из объединенных источников или из наборов вида родитель–потомок, используются представления наборов данных (класс *DataSetView*). В интерфейсе, основанном на представлениях, доступ к данным таблиц предоставляется с помощью динамических представлений наборов данных, а не напрямую через таблицы. Для доступа к данным таблиц, соединенных в запросе внешним или внутренним объединением, достаточно наличия одного представления с таким же названием, как и у основного источника данных в объединении. В случае двух источников данных, которые объединялись с помощью *active join*, присутствуют два представления: одно называется аналогично родительскому источнику данных, а другое – аналогично дочернему. Дочерний источник данных содержит только записи, которые относятся к текущей записи родительского источника. Каждое представление набора данных может содержать ноль или более записей, в зависимости от выбранных данных. Кроме того, у каждого представления набора данных есть соответствующее специальное представление, которое содержит лишь одну текущую запись; оно называется так же, как и исходное, но с суффиксом *_Current*.

На рис. 7-4 показаны представления набора данных в рамках набора данных, а также привязка данных с использованием наборов данных.

Как уже упоминалось, наборы данных предлагают обширную и знакомую программную модель на базе кода X++. Некоторые из часто используемых в процессе разработки методов включают *init*, *run*, *pack* и *unpack*.

- ***init***. Метод *init* вызывается при инициализации набора данных сразу после оператора *new* и создает экземпляр времени выполнения для набора данных. Обычно метод *init* используется для инициализации переменных и запросов, применения фильтров к выбираемым данным, проверки передаваемых аргументов.
- ***run***. Метод *run* вызывается по завершении инициализации и открытия набора данных, сразу после *init*. Обычно метод *run* содержит зависящую от определенных условий логику управления видимостью полей, изменения уровня доступа к ним, а также изменения запросов.

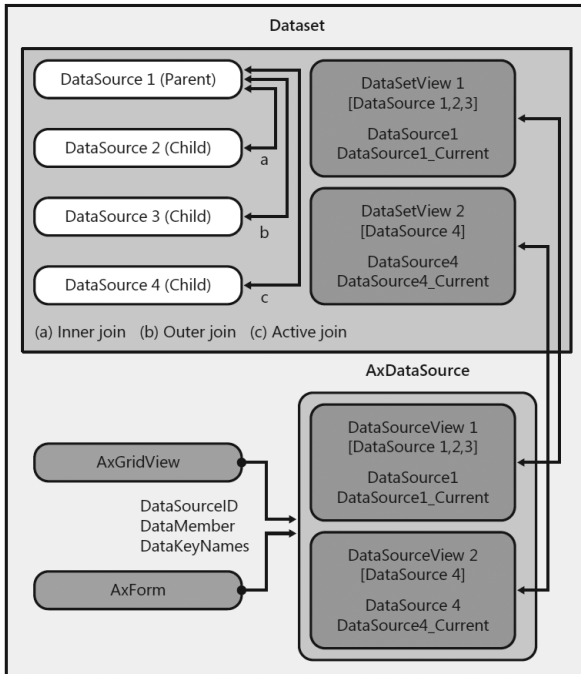


Рис. 7-4. Представления наборов данных Корпоративного портала

- **pack.** Метод *pack* вызывается после выполнения метода *run* набора данных. Обычно пара методов *pack/unpack* реализуется для сохранения состояния объекта и последующего его воссоздания. Метод *pack*, соответственно, применяется для сохранения значения переменной, используемой набором данных, между обработками действий элементов управления, вызывающих обратную передачу данных.
- **unpack.** Метод *unpack* вызывается при попытке доступа к набору данных, который был предварительно упакован с помощью метода *pack*. В этом случае методы *init* и *run* на наборе данных не вызываются, а вместо этого вызывается метод *unpack*.

Источники данных в наборе данных также содержат набор базовых методов, которые могут быть перекрыты. Эти методы схожи с аналогичными в классе *FormDataSource* клиента Microsoft Dynamics AX и могут использоваться для инициализации значений по умолчанию, а также проверки значений и действий на источниках данных. Более подробную информацию

об этих событиях, в том числе описание порядка вызова и сценариев использования, вы можете найти в разделе «Methods on a Form Data Source» в MSDN (<http://msdn.microsoft.com/en-us/library/aa893931.aspx>).

Элементы управления инфраструктуры Корпоративного портала

Инфраструктура Корпоративного портала включает набор встроенных элементов управления, которые вы можете использовать для осуществления доступа, отображения и манипуляции данными Microsoft Dynamics AX.

AxDataSource

Элемент управления *AxDataSource* расширяет возможности *DataSourceControl* из ASP.NET, предоставляя декларативный и не зависящий от хранилища данных способ читать и записывать данные Microsoft Dynamics AX. С помощью этого элемента управления наборы данных, которые вы создаете в AOT, становятся доступны в ASP.NET. Элементы пользовательского интерфейса ASP.NET можно связать с элементом управления *AxDataSource* с помощью их свойства *DataSourceID*. Использование стандартной привязки данных ASP.NET позволяет разрабатывать презентационную логику, не требуя от разработчика специфических знаний того, как подключиться и выбрать данные из Microsoft Dynamics AX.

Элемент управления *AxDataSource* является контейнером для одного или более уникально называющихся представлений типа *AxDataSourceView*. Класс *AxDataSourceView* наследует от класса *DataSourceView* из Microsoft .NET Framework и реализует функциональность чтения и записи данных. Элемент управления, привязанный к данным, по свойствам *AxDataSourceView* может определить перечень доступных возможностей и использовать эту информацию, чтобы отображать, скрывать, делать доступными или недоступными компоненты пользовательского интерфейса. *AxDataSourceView*, в свою очередь, указывает на представление набора данных из Microsoft Dynamics AX. Элемент управления *AxDataSource* автоматически создает объекты *AxDataSourceView* на основе набора данных, на который он ссылается. Количество создаваемых объектов зависит от источников данных и связей между ними, определенных в наборе данных. Для выбора конкретного представления используется свойство *DataMember* элемента интерфейса, привязанного к данным.

Элемент управления *AxDataSource* также обеспечивает поддержку фильтрации записей как внутри самого элемента источника данных, так и

между несколькими такими элементами. При установке активной записи на представлении источника данных в элементе *AxDataSource* все дочерние представления фильтруются в зависимости от активной записи. Вы можете осуществлять фильтрацию между элементами источников данных с помощью контекста записи. При использовании контекста записи один из элементов источника данных выступает провайдером контекста, а один или более элементов источника данных выступают в роли потребителей этого контекста. Один и тот же источник может одновременно быть и провайдером, и потребителем контекста записи. При смене текущей записи на элементе источника данных, который является провайдером контекста записи, потребители этого контекста получают данные новой записи и применяют соответствующие фильтры. Для указания, будет ли элемент *AxDataSource* выступать в качестве провайдера контекста, потребителя или в обеих ролях, используются свойства *Role* и *ProviderView*. Один пользовательский веб-элемент управления может содержать много элементов *AxDataSource*, однако только у одного из них свойство *Role* может быть установлено в значение *Provider*. При этом у любого количества элементов *AxDataSource* свойство *Role* может быть установлено в значение *Consumer*.

Объекты типа *DataSetViewRow* используются для доступа к строкам представления набора данных *DataSetView*. Текущую строку возвращает метод *GetCurrent()*, как показано в следующем примере:

```
DataSetViewRow row = this.AxDataSource1.GetDataSourceView("View1").DataSetView.  
GetCurrent();
```

Метод *GetDataSet()* объекта *AxDataSourceControl* указывает на набор данных, который привязывается управляющим элементам интерфейса. Свойство *DataSetRun* предоставляет доступ к экземпляру набора данных, чье свойство *AxaptaObjectAdapter* можно использовать для вызова методов этого набора данных.

```
this.AxDataSource1.GetDataSet().DataSetRun.AxaptaObjectAdapter.Call("method1");
```

AxForm

С помощью элемента управления *AxForm* вы можете предоставить пользователям возможность создавать, просматривать и обновлять отдельную запись. Этот элемент управления отображает одну запись из источника данных в виде формы с отдельными полями. *AxForm* – элемент управления с привязкой данных, имеющий встроенную поддержку модификации данных. При использовании этого элемента управления совместно с де-

кларативным элементом *AxDataSource* обеспечивается возможность отображения и модификации данных без единой строки кода. Возможности привязки к данным элемента управления *AxForm* определяются свойствами *DataSourceID*, *DataMember* и *DataKeyNames*. Кроме того, этот элемент управления предоставляет свойства для автогенерации кнопок действий с возможностью указания на них текста, а также их состояния. Если необходимо обновить курсор записи при обратной передаче данных, нужно установить свойство *UpdateOnPostback*; это позволит другим элементам управления считать измененные данные записи. *AxForm* также предоставляет события для всех доступных на форме действий, как до вызова действия, так и после. В обработчиках этих событий вы можете написать код, изменяющий интерфейсные элементы или реализующий иную прикладную логику.

AxMultiSection

Элемент управления *AxMultiSection* выступает в качестве контейнера для набора элементов управления *AxSection* (см. следующий раздел). Все элементы *AxSection* внутри одного элемента *AxMultiSection* отображаются в виде строк, расположенных одна под другой, которые пользователь может разворачивать и сворачивать. Вы можете настроить *AxMultiSection* так, чтобы в любой момент времени развернутой была лишь одна секция. В таком режиме разворачивание секции приводит к тому, что она становится активной, а любая ранее развернутая секция сворачивается. Для включения этого режима работы установите свойство *ActiveMode* в *true*. Затем вы можете использовать свойство *ActiveSectionIndex*, чтобы получить или установить индекс активной секции.

AxSection

AxSection – это контейнер общего назначения для других элементов управления. Внутри *AxSection* вы можете поместить любой другой элемент управления. Каждый элемент управления *AxSection* включает заголовок, содержащий название секции, и кнопку, с помощью которой пользователь может разворачивать и сворачивать секцию. У *AxSection* есть свойства, с помощью которых можно отображать и скрывать заголовок и обрамляющую рамку. *AxSection* предоставляет события, в обработчиках которых вы можете писать код, выполняющийся при разворачивании и сворачивании секции. Элемент управления *AxSection* может содержаться только в элементе управления *AxMultiSection*.

AxMultiColumn

Элемент управления *AxMultiColumn* выступает в качестве контейнера для набора элементов управления *AxColumn*. Все элементы *AxColumn* внутри одного элемента *AxMultiColumn* отображаются в виде нескольких колонок. С помощью *AxMultiColumn* вы можете легко разбить дизайн веб-формы или страницы на несколько колонок, чтобы оптимально использовать пространство на экране. Обычно *AxMultiColumn* размещается внутри элемента управления *AxSection*.

AxColumn

AxColumn – это контейнер общего назначения для других элементов управления. Внутри *AxColumn* вы можете поместить любой другой элемент управления, однако сам *AxColumn* может быть помещен лишь внутри элемента управления *AxMultiColumn*.

AxGroup

Элемент управления *AxGroup* содержит набор полей, привязанных к данным, которые отображают информацию из записи таблицы. Вы можете поместить *AxGroup* внутри элементов управления *AxSection* или *AxColumn*.

На рис. 7-5 показана страница Корпоративного портала, содержащая несколько элементов управления, которые были описаны выше.

Ниже приведены некоторые абстрактные структуры вложенных элементов управления для различных дизайнов форм. Первым идет пример достаточно распространенного в Корпоративном портале шаблона. В нем отображаются две сворачиваемые секции, одна под другой, в каждой из которых поля отображаются в две колонки. Если вы хотите отображать дополнительные секции или колонки, то можете добавить новые элементы управления *AxSection* и *AxColumn*.

```
<AxMultiSection>
  <AxSection>
    <AxMultiColumn>
      <AxColumn>
        <AxGroup><Fields>BoundFields or TemplateFields...</Fields></AxGroup>
      </AxColumn>
      <AxColumn>
        <AxGroup><Fields>BoundFields or TemplateFields...</Fields></AxGroup>
      </AxColumn>
    </AxMultiColumn>
  </AxSection>
</AxSection>
```

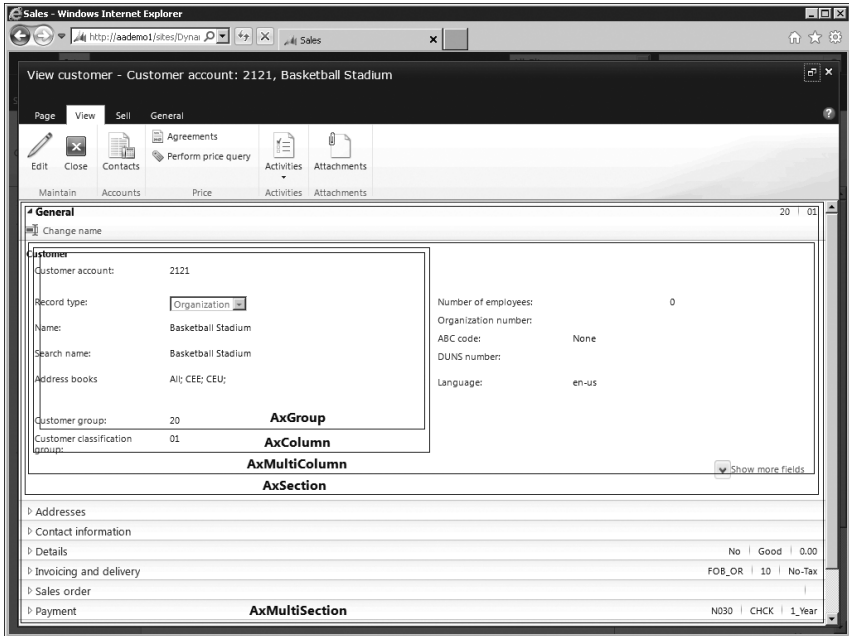


Рис. 7-5. Страница Корпоративного портала, содержащая элементы управления: секцию, колонку и группу

```

<AxMultiColumn>
  <AxColumn>
    <AxGroup><Fields>BoundFields or TemplateFields...</Fields></AxGroup>
  </AxColumn>
  <AxColumn>
    <AxGroup><Fields>BoundFields or TemplateFields...</Fields></AxGroup>
  </AxColumn>
</AxMultiColumn>
</AxSection>
</AxMultiSection>

```

Следующий пример разметки – это две сворачиваемые секции, одна под другой, отображающие поля в одну колонку.

```

<AxMultiSection>
  <AxSection>
    <AxGroup><Fields>BoundFields or TemplateFields...</Fields></AxGroup>
    <AxGroup><Fields>BoundFields or TemplateFields...</Fields></AxGroup>
  </AxSection>

```



```

<AxSection>
  <AxGroup><Fields>BoundFields or TemplateFields...</Fields></AxGroup>
  <AxGroup><Fields>BoundFields or TemplateFields...</Fields></AxGroup>
</AxSection>
</AxMultiSection>

```

Следующий пример разметки – это мастер ASP.NET с двумя шагами выполнения.

```

<asp:Wizard>
  <WizardSteps>
    <asp:WizardStep>
      <AxGroup><Fields>BoundFields or TemplateFields...</Fields></AxGroup>
    </asp:WizardStep>
    <asp:WizardStep>
      <AxGroup><Fields>BoundFields or TemplateFields...</Fields></AxGroup>
    </asp:WizardStep>
  </WizardSteps>
</asp:Wizard>

```

AxGridView

Элемент управления *AxGridView* отображает значения из источника данных в табличной форме. Каждая колонка представляет определенное поле, а каждая строка – определенную запись. Элемент управления *AxGridView* наследует от *GridView* из ASP.NET и, таким образом, предоставляет возможности для выделения и группировки строк, развернутого представления строк, фильтрации записей, отображения контекстного меню, а также другие расширенные возможности.

Кроме того, у *AxGridView* есть встроенные возможности редактирования данных. При использовании *AxGridView* совместно с декларативным элементом *AxDataSource* обеспечивается возможность отображения и модификации данных без единой строки кода. Помимо этого, у *AxGridView* есть множество свойств, методов и событий, которые вы можете использовать для реализации необходимой презентационной логики.

В табл. 7-1 представлены некоторые из свойств и событий *AxGridView*. Полный список свойств, методов и событий *AxGridView* вы можете найти в одноименном разделе MSDN по адресу: <http://msdn.microsoft.com/en-us/library/cc584514.aspx>.

Табл. 7–1. Свойства и события *AxGridView*

Свойство или событие	Описание
<i>AllowDelete</i>	Если включено и если у пользователя есть право на удаление выбранной записи, то <i>AxGridView</i> отображает кнопку Удалить, позволяющую удалить выбранную строку
<i>AllowEdit</i>	Если включено и если у пользователя есть право на обновление выбранной записи, то <i>AxGridView</i> отображает кнопки Сохранить и Отмена для выбранной строки, а также позволяет пользователю произвести изменения и сохранить их или отменить. При выборе строки, она автоматически переходит в режим редактирования, и для всех полей выбранной записи, для которых свойство <i>AllowEdit</i> установлено в <i>true</i> , отображаются соответствующие поля ввода
<i>AllowGroupCollapse</i>	Если включена группировка, это свойство разрешает пользователю свернуть сгруппированные строки
<i>AllowGrouping</i>	Если установлено в <i>true</i> и указано поле для группировки, строки отображаются группами и сортируются по значению поля группировки. Поддерживается настройка размера страницы, так что одна группа может быть разбита на несколько страниц
<i>AllowSelection</i>	Если установлено в <i>true</i> , то пользователь может выбрать отдельную запись
<i>ContextMenuName</i>	Если включено свойство <i>ShowContextMenu</i> , то <i>ContextMenuName</i> содержит название веб-меню в АОТ, которое должно использоваться в качестве контекстного меню при щелчке правой кнопкой мыши по выбранной строке
<i>DataBound</i>	Событие, срабатывающее после привязки элемента управления к источнику данных
<i>DataMember</i>	Указывает таблицу или набор данных, к которому привязан элемент <i>grid</i>
<i>DataSourceID</i>	Указывает элемент управления <i>AxDataSource</i> , который используется для получения данных
<i>DisplayGroupFieldName</i>	Если установлено в <i>true</i> , то для сгруппированного представления в тексте заголовка группы отображается <i>GroupFieldName</i>
<i>ExpansionColumnIndexesHidden</i>	Список целых чисел, разделенных запятыми, представляющий номера колонок, которые необходимо скрыть в развернутом представлении строки. Номера колонок начинаются с 1

Табл. 7-1. Свойства и события *AxGridView* (окончание)

Свойство или событие	Описание
<i>Expansion-Tooltip</i>	Текст подсказки, отображаемый для ссылки на развернутое представление строки
<i>GridColumnIndexesHidden</i>	Список целых чисел, разделенных запятыми, представляющий номера колонок, которые необходимо скрыть в строке элемента <i>grid</i> . Номера колонок начинаются с 1
<i>GroupField</i>	Указывает поле данных, которое используется для группировки строк в режиме сгруппированного представления данных элемента <i>grid</i> . Это свойство используется только в случае, когда <i>AllowGrouping</i> установлено в <i>true</i>
<i>GroupFieldDisplayName</i>	Позволяет получить или установить название для поля группировки, отображаемое в тексте заголовка группы в режиме сгруппированного представления данных. Если не указано, то по умолчанию используется метка поля <i>GroupField</i>
<i>Row*</i>	События, которые срабатывают в ответ на различные действия со строкой. К примеру, <i>RowCommand</i> срабатывает, когда нажата кнопка, привязанная к строке, а <i>RowDeleted</i> срабатывает, когда для строки нажата кнопка Удалить
<i>SelectedIndexChanged</i>	Событие, срабатывающее при выделении строки элемента <i>grid</i>
<i>ShowContextMenu</i>	Если установлено в <i>true</i> , то при щелчке правой кнопкой мыши по выбранной строке отображается меню <i>ContextMenuName</i>
<i>ShowExpansion</i>	Указывает, доступно ли развернутое представление для каждой строки элемента <i>grid</i> . Когда установлено в <i>true</i> , для каждой строки элемента <i>grid</i> отображается ссылка на развернутое представление строки
<i>ShowFilter</i>	Если установлено в <i>true</i> , отображает над элементом <i>grid</i> элемент управления для фильтрации записей

AxHierarchicalGridView

Используйте элемент управления *AxHierarchicalGridView*, когда хотите отобразить иерархические данные в табличной форме. К примеру, у вас может быть элемент *grid*, отображающий список задач проекта. У каждой задачи могут быть подзадачи, и с помощью этого элемента управления все задачи и подзадачи можно представить в одном элементе *grid*, как показано на рис. 7-6.

Title	StartDate	EndDate
Task A	1/4/2011	2/10/2011
Task B	1/4/2011	1/29/2011
Task C	1/9/2011	2/10/2011
Task D	1/20/2011	5/1/2011
Task E	1/20/2011	3/2/2011
Task G	3/5/2011	4/12/2011
Task F	2/15/2011	4/12/2011

Рис. 7-6. Пример использования *AxHierarchicalGridView* в пользовательском интерфейсе

Поле, уникально идентифицирующее строку, задается в свойстве *HierarchyIdFieldName*, а идентифицирующее родительскую строку – в свойстве *HierarchyParentIdFieldName*. В следующем примере представлена разметка для элемента управления *AxHierarchicalGridView*.

```
<dynamics:AxDataSource ID="AxDataSource1" runat="server" DataSetName="Tasks"
ProviderView="Tasks">
</dynamics:AxDataSource>
<dynamics:AxHierarchicalGridView ID="AxHierarchicalGridView1" runat="server"
BodyHeight="" DataKeyNames="RecId" DataMember="Tasks"
DataSetCachingKey="e779ece0-43b7-4270-9dc9-33f4c61d42b7"
DataSourceID="AxDataSource1" EnableModelValidation="True"
HierarchyIdFieldName="TaskId" HierarchyParentIdFieldName="ParentTaskId">
<Columns>
<dynamics:AxBoundField DataField="Title" DataSet="Tasks"
DataSetView="Tasks" SortExpression="Title">
</dynamics:AxBoundField>
<dynamics:AxBoundField DataField="StartDate" DataSet="Tasks"
DataSetView="Tasks" SortExpression="StartDate">
</dynamics:AxBoundField>
<dynamics:AxBoundField DataField="EndDate" DataSet="Tasks"
DataSetView="Tasks" SortExpression="EndDate">
</dynamics:AxBoundField>
</Columns>
</dynamics:AxHierarchicalGridView>
```

AxContextMenu

Используйте элемент управления *AxContextMenu* для создания и отображения контекстного меню. Этот элемент управления предоставляет методы для добавления и удаления во время выполнения пунктов меню и разделителей. Он также предоставляет методы для распознавания ссылок клиента или Корпоративного портала, как показано в следующем примере.

```
AxUrlMenuItem myUrlMenuItem = new AxUrlMenuItem("MyUrlMenuItem");
AxContextMenu myContextMenu = new AxContextMenu();
myContextMenu.AddMenuItemAt(0, myUrlMenuItem);
```

AxGridView использует *AxContextMenu* в случае, если свойство *ShowContextMenu* установлено в *true*. Вы можете получить доступ к объекту *AxContextMenu*, используя синтаксис *AxGridView.ContextMenu*.

AxFilter

Используйте элемент управления *AxFilter* для фильтрации данных, получаемых из источника данных. Элемент управления фильтрации устанавливает фильтр на объект *AxDataSetView* через объект *AxDataSourceView*, который отвечает за синхронизацию данных с условиями фильтрации. Для этого при изменении данных вызываются методы *SetAsChanged* и *ExecuteQuery*. На объектах *AxDataSourceView* и *DataSetView* доступны следующие свойства для программного управления фильтрацией данных.

- **SystemFilter.** Помещает в свойство *conditionCollection* объекта фильтрации полный список диапазонов запроса, включая открытые, скрытые и заблокированные.
- **UserFilter.** Помещает в свойство *conditionCollection* объекта фильтрации только открытые диапазоны из свойства *QueryRun*.
- **ResetFilter.** Очищает фильтр, установленный в свойстве *QueryRun*, и таким образом сбрасывает фильтр (диапазоны запроса), установленный программно.

В коде X++ вы можете установить диапазон запроса в источнике данных следующим образом.

```
qbrBlocked = qbds.addRange(fieldnum(CustTable,Blocked));
qbrBlocked.value(queryValue(CustVendorBlocked::No));
qbrBlocked.status(RangeStatus::Hidden);
```

В веб-элементе управления *Web User Control* можно получить значение фильтра, установленного на источнике данных, одним из следующих способов.

```
this.AxDataSource1.GetDataSourceView(this.AxGridView1.DataMember).SystemFilter.ToXml();
```

или

```
this.AxDataSource1.GetDataSet().DataSetViews[this.AxGridView1.DataMember].SystemFilter.ToXml();
```

Возвращаемое значение будет выглядеть примерно так.

```
<?xml version="1.0" encoding="utf-16"?><filter xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="CustTable"><condition attribute="Blocked" operator="eq" value="No" status="hidden" /></filter>
```

Вы также можете установить фильтр программно.

```
string myFilterXml = @"<filter name='CustTable'><condition attribute='CustGroup' status='open' value='10' operator='eq' /></filter>";  
this.AxDataSource1.GetDataSourceView(this.AxGridView1.DataMember).SystemFilter.  
AddXml(myFilterXml);
```

Элемент управления *AxGridView* также использует *AxFilter*, когда свойство *ShowFilter* установлено в *true*. Вы можете получить доступ к объекту *AxFilter* через *AxGridView.FilterControl*, а к XML фильтра – через *AxGridView.Filter*. Фильтр считывает метаданные из компонента *AxDataSourceControl*, к данным которого привязан компонент *AxGridView*, и динамически отображает элементы управления, позволяя пользователю фильтровать данные источника по любому из его полей, кроме тех, которые скрыты или заблокированы для фильтрации. Эти элементы управления отображаются над элементом *grid*.

AxLookup

Используйте элемент управления *AxLookup* на страницах ввода данных, чтобы помочь пользователю выбирать корректные значения полей, которые ссылаются на ключевые значения из других таблиц. В Корпоративном портале формы таблиц подстановки по умолчанию основаны на метаданных и автоматически доступны для всех полей, для которых в АОТ настроено отношение с другой таблицей. Примером является форма таблицы подстановки для группы клиента на странице создания клиента. Метаданные расширенного типа данных (EDT) и отношения между таблицами в АОТ определяют отношение между таблицей клиентов и таблицей групп клиентов, поэтому при создании записи нового клиента в соответствующем поле автоматически отображается форма таблицы подстановок для выбора группы клиентов. Для активации такого поведения не требуется написания кода – все происходит автоматически.

Однако в некоторых случаях такого автоматического поведения недостаточно, и требуется перекрыть метод отображения формы таблицы подстановок. Инфраструктура Корпоративного портала позволяет разработ-

чикам менять поведение, интерфейс и данные, возвращаемые из формы таблицы подстановок, как из кода X++, так и из кода C#.

Для управления поведением формы таблицы подстановок можно в узле *Data Sets* АОТ перекрыть метод *dataSetLookup* на поле источника данных. К примеру, если необходимо добавить фильтрацию отображаемых данных, то можно перекрыть метод *dataSetLookup*, как показано в следующем примере на X++.

```
void dataSetLookup(SysDataSetLookup _sysDataSetLookup)
{
    List                list;
    Query               query = new Query();
    QueryBuildDataSource queryBuildDataSource;
    Args                args;

    args = new Args();
    list = new List(Types::String);
    list.addEnd(fieldstr(HcmGoalHeading, GoalHeadingId));
    list.addEnd(fieldstr(HcmGoalHeading, Description));

    queryBuildDataSource = query.addDataSource(tablenum(HcmGoalHeading));

    queryBuildDataSource.addRange(fieldnum(HcmGoalHeading, Active)).value(
        queryValue(NoYes::Yes));

    _sysDataSetLookup.parmLookupFields(list);
    _sysDataSetLookup.parmSelectField(fieldStr(HcmGoalHeading, GoalHeadingId));

    // Pass the query to SysDataSetLookup so it result is rendered in the lookup page.
    _sysDataSetLookup.parmQuery(query);
}
```

В предыдущем примере вся таблица подстановок строится динамически, а для ограничения возможных значений поля используется метод *addRange*. Класс *SysDataSetLookup* в X++ предоставляет множество свойств и методов для управления поведением формы таблицы подстановок.

Формы таблицы подстановки можно изменять и на языке C# путем написания кода в методе-обработчике события *Lookup* на полях, привязанных к данным, или с помощью элемента *AxLookup* на полях, которые не привязаны к данным. Для использования элемента *AxLookup*, который позволяет отображать формы таблицы подстановки на управляющих элементах ASP.NET, не привязанных к источнику данных, необходимо

выбрать этот управляющий элемент ASP.NET в свойстве *TargetControlID* элемента *AxLookup*. Также основой для *AxLookup* может служить расширенный тип данных, набор данных, пользовательский набор данных или пользовательский элемент управления. Для этого необходимо указать выбранное значение в свойстве *LookupType*. С помощью этого элемента также можно выбрать поля, которые будут отображаться в форме таблицы подстановок, а также поле, значение которого необходимо вернуть в вызывающий элемент управления. Кроме этого, можно перекрыть метод-обработчик события *Lookup* и контролировать поведение формы таблицы подстановок из кода, как в следующем примере.

```
protected void AxLookup1_Lookup(object sender, AxLookupEventArgs e)
{
    AxLookup lookup = (AxLookup)sender;

    // Specify the lookup fields
    lookup.Fields.Add(AxBoundFieldFactory.Create(this.AxSession,
        lookup.LookupDataSetViewMetadata.ViewFields["CustGroup"]));

    lookup.Fields.Add(AxBoundFieldFactory.Create(this.AxSession,
        lookup.LookupDataSetViewMetadata.ViewFields["Name"]));
}
```

AxActionPane

Элемент управления *AxActionPane* выполняет функцию, схожую с веб-частью *Action pane*. Вы можете использовать его для отображения панели операций в верхней части страницы аналогично ленте (ribbon) *SharePoint*. Используйте свойство *WebMenuName* элемента управления *AxActionPane* для указания веб-меню, содержащего пункты меню, которые нужно отобразить на панели операций в виде кнопок. Чтобы упростить поиск кнопок для пользователя, они отображаются на вкладках и в группах. Чтобы связать кнопки панели операций с данными, вы можете использовать свойства *DataSource* и *DataMember* элемента управления *AxActionPane*.

Для использования элемента управления *AxActionPane* в пользовательском веб-элементе управления необходимо добавить ссылку на сборку *Microsoft.Dynamics.Framework.Portal.SharePoint*. Это можно сделать с помощью приведенных ниже строк разметки для пользовательского веб-элемента управления.

```
<%@ Register Assembly="Microsoft.Dynamics.Framework.Portal.SharePoint, Version=6.0.0.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35" Namespace="Microsoft.Dynamics.
Framework.Portal.SharePoint.UI.WebControls" TagPrefix="dynamics" %>
```


При желании в качестве альтернативы элементу управления *AxAction-Pane* вы можете использовать веб-часть *Action pane*.

AxToolbar

Элемент управления *AxToolbar* выполняет функцию, схожую с веб-частью *Toolbar*. Вы можете использовать его для отображения панели инструментов в нужной части страницы вместо использования панели операций вверху страницы. К примеру, вы можете предпочесть отображать панель инструментов с действиями Создать, Правка и Удалить над элементом управления *grid*.

Для непосредственной реализации своих функций *AxToolbar* использует элементы управления панели инструментов *SharePoint*. *AxToolbarButton*, используемый внутри *AxToolbar*, наследует от *SPLinkButton*. Он применяется для отображения кнопок верхнего уровня. Аналогично *AxToolBarMenu*, используемый внутри *AxToolbar*, наследует от *Microsoft.SharePoint.WebControls.Menu*. Этот элемент управления с использованием функции обратного вызова отображает выпадающее меню, когда пользователь щелкает по кнопке. За счет этого перед отображением пунктов меню их свойства могут быть изменены.

При желании в качестве альтернативы элементу управления *AxToolbar* вы можете использовать веб-часть *Toolbar*. Она используется в большинстве случаев, когда необходимо управлять отображаемыми пунктами панели инструментов. Однако для страниц заданий, на которых отображаются данные из двух таблиц, связанных как родитель–потомок, таких как заголовки и строки заявок на покупку, следует над управляющим элементом *AxGridView* строк заявок использовать элемент управления *AxToolbar*, что позволит пользователю добавлять и редактировать строки заявок.

Элемент *AxToolbar* может быть привязан к источнику данных *AxDataSource*. В таком случае контекст выбранного пункта меню автоматически определяется на основании выбранной записи. В случае, когда элемент панели инструментов не привязан к данным, для управления контекстом придется дополнительно писать код.

При определении панели инструментов можно указать в свойстве *WebMenuName* ссылку на веб-меню в АОТ. Веб-меню позволяют определить многоуровневую структуру меню путем добавления на него элементов подменю *SubMenu*, ссылок меню *MenuReference* и пунктов меню *MenuItem*. Каждый пункт меню верхнего уровня с помощью элемента управления *AxToolbarButton* отображается в виде ссылки. Каждое подме-

ню верхнего уровня с помощью элемента управления *AxToolbarMenu* отображается в виде выпадающего меню. При наличии вложенных подменю второй и последующие уровни отображаются как подменю.

SetMenuItemProperties, *ActionMenuItemClicking* и *ActionMenuItemClicked* — это события на *AxToolbar*, связанные с элементами управления этой панели инструментов. Событие *SetMenuItemProperties* используется для изменения поведения выпадающих меню, например сокрытия или показа пунктов меню в зависимости от выбранной в данный момент записи, добавления или удаления контекста записи и т.д. В следующем примере кода показано, как можно изменять контекст записи пункта меню в методе-обработчике события *SetMenuItemProperties*.

```
void Webpart_SetMenuItemProperties(object sender, SetMenuItemPropertiesEventArgs e)
{
    // Не передавать контекст записи выбранной записи клиента, так как данный пункт
    // меню используется для создания новых клиентов (строка запроса должна быть
    // пустой)
    if (e.MenuItem.MenuItemAOTName == "EPCustTableCreate")
    {
        ((AxUrlMenuItem)e.MenuItem).MenuItemContext = null;
    }
}
```

Если предпочтительная логика работы интерфейса пользователя для определенного пункта меню панели инструментов определена в элементе *Web User Control* и необходимо вызвать эту логику вместо той, которая определена в AOT, используются обработчики событий *ActionMenuItemClicking* и *ActionMenuItemClicked*. К примеру, с помощью события *ActionMenuItemClicking* можно предотвратить вызов действия, определенного в AOT. Вместо этого можно написать обработку этого события в коде C#, в методе *ActionMenuItemClicked*, как показано ниже.

```
void webpart_ActionMenuItemClicking(object sender, ActionMenuItemClickingEventArgs e)
{
    if (e.MenuItem.MenuItemAOTName.ToLower() == "EPCustTableDelete")
    {
        e.RunMenuItem = false;
    }
}

void webpart_ActionMenuItemClicked(object sender, ActionMenuItemEventArgs e)
{
}
```

```

        if (e.MenuItem.MenuItemAOTName.ToLower() == "EPCustTableDelete")
        {
            int selectedIndex = this.AxGridView1.SelectedIndex;

            if (selectedIndex != -1)
            {
                this.AxGridView1.DeleteRow(selectedIndex);
            }
        }
    }
}

```

AxPopup

Используйте элемент управления *AxPopup* для открытия страницы во всплывающем окне браузера, закрытия всплывающего окна, а также для передачи данных между страницей всплывающего окна и вызывающей веб-страницей с последующим вызовом обработчика серверного события *PopupClosed*. Данная функциональность инкапсулирована в двух элементах: *AxPopupParentControl*, который используется на вызывающей родительской странице, и *AxPopupChildControl*, который используется на самой странице всплывающего окна. Оба этих элемента управления наследуют от *AxPopupBaseControl*. Данные элементы управления совместимы с AJAX, поэтому создавать их можно по определенному условию при частичном обновлении страницы.

Элемент *AxPopupParentControl* позволяет странице, обычно содержащей какую-то веб-часть, открыться во всплывающем окне браузера. Всплывающее окно можно открыть из сценария на стороне клиента с помощью метода *GetOpenPopupEventReference*. Возвращаемая методом строка является выражением JavaScript, которое может быть присвоено атрибуту кнопки *onclick* или пункту меню панели инструментов. В следующем примере показано, как открыть всплывающее окно с помощью сценариев на стороне клиента путем изменения события *OnClick*.

```

protected void SetPopupWindowToMenuItem(SetMenuItemPropertiesEventArgs e)
{
    AxUriMenuItem menuItem = new AxUriMenuItem("EPCustTableCreate");

    // вызываем функцию JavaScript для установки свойств веб-страницы,
    // вызываемой пунктом меню
    e.MenuItem.ClientOnClickScript =
        this.AxPopupParentControl1.GetOpenPopupEventReference(menuItem);
}

```

Всплывающее окно также можно открыть из кода, выполняющегося на стороне сервера, с помощью метода *OpenPopup*. Поскольку всплывающие окна, вызываемые сервером, часто блокируются браузером, метод *OpenPopup* следует использовать только в случае крайней необходимости, к примеру, когда только в серверном коде может быть принято решение об открытии всплывающего окна.

Элемент *AxPopupChildControl* при помещении на страницу всплывающего окна позволяет контролировать закрытие этой страницы. Со стороны клиента всплывающее окно можно закрыть с помощью метода *GetClosePopupEventReference*, как показано в примере ниже.

```
this.BtnOk.Attributes.Add("onclick",  
    this.popupChild.GetClosePopupEventReference(true, true) + "; return false;");
```

Со стороны сервера всплывающее окно можно закрыть с помощью метода *ClosePopup*. Используйте серверный метод для выполнения дополнительной обработки, если таковая необходима при закрытии окна, к примеру, для выполнения какой-либо операции или расчета значений, которые потом передаются назад в вызывающую страницу. Методы *ClosePopup* и *OpenPopup* принимают на вход два параметра.

- ***setFieldValues***. Значение *true* указывает на необходимость передачи каких-либо данных назад в вызывающую страницу.
- ***updateParent***. Значение *true* указывает на необходимость выполнения обратной передачи для вызывающей страницы. Элемент *AxPopupChildControl* посылает (с помощью клиентского сценария) вызывающей странице запрос на обратную передачу, которая должна быть выполнена элементом *AxPopupParentControl*. После этого элемент *AxPopupParentControl* генерирует серверное событие *PopupClosed*, по которому прикладной код вызывающей страницы получает доступ к значениям, переданным из всплывающего окна, и может выполнить какую-либо операцию или просто обновить свое состояние.

Для передачи данных со страницы всплывающего окна назад в вызывающую страницу используются объекты типа *AxPopupField*. Эти объекты доступны через свойство *Fields* элемента *AxPopupBaseControl*, от которого унаследованы как *AxPopupParentControl*, так и *AxPopupChildControl*.

AxPopupParentControl и *AxPopupChildControl* содержат поля с одинаковыми названиями. При закрытии страницы всплывающего окна значение каждого поля элемента *AxPopupChildControl* присваивается

(с помощью клиентского сценария) соответствующему полю элемента *AxPopupParentControl*.

AxPopupField может быть связан с другим элементом управления, таким как *TextBox* или любым другим, путем установки идентификатора этого управляющего элемента в качестве значения свойства *TargetId* на элементе *AxPopupField*. Это может быть полезно, к примеру, когда на странице всплывающего окна присутствует элемент управления типа *TextBox*. Для передачи значений ввода пользователя в вызывающую страницу при закрытии страницы всплывающего окна, необходимо связать поле с элементом *TextBox* на странице всплывающего окна. При этом операция передачи значения будет полностью выполнена на стороне клиента во избежание лишних обращений к серверу. Если к элементу *AxPopupField* явно не привязан какой-либо элемент управления, то он неявно связывается с автоматически создаваемым (элементом *AxPopupParentControl* или *AxPopupChildControl*) элементом управления *HiddenField*.

После этого значение поля на стороне сервера можно установить с помощью метода *SetFieldValue*. Обычно метод *SetFieldValue* вызывается на элементе *AxPopupChildControl*, причем вызвать его можно в любой момент взаимодействия пользователя со страницей всплывающего окна, включая время инициализации или закрытия страницы. Значение поля можно получить с помощью метода *GetFieldValue*. Обычно метод *GetFieldValue* вызывается на элементе *AxPopupParentControl* во время обработки события *PopupClosed*. Значения полей, не связанных с элементами управления, можно очистить с помощью вызова метода *ClearFieldValues*.

Значениями полей *AxPopupFields* можно также манипулировать и на стороне клиента, изменяя значения связанных управляющих элементов. Объект связанного управляющего элемента (будь то явная или неявная связь) доступен через свойство *TargetControl*.

Элементы управления *BoundField*

Элементы *BoundField* используются элементами управления, привязанными к данным (такими как *AxGridView*, *AxGroup*, ASP.NET *GridView* и *DetailsView*), для отображения значения поля с помощью привязки данных. Отображение этих элементов в интерфейсе пользователя зависит от того элемента управления, в котором они используются. К примеру, *AxGridView* отображает их в виде столбцов, тогда как *AxGroup* – в виде строки.

Инфраструктура Корпоративного портала содержит набор усовершенствованных элементов *AxBoundField*, которые наследуют от стандарт-

ных элементов ASP.NET *BoundField*, но содержат интеграцию с метаданными Microsoft Dynamics AX.

В табл. 7-2 описаны основные элементы *AxBoundField*.

Табл. 7-2. Элементы *AxBoundField*

Элемент	Описание
<i>AxBoundField</i>	Используется для отображения текстовых значений. Свойства <i>DataSet</i> , <i>DataSetView</i> и <i>DataField</i> определяют источник данных для этого элемента
<i>AxHyperLink-BoundField</i>	Используется для отображения гиперссылок. Свойство <i>MenuItem</i> данного элемента используется для указания веб-пункта меню в AOT для определения URL-перехода, а свойства <i>DataSet</i> , <i>DataSetView</i> и <i>DataField</i> используются для источника данных для этого элемента. В случае если название веб-пункта меню хранится в выбранной записи данных, то вместо свойства <i>MenuItem</i> используется свойство <i>DataMenuItemField</i>
<i>AxBound-FieldGroup</i>	Используется для отображения табличных групп полей, заданных в AOT. Свойства <i>DataSet</i> , <i>DataSetView</i> и <i>FieldGroup</i> определяют источник данных для этого элемента
<i>AxCheckBox-BoundField</i>	Используется для отображения кнопки флажка логического типа. Свойства <i>DataSet</i> , <i>DataSetView</i> и <i>FieldGroup</i> определяют источник данных для этого элемента
<i>AxDropDown-BoundField</i>	Используется для отображения набора значений в выпадающем списке. Свойства <i>DataSet</i> , <i>DataSetView</i> и <i>FieldGroup</i> определяют источник данных для этого элемента
<i>AxRadioButton-BoundField</i>	Используется для отображения набора значений в виде радиокнопок. Свойства <i>DataSet</i> , <i>DataSetView</i> и <i>FieldGroup</i> определяют источник данных для этого элемента. С помощью свойства <i>RepeatDirection</i> вы можете указать, должны элементы выводиться горизонтально или вертикально
<i>AxReference-BoundField</i>	Используется для суррогатного ключа. Суррогатный ключ обычно является кодом записи в другой связанной таблице. Вместо непосредственно суррогатного ключа отображаться будут поля из табличной группы полей <i>AutoIdentification</i> . Для пользователя они являются более понятными и дружелюбными

Инструмент Bound Field Designer в Visual Studio в зависимости от типа полей автоматически группирует их в дизайнера по соответствующим типам элементов *AxBoundField*.

AxContentPanel

Элемент управления *AxContentPanel* наследует от элемента ASP.NET *UpdatePanel*. Он выступает в роли контейнера для других элементов и поддерживает частичное обновление страницы для элементов управления, помещенных внутри него, за счет чего отпадает необходимость обновлять страницу целиком. Также для своих дочерних элементов управления он реализует механизм предоставления и использования информации о контексте записи.

AxPartContentArea

Используйте *AxPartContentArea* для определения области информационных панелей (FactBox) в рамках элемента управления. Этот элемент выступает в роли контейнера для *AxInfoPart*, *AxFormPart* и *CueGroupPartControl*.

AxInfoPart

Используйте элемент управления *AxInfoPart* для отображения элемента данных (Info Part). Этот элемент должен размещаться внутри элемента управления *AxPartContentArea*.

AxFormPart

Используйте элемент управления *AxFormPart* для отображения элемента формы (Form Part). Этот элемент должен размещаться внутри элемента управления *AxPartContentArea*.

CueGroupPartControl

Используйте элемент управления *CueGroupPartControl* для отображения группы элементов Cue. Этот элемент должен размещаться внутри элемента управления *AxPartContentArea*.

AxDatePicker

Используйте элемент управления *AxDatePicker* для отображения календаря, позволяющего пользователю выбрать дату.

AxReportViewer

Используйте элемент управления *AxReportViewer* для отображения SSRS-отчета.

Разработка для Корпоративного портала

Для разработки приложений Корпоративного портала совместно используются MorphX, Visual Studio, а также SharePoint.

- **MorphX** используется для разработки компонентов бизнес-логики и доступа к данным в вашем приложении. Кроме того, **MorphX** применяется для определения элементов навигации, хранения унифицированных метаданных и файлов, импорта и развертывания элементов управления, страниц и определений списков, а также генерации прокси-классов. Более детальную информацию о MorphX вы можете найти в главе 2.
- **Visual Studio** используется для разработки и отладки пользовательских веб-элементов управления. Расширение **Visual Studio для Корпоративного портала** предоставляет шаблоны проектов и элементов управления, позволяя ускорить процесс разработки. Visual Studio дает возможность с легкостью добавлять новые элементы управления в AOT, обеспечивает инструменты импорта элементов управления и таблиц стилей из AOT, а также возможность работы с прокси-классами. Инфраструктура Корпоративного портала предоставляет различные API для доступа к данным и метаданным.
- **SharePoint** используется для разработки веб-частей и списков. Также с его помощью вы можете редактировать шаблоны страниц, которые содержат общие элементы для всех страниц веб-сайта. В браузере вы можете воспользоваться инструментами **SharePoint по созданию и редактированию** страниц, чтобы разработать вашу страницу, использующую веб-части. Кроме того, вы можете воспользоваться инструментом **SharePoint Designer**, чтобы создавать или изменять как страницы, использующие веб-части, так и шаблоны страниц.

AOT управляет всеми метаданными для Корпоративного портала и хранит все элементы управления и страницы, которые вы разрабатываете в Visual Studio и SharePoint. В нем также хранятся другие вспомогательные файлы, определения и возможности, которые можно найти в узле *Web*.

В следующем разделе вы найдете пошаговое руководство по созданию страниц списков и страниц подробных сведений Корпоративного портала, а также разъяснения о том, как можно улучшить производительность с помощью AJAX. Информацию о пользовательском интерфейсе Корпоративного портала см. в главе 5.

Создание управляемой моделями страницы списка

В Microsoft Dynamics AX 2012 появился новый управляемый моделями способ создания страниц списков. В Microsoft Dynamics AX 2009 вам при-

ходилось создавать форму для отображения в Windows-клиенте Microsoft Dynamics AX и веб-страницу для отображения в Корпоративном портале. В случае управляемых моделями страниц списков вы создаете страницу списка один раз и затем можете использовать ее как в Windows-клиенте, так и в Корпоративном портале. Форма, отображаемая в клиенте Microsoft Dynamics AX, и веб-страница, отображаемая в Корпоративном портале, разделяют общий код и метаданные. Любые изменения в форме автоматически отражаются как в клиенте, так и в Корпоративном портале. Это дает целый ряд преимуществ, таких как снижение трудоемкости разработки, использование общего кода и упрощение поддержки.

На рис. 7-7 показан пример среды разработки для создания управляемой моделями страницы списка.

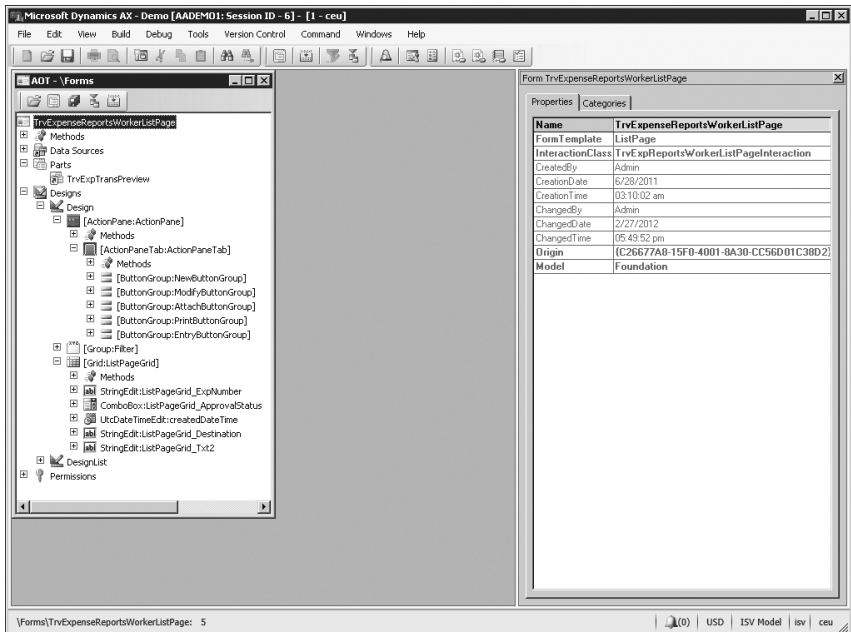


Рис. 7-7. Разработка управляемой моделями страницы списка

Чтобы создать управляемую моделями страницу списка, вы можете воспользоваться представленным ниже высокоуровневым пошаговым руководством.

1. Откройте рабочую область разработки.

2. Создайте новую форму в AOT и установите свойство *FormTemplate* в значение *ListPage*. Эта настройка автоматически добавляет элементы дизайна, такие как фильтр, элемент *grid* и панель операций.
3. Настройте запрос на форме для получения данных, которые вы хотите на ней отображать.
4. В свойстве *DataSource* элемента *grid* выберите нужное представление данных.
5. Добавьте поля, которые хотите отображать в элементе *grid*.
6. Создайте и добавьте панель операций, а также при необходимости элементы данных (**Info Parts**). **В идеальном случае вам следует создать один элемент данных для отображения в области просмотра (под элементом *grid*) и один или более элементов данных, элементов форм и групп Cue для отображения в области информационных панелей (справа от элемента *grid*).**

Область просмотра должна отображать подробную информацию о выбранной записи, а информационные панели – связанную информацию. Чтобы связать эти части со страницей списка, вам потребуется создать соответствующие пункты меню отображения.

7. Создайте пункт меню отображения, который указывает на форму. Щелкните правой кнопкой мыши по этому пункту меню и выберите нужное значение из контекстного меню Развернуть на корпоративному портале.
8. Когда появится запрос, выберите модуль, в котором вы хотите развернуть страницу. В результате для страницы списка Корпоративного портала будет автоматически создана страница **SharePoint**, использующая веб-части. Также будет создан веб-пункт меню с URL, и соответствующее определение страницы будет импортировано в AOT.
9. В свойстве *HyperLinkMenuItem* первого поля в элементе *grid* укажите пункт меню отображения, который соответствует странице подробных сведений, и затем обновите AOD. В результате в первой колонке будут созданы гиперссылки, с помощью которых выбранную запись можно открыть на связанной странице подробных сведений.

Определение класса взаимодействия для страницы списка

Для достижения более полного контроля над поведением вашей управляемой моделями страницы списка в свойстве формы *InteractionClass* вы

можете указать название разработанного вами класса взаимодействия. Название класса должно заканчиваться на *ListPageInteraction*, а сам класс может быть наследником или более простого в использовании класса *SysListPageInteractionBase*, или более гибкого класса *ListPageInteraction*.

Класс *SysListPageInteractionBase* предоставляет методы, перегрузив которые, вы можете разместить необходимую прикладную логику. Ниже приведены некоторые из этих методов.

- ***initializing*** вызывается при инициализации страницы списка.
- ***selectionChanged*** вызывается, когда пользователь выбирает на странице списка другую запись.
- ***setButtonEnabled*** управляет доступностью кнопок, вызывается из метода *selectionChanged*.
- ***setButtonVisibility*** управляет видимостью кнопок. Этот метод вызывается один раз при открытии формы.
- ***setGridFieldVisibility*** управляет видимостью полей элемента *grid*. Этот метод вызывается один раз при открытии формы.

Более подробную информацию вы можете найти в разделе «*SysListPageInteractionBase Class*» по адресу: <http://msdn.microsoft.com/en-us/library/syslistpageinteractionbase.aspx>.

Создание страницы подробных сведений

Страница подробных сведений в Корпоративном портале отображает детальную информацию о той или иной записи.

Используйте представленное ниже высокоуровневое пошаговое руководство для создания страницы подробных сведений.

1. В Visual Studio создайте новый проект, используя шаблон проекта EP Web Application Project (в категории Microsoft Dynamics AX).
2. Добавьте в проект новый элемент, используя шаблон EP User Control with the Form (в категории Microsoft Dynamics AX). При этом элемент управления будет автоматически добавлен в АОТ.
3. Переключитесь в режим отображения дизайна, выберите элемент управления *AxDataSource* (рис. 7-8) и затем установите название *DataSet*.

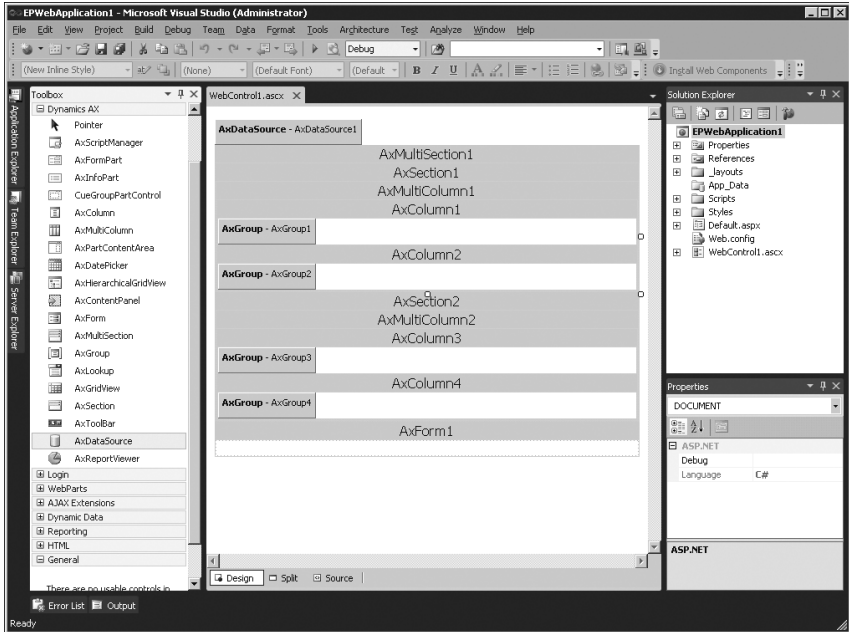


Рис. 7–8. Создание страницы подробных сведений в Visual Studio

4. Выберите элемент управления *AxForm*, затем убедитесь, что *DataSourceID* ссылается на *AxDataSource*.
5. Установите *DataMember* и *DataKeyNames* на форме соответствующим образом.
6. Если необходимо, измените режим работы формы по умолчанию на *Edit* или *Insert* (по умолчанию режим – *ReadOnly*).
7. Чтобы кнопки Сохранить и Закрыть были сгенерированы автоматически, выполните следующее.
 - ▶ В режиме *ReadOnly* установите *AutoGenerateCancelButton* в *true*.
 - ▶ В режиме *Edit* установите *AutoGenerateEditButton* в *true*.
 - ▶ В режиме *Insert* установите *AutoGenerateInsertButton* в *true*.
 - ▶ Выберите элемент управления *AxGroup* и убедитесь, что свойство *FormID* заполнено.
8. Щелкните по ссылке **Edit Fields** и добавьте необходимые поля в элемент управления *AxGroup*.

9. С помощью меню **Build** скомпилируйте веб-приложение Корпоративного портала и убедитесь, что операция завершилась без ошибок. Компиляция приложения приводит к автоматическому развертыванию элемента управления в каталог SharePoint.
10. В Microsoft Dynamics AX откройте рабочую область разработки, затем перейдите к \Web\Web Content\Managed.
11. Щелкните правой кнопкой мыши по узлу **AOT, связанному с созданным вами пользовательским веб-элементом управления**, и выберите из контекстного меню **Развернуть** на корпоративном портале.
12. Когда появится запрос, выберите модуль, в котором вы хотите развернуть страницу. В результате для страницы Корпоративного портала будет автоматически создана страница SharePoint, использующая веб-части, и ваш пользовательский веб-элемент управления будет помещен на страницу с помощью веб-части **Web User control**. Также будет создан веб-пункт меню с **URL, и соответствующее определение страницы** будет импортировано в AOT.
13. Выберите созданный для страницы веб-пункт меню и в его свойствах установите *WindowMode* в *Modal*. За счет этого страница будет открываться в модальном диалоговом окне.
14. Создайте новый пункт меню отображения и в его свойстве *WebMenuItemName* укажите веб-пункт меню, связанный со страницей подробных сведений.
15. Используйте этот пункт меню отображения для связи страницы подробных сведений с элементом *grid* страницы списка, как описано в разделе «Создание управляемой моделями страницы списка» ранее в этой главе.

Настройки модальных диалоговых окон

Корпоративный портал использует модальные диалоговые окна при реализации для страниц стандартных шаблонов взаимодействия. В Microsoft Dynamics AX 2012 в веб-пунктах меню для Корпоративного портала появились два новых свойства – *WindowMode* и *WindowSize*, с помощью которых вы можете реализовывать эти шаблоны взаимодействия без необходимости писать код.

Для *WindowMode* доступны такие четыре возможных значения.

- **Inline.** Целевой URL откроется в том же самом окне. При такой настройке текущая страница замещается целевой страницей, с которой связан пункт меню.

- **Modal.** Целевой URL откроется в модальном диалоговом окне поверх текущего окна. Текущая страница все еще будет доступна в фоне, однако, поскольку диалоговое окно будет модальным, пользователь сможет взаимодействовать только с этим модальным окном, но не со страницей, на фоне которой оно откроется.

Если пункт меню со свойством *WindowMode*, установленным в *Modal*, открывается из модального диалогового окна, то будет использовано то же окно. Страница, открытая в модальном диалоговом окне, будет замещена страницей, на которую ссылается пункт меню.

- **NewModal.** Работает аналогично настройке *Modal*, но не использует повторно существующее модальное диалоговое окно. Таким образом, если пункт меню со свойством *WindowMode*, установленным в *NewModal*, открывается из модального диалогового окна, то поверх исходного модального диалогового окна откроется модальное диалоговое окно второго уровня, как показано на рис. 7-9.
- **NewWindow.** Целевой URL откроется в новом окне.

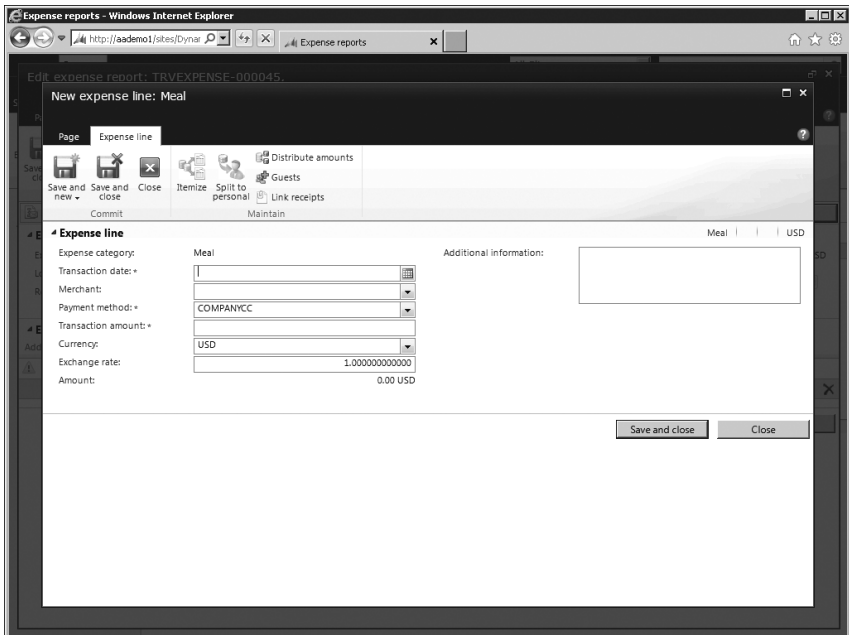


Рис. 7-9. Пример страницы Корпоративного портала с двумя уровнями модальных диалоговых окон

Для *WindowSize* доступны пять значений: *Maximum*, *Large*, *Medium*, *Small* и *Smallest*. Они соответствуют пяти предопределенным размерам модальных диалоговых окон.

AJAX

Технология ASP.NET AJAX позволяет разработчикам создавать веб-страницы ASP.NET, которые могут обновлять отображаемые данные без необходимости перезагрузки всей страницы. AJAX предоставляет клиентские и серверные компоненты, использующие объект *XMLHttpRequest* совместно с JavaScript и DHTML, которые позволяют асинхронно обновлять отдельные части веб-страницы без повторной загрузки всей страницы. С помощью AJAX разработчики могут создавать веб-страницы Корпоративного портала аналогично созданию любой обычной веб-страницы ASP.NET, с возможностью декларативного указания компонентов, которые необходимо отображать асинхронно.

С помощью серверного элемента управления *UpdatePanel* можно выбрать те секции веб-страницы, которые должны отображаться без необходимости обратной передачи всей веб-страницы. Веб-часть *User Control* уже содержит экземпляр элемента *UpdatePanel*, а библиотека скриптов включена в шаблон страницы, поэтому любой элемент управления может использовать AJAX без единой строки кода и явных указаний в разметке страницы.

К примеру, если добавить на страницу поле ввода текста и кнопку и написать без использования AJAX серверный код обработчика события нажатия на эту кнопку, то при нажатии на нее будет полностью обновляться вся веб-страница. А если те же элементы поместить в веб-часть *User Control*, как в примере ниже, кнопка будет автоматически использовать AJAX и обновит содержимое поля ввода без обновления всей веб-страницы.

```
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
<asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Button" />
```

В коде обработчика нажатия кнопки установите в поле ввода значение текущего времени спустя пять секунд после нажатия.

```
protected void Button1_Click(object sender, EventArgs e)
{
    System.Threading.Thread.Sleep(5000);
    TextBox1.Text = System.DateTime.Now.ToShortTimeString();
}
```

Если же вы хотите изменить поведение AJAX по умолчанию и принудительно выполнить обратную передачу всей страницы, то можете воспользоваться элементом *PostBackTrigger*, как показано ниже.

```
<%@ Register assembly="System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" Namespace="System.Web.UI" TagPrefix="asp" %>
```

```
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
  <ContentTemplate>
    <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    <asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Button" />
  </ContentTemplate>
  <Triggers>
    <asp:PostBackTrigger ControlID="Button1" />
  </Triggers>
</asp:UpdatePanel>
```

Кэширование и освобождение сессии

Все веб-части на одной веб-странице используют одну общую сессию (связан с Microsoft Dynamics AX). После завершения загрузки страницы сессия завершается. В целях оптимизации производительности вы можете изменить время до завершения сессии; в файле настроек **Web.config** вы можете указать таймаут, а также количество закэшированных одновременно активных сессий.

К примеру, чтобы установить максимальное количество закэшированных активных сессий в 300, а таймаут завершения – в 45 секунд, добавьте в файл **Web.config** секцию *<Microsoft.Dynamics>* после элемента *</system.web>*, как показано ниже. Помните, что увеличение любого из этих параметров ведет к увеличению потребления оперативной памяти.

```
<Microsoft.Dynamics>
  <Session MaxSessions="300" Timeout="45" />
</Microsoft.Dynamics>
```

Многие из методов, которые используются в Корпоративном портале для написания кода на пользовательских веб-элементах управления, требуют доступа к объекту *Session*. Объект *Session* также нужно передавать в методы при использовании прокси-классов. Объект сессии доступен через веб-часть, в которой помещены веб-элементы управления, как показано ниже:


```
AxBaseWebPart webpart = AxBaseWebPart.GetWebpart(this);  
return webpart == null ? null : webpart.Session;
```

По умолчанию Корпоративный портал использует состояние сессии (session state) ASP.NET. Однако в сценариях, где используются фермы серверов, вы можете настроить и использовать совместно с Корпоративным порталом систему распределенного кэширования Windows Server AppFabric, чтобы еще больше повысить производительность. После установки и настройки Windows Server AppFabric вы можете указать в файле Web.config, какое название и регион должен использовать Корпоративный портал.

К примеру, чтобы задать название кэша *MyCache* и регион *MyRegion*, добавьте в файл Web.config Корпоративного портала секцию *<Microsoft.Dynamics>* после элемента *</system.web>*, как показано ниже.

```
<Microsoft.Dynamics>  
  <AppFabricCaching CacheName="MyCache" Region="MyRegion" />  
</Microsoft.Dynamics>
```

Контекст

Контекст – это структура данных, которая позволяет обмениваться данными активной среды с другими участками веб-приложения при выполнении действий пользователем или других событиях. С помощью контекста можно узнать о событиях, происходящих с одним элементом управления, в другом элементе управления или веб-части и отреагировать на эти события или, к примеру, передать полученную информацию новой веб-странице. Обычно контекст формируется из информации о текущей выбранной записи, с которой работает пользователь. Например, при выборе пользователем новой строки источника данных, другим управляющим элементам может потребоваться получить информацию о выбранной пользователем строке, чтобы отреагировать соответствующим образом.

Концепция контекста инкапсулирована в абстрактном классе *AxContext*. Классы *AxTableContext* и *AxViewContext* наследуют и реализуют *AxContext*. Класс *AxTableContext* предназначен для табличного контекста, а класс *AxViewContext* – для контекста представления набора данных. Представление может содержать более одной таблицы, поэтому класс *AxViewContext* содержит объект *AxTableContext* для каждой таблицы из представления в коллекции *TableContextList*. Свойство класса *RootTableContext* возвращает объект *TableContext* для главной таблицы представления набора данных. Объект *AxViewDataKey* уникально иден-

тифицирует объект *AxViewContext* и содержит коллекцию *TableDataKeys* для уникальной идентификации табличного контекста *AxTableContext* с помощью класса *AxTableDataKey*. Каждый раз при изменении контекста генерируется соответствующее событие. При изменении контекста внутри элемента *Web User Control* генерируется событие *CurrentContextChanged*. Если изменяется контекст других веб-частей, связанных с текущей, то генерируется событие *ExternalContextChanged*.

В обработчиках этих событий на элементе *AxBaseWebPart* можно писать код, используя свойство *CurrentContextProviderView* или *ExternalContextProviderView*, а также свойство *ExternalRecord* для получения записи данных, связанной с данным контекстом. Эти события также могут создаваться программно из бизнес-логики приложения путем вызова *FireCurrentContextChanged* или *FireExternalContextChanged*, что позволит всем остальным элементам управления отреагировать на изменение данных, выполненное из кода. Ниже приводится простой пример кода для создания события *CurrentContextChanged*.

```
void CurrentContextProviderView_ListChanged(object sender,
    System.ComponentModel.ListChangedEventArgs e)
{
    /* The current row (which is the current context) has changed update the consumer webparts.
    Fire the current context change event to refresh (re-execute the query) the consumer
    web parts
    */
    AxBaseWebPart webpart = this.WebPart;
    webpart.FireCurrentContextChanged();
}
```

В следующем примере запись получается из связанной веб-части. Сначала в веб-части, являющейся потребителем, необходимо подписаться на событие *ExternalContextChanged*, как показано ниже.

```
protected void Page_Load(object sender, EventArgs e)
{
    //Add Event handler for the ExternalContextChange event.
    //Whenever selecting the grid of the provider web part changes, this event gets fired.
    (AxBaseWebPart.GetWebpart(this)).ExternalContextChanged +=
    new EventHandler<Microsoft.Dynamics.Framework.Portal.UI.AxExternalContextChangedEvent
    Args>
    (AxContextConsumer_ExternalContextChanged);
}
```

Затем нужно получить запись, переданную через внешний контекст.

```
void AxContextConsumer_ExternalContextChanged(object sender,
    Microsoft.Dynamics.Framework.Portal.UI.AxExternalContextChangedEventArgs e)
{
    //Get the AxTableContext from the ExternalContext passed through web part connection
    // and construct the record object and get to the value of the fields
    IAxaptaRecordAdapter currentRecord = (AxBaseWebPart.GetWebpart(this)).ExternalRecord;

    if (currentRecord != null)
    {
        lblCustomer.Text = (string)currentRecord.GetField("Name");
    }
}
```

Данные

Элементы управления ASP.NET получают доступ и манипулируют данными посредством привязки к данным через *AxDataSource*. Вы можете напрямую получить доступ к данным при помощи API. Пространство имен *Microsoft.Dynamics.AX.Framework.Portal.Data* содержит несколько классов, которые вместе могут использоваться для получения данных.

Например, для получения текущей записи из *DataSetView* можно использовать следующий код.

```
private DataSetViewRow CurrentRow
{
    get
    {
        try
        {
            DataSetView dsv = this.ContactInfoDS.GetDataSet().DataSetViews[this.
            ContactInfoGrid.DataMember];

            return (dsv == null) ? null : dsv.GetCurrent();
        }
        // CurrentRow on the dataset throws exception in empty data scenarios
        catch (System.Exception)
        {
            return null;
        }
    }
}
```

Чтобы для пункта меню установить контекст с текущей записью, используйте такой код.

```

DataRow currentContact =
    this.dsEPVendTableInfo.GetDataSourceView(gridContacts.DataMember).DataSetView.
GetCurrent();

using (IAxaptaRecordAdapter contactPersonRecord = currentContact.GetRecord())
{
    ((AxUrlMenuItem)e.MenuItem).MenuItemContext =
        AxTableContext.Create(AxTableDataKey.Create(
            this.BaseWebpart.Session, contactPersonRecord, null));
}

```

Метаданные

Инфраструктура Корпоративного портала предоставляет богатый набор программных интерфейсов для доступа к метаданным **АОТ из управляемого кода**. Пространство имен *Microsoft.Dynamics.AX.Framework.Services.Client* содержит несколько классов, которые вместе могут использоваться для получения метаданных АОТ. Элементы управления Корпоративного портала используют метаданные для получения информации о форматировании, проверке правильности ввода, правах доступа и прочее, а затем автоматически применяют ее к пользовательскому интерфейсу. Вы также можете использовать эти программные интерфейсы для получения метаданных и использования их в презентационной логике.

Класс *MetadataCache* является основной точкой входа при доступе к метаданным и предоставляет для этого статические методы. Скажем, чтобы получить метаданные о перечислении, используйте класс *EnumMetadata* и метод *MetadataCache.GetEnumMetadata*, как в примере ниже.

```

/// <summary>
/// Loads the drop-down list with the enum values.
/// </summary>
private void LoadDropDownList()
{
    EnumMetadata salesUpdateEnum = MetadataCache.GetEnumMetadata(
        this.AxSession, EnumMetadata.EnumNum(this.AxSession, "SalesUpdate"));

    foreach (EnumEntryMetadata entry in salesUpdateEnum.EnumEntries)
    {
        ddlSelectionUpdate.Items.Add(new ListItem(entry.GetLabel(this.AxSession),
            entry.Value.ToString()));
    }
}

```

```
    }
}
```

Для получения метки табличного поля используйте следующий код.

```
TableMetadata tableSalesQuotationBasketLine =
    MetadataCache.GetTableMetadata(this.AxSession, "CustTable");

TableFieldMetadata fieldItemMetadata = tableSalesQuotationBasketLine.
FindDataField("AccountNum");

String s = fieldItemMetadata.GetLabel(this.AxSession);
```

На рис. 7-10 показана часть иерархии классов, используемых для доступа к метаданным. Чтобы упростить рисунок, в него включены не все классы.

Прокси-классы

Если вам нужно получить доступ к классам X++, вызывать табличные методы или использовать перечисления в вашем элементе управления, инфраструктура Корпоративного портала предоставит простой способ создания управляемых оберток для этих объектов X++. **Прокси-файл** является оберткой для вызовов .NET Business Connector и предоставляет простой, строго типизированный интерфейс для приложений на C#.

Для использования с Корпоративным порталом доступно несколько готовых прокси-классов. Они определены в проектах *EPApplicationProxies* и *EPApplicationProxies1* в AOT, в узле \Visual Studio Projects\C Sharp Projects. Чтобы использовать эти проекты прокси-классов, откройте проект вашего веб-приложения в Visual Studio и добавьте ссылку на эти проекты через меню **Project > Add EP Proxy Project**. Затем в веб-элементе приложения добавьте оператор *using* для доступа к пространству имен прокси-классов, как показано ниже.

```
using Microsoft.Dynamics.Portal.Application.Proxy;
```

Если нужно создать новые прокси-классы, вы можете создать собственный проект библиотеки классов Visual C# в Visual Studio следующим образом.

1. Для проекта установите по умолчанию пространство имен *Microsoft.Dynamics.Portal.Application.Proxy*.

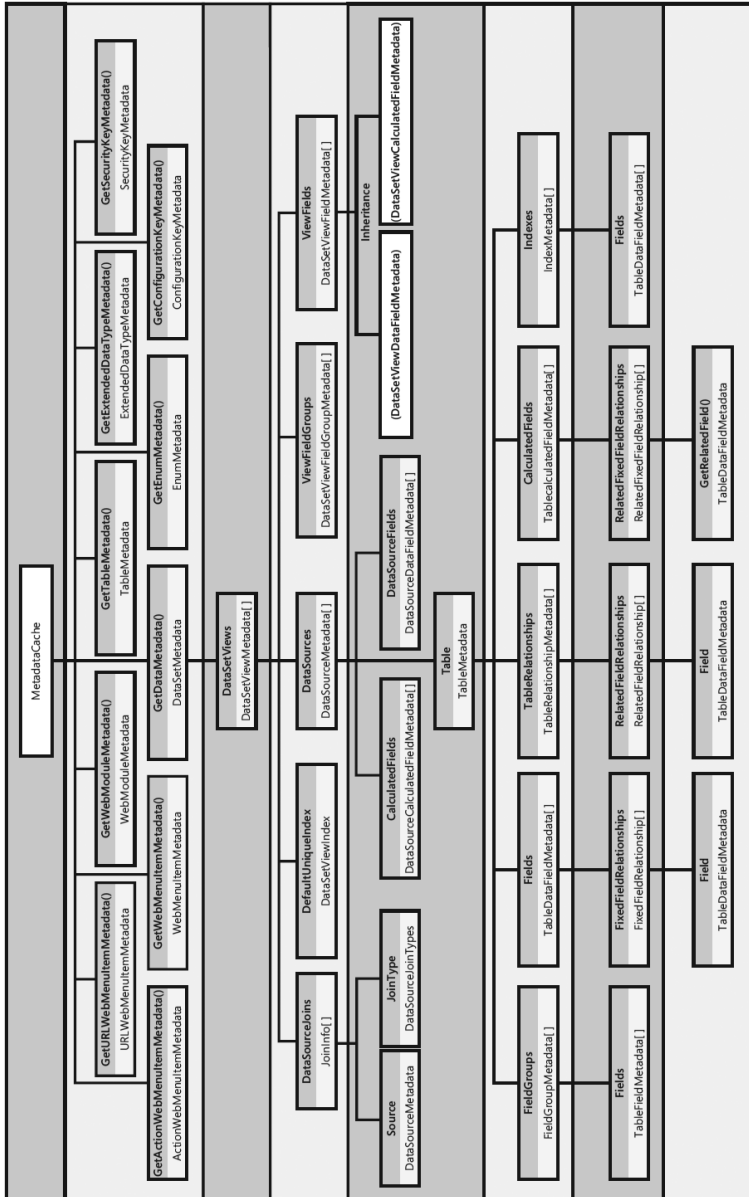


Рис. 7-10. Иерархия объектов метаданных

- В меню File выберите соответствующий пункт для добавления проекта в AOT.
- В свойствах проекта установите *Deploy to EP* в значение *Proxies*.

Затем можете добавить в проект объекты из Application Explorer, и инфраструктура Корпоративного портала автоматически сгенерирует прокси-классы и произведет развертывание соответствующих сборок в каталог *App_Code* веб-сайта IIS. После добавления ссылки на проект с прокси-классами вы можете вызывать методы объектов X++, как если бы они были написаны на C# (рис. 7-11).

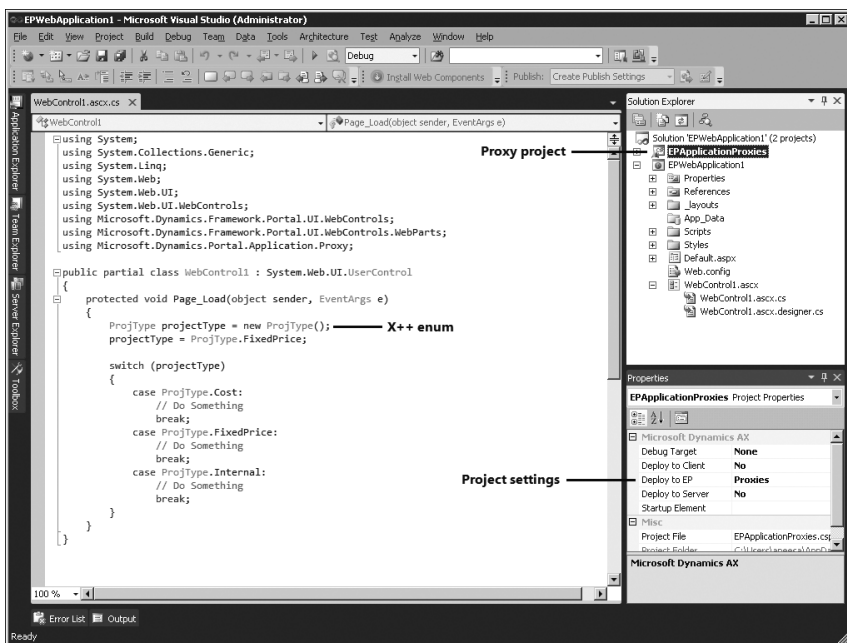


Рис. 7-11. Работа с прокси-классами в Visual Studio

Состояние *ViewState*

Веб не хранит состояние, это значит, что каждый запрос веб-страницы рассматривается как новый запрос, без какой-либо разделяемой информации. При загрузке каждая страница ASP.NET проходит через обычный жизненный цикл, от инициализации к загрузке страницы и далее. Когда

взаимодействие пользователя со страницей требует от сервера обработать управляющие события, ASP.NET с помощью `http`-метода `POST` отсылает значения из полей формы той же самой странице, чтобы обработать соответствующее событие на сервере. Каждый раз, когда страница запрашивается с сервера, создается новый экземпляр класса веб-страницы. Во время обратной передачи данных ASP.NET использует функционал *ViewState* для сохранения состояния страницы и элементов управления, чтобы изменения страницы, произошедшие во время передачи данных от клиента серверу и обратно, не потерялись. Инфраструктура Корпоративного портала и управляющие элементы ASP.NET **Корпоративного портала** автоматически сохраняют свое состояние во *ViewState*. Страница ASP.NET считывает *ViewState* и воссоздает свое состояние и состояние элементов управления во время обычного жизненного цикла страницы. За счет этого при использовании элементов управления Корпоративного портала вам не нужно писать какой-либо код для управления состоянием. Однако если вы хотите сохранять значения ваших переменных, то можете написать код для добавления или удаления элементов из класса *StateBag* ASP.NET, как показано ниже.

```
public int Counter
{
    get
    {
        Object counterObject = ViewState["Counter"];

        if (counterObject == null)
        {
            return 0;
        }

        return (int)counterObject;
    }

    set
    {
        ViewState["Counter"] = value;
    }
}
```

Если вы хотите сохранять состояние набора данных (dataset) X++, то можете использовать для этого шаблон проектирования *pack-unpack*. Бо-

лее подробную информацию можно найти в разделе «Pack-Unpack Design Pattern» по адресу: <http://msdn.microsoft.com/en-us/library/aa879675.aspx>.

Инфраструктура Корпоративного портала использует свойство ASP.NET *ViewState* для хранения состояния большинства элементов управления.

Метки

В приложении Microsoft Dynamics AX для вывода сообщений пользователям, текста на элементах интерфейса, текста справки, строки состояния и заголовка используются локализуемые файлы текстовых ресурсов, называемые «файлами меток». Метки могут использоваться и для указания текста, отображаемого на элементах веб-интерфейса в пользовательских веб-элементах управления, и в свойствах узлов AOT *Web*. Метки можно добавлять путем установки свойства *Label* в AOT или из кода X++. При использовании элементов управления, привязанных к данным, таких как *AxGridView* или *AxForm*, привязанные поля автоматически подхватывают метки соответствующих полей из AOT и отображают их на языке пользователя во время выполнения.

Для отображения метки на элементе управления, который не привязан к данным, используется выражение *AxLabel*. *AxLabel* – это стандартное ASP.NET-выражение, которое считывает значения меток из определения в AOT и отображает их на веб-странице на языке пользователя. Чтобы добавить новое выражение *AxLabel*, используется редактор выражений, доступный из режима редактирования выбранного элемента управления по нажатию кнопки из свойства *Expressions*. Как альтернатива, можно написать код выражения непосредственно в разметке страницы.

```
<asp:Button runat="server" ID="ButtonChange" Text="<%"$ AxLabel:@SYS70959 %">  
OnClick="ButtonChange_Click" />
```

Значение метки можно также получить из кода с помощью класса *Labels*, как в этом примере.

```
string s = Microsoft.Dynamics.Framework.Portal.UI.Labels.GetLabel("@SYS111587");
```

Корпоративный портал для улучшения производительности кэширует тексты меток для всех поддерживаемых языков. Поэтому при добавлении или изменении метки в AOT на сайте Корпоративного портала необходимо очистить кэш с помощью команды Обновить AOD из меню администрирования на домашней странице Корпоративного портала.

Форматирование

Приложение Microsoft Dynamics AX – это действительно глобальный продукт, который поддерживает множество языков и используется во многих странах. Отображение данных в правильном формате для каждой из локализаций является важным требованием для любого глобального программного продукта. Инфраструктура Корпоративного портала определяет из метаданных приложения текущее местоположение пользователя и системные настройки, и в интерфейсе пользователя автоматически отображает данные в правильном формате и на правильном языке.

Если вы используете элементы управления ASP.NET, не привязанные к данным, и хотите, чтобы они отображали данные, отформатированные аналогично элементам управления Корпоративного портала, то можете использовать класс *AxValueFormatter* из инфраструктуры Корпоративного портала. Данный класс реализует интерфейсы *ICustomFormatter* и *IFormatProvider* и определяет метод, который поддерживает пользовательскую настройку форматирования значения объекта, а также обеспечивает механизм для получения ссылки на объект для управления его форматированием. Для различных типов данных существуют конкретные классы **ValueFormatter*, наследуемые от класса *AxValueFormatter*: *AxStringValueFormatter*, *AxDateValueFormatter*, *AxDateTimeValueFormatter*, *AxTimeValueFormatter*, *AxRealValueFormatter*, *AxNumberValueFormatter*, *AxGuidValueFormatter* и *AxEnumValueFormatter*.

Для создания объектов типа *AxValueFormatter* используется класс фабрики *AxValueFormatterFactory*. С помощью этой фабрики можно создать любой из вышеприведенных классов форматирования, а также создать класс форматирования для того или иного расширенного типа данных Microsoft Dynamics AX. Из метаданных AOT считывается базовый тип данных выбранного расширенного типа, а информация о региональных настройках приходит вместе с контекстом. Различные культурные и языковые правила, такие как числовые форматы, символы валюты, порядок сортировки данных, собраны в набор стандартных региональных настроек. Инфраструктура Корпоративного портала определяет выбранные региональные настройки в зависимости от языка, выбранного в настройках пользователя в Microsoft Dynamics AX, и помещает эту информацию в текущий контекст. Объекты форматирования имеют метод *Parse*, который позволяет конвертировать строковое значение в соответствующий классу форматирования тип данных. Например код,

приведенный ниже, форматирует данные на основе указанного расширенного типа.

```
private string ToEDTFormattedString(object data, string edtDataType)
{
    ExtendedDataTypeMetadata edtType = MetadataCache.GetExtendedDataTypeMetadata(
        this.AxSession, ExtendedDataTypeMetadata.TypeNum(this.AxSession, edtDataType));

    IAxContext context = AxContextHelper.FindIAxContext(this);

    AxValueFormatter valueFormatter = AxValueFormatterFactory.CreateFormatter(
        this.AxSession, edtType, context.CultureInfo);

    return valueFormatter.FormatValue(data);
}
```

Проверка правильности ввода

Проверяющие элементы управления ASP.NET используются для проверки правильности ввода значений на сервере, а также опционально на клиенте (браузере). Инфраструктура Корпоративного портала содержит набор проверяющих элементов специально для Microsoft Dynamics AX: *AxBaseValidator* наследуется от класса *System.Web.UI.WebControls.BaseValidator*, а *AxValueFormatValidator* наследуется от *AxBaseValidator*. Оба элемента основаны на метаданных и используются внутри полей, привязанных к данным, но могут применяться и в других сценариях, когда нет привязки к данным.

Проверка значений выполняется автоматически с помощью проверяющих элементов ASP.NET при обратной передаче страницы. К примеру, нажатие на управляющий элемент кнопки ASP.NET *Button* приводит к вызову проверки ввода как на клиенте, так и на сервере. Проверяется ввод всеми проверяющими элементами, зарегистрированными на веб-странице. Если в результате проверки какой-либо из проверяющих элементов указывает на некорректность ввода, то вся веб-страница тоже считается неверной и метод *Page.IsValid* возвращает *false*.

Важность метода *Page.IsValid* лучше всего демонстрируется на примере. Допустим, на веб-страницу добавляется кнопка, которая в методе обработки нажатия *OnClick* выполняет определенную бизнес-логику перед обратной передачей страницы. Как уже упоминалось ранее, при нажатии кнопки проверка вызывается автоматически и выполняется до создания события *OnClick*. Если не будет выполнена проверка правильности ввода

до отработки события *OnClick*, то будут выполнены прикладной код и обратная передача страницы, несмотря на то, что существуют ошибки ввода, которые пользователь должен исправить.

Элементы управления Корпоративного портала, такие как *AxForm* и *AxGridView*, не выполняют запрашиваемых действий, если проверка данных не пройдет успешно. Проверяющие элементы управления Microsoft Dynamics AX автоматически добавляют все сообщения об ошибках ввода в веб-часть Infolog.

Когда вместо элементов управления Корпоративного портала вы напрямую используете элементы управления ASP.NET, рекомендуется проверять состояние страницы через метод *Pages.IsValid* перед выполнением любых действий, таких как переход с текущей веб-страницы на другую. Это следует делать для того, чтобы в случае возникновения ошибок фокус остался на текущей веб-странице, а в Infolog вывелись сообщения об ошибках, которые помогут пользователям принять соответствующие меры.

Обработка ошибок

В Корпоративном портале, .NET Business Connector (включая прокси-классы), метаданные и слой доступа к данным приложения генерируют исключения в случае возникновения ошибочных ситуаций. Элементы управления Корпоративного портала автоматически обрабатывают эти исключения, выполняя соответствующие действия и вывода сообщения об ошибках в Infolog.

Исключения в Корпоративном портале делятся на три категории. Эти категории исключений определены в перечислимом типе *AxExceptionCategory*.

- **NonFatal.** Код обработчика данного типа исключения должен выполнить соответствующие действия и разрешить дальнейшее выполнение запроса.
- **AxFatal.** Означает, что возникла неисправимая исключительная ситуация в Корпоративном портале. В этом случае контент Корпоративного портала отображаться не будет. Весь остальной контент, не связанный с Корпоративным порталом, должен отображаться, как ожидалось.
- **SystemFatal.** Означает, что возникла серьезная ошибка, такая как нехватка памяти, и запрос отображения страницы должен быть аварийно прекращен. Ошибки такого типа часто приводят к ошибке HTTP с кодом 500.

Ваш код должен обрабатывать любые возможные исключения, если он выполняет что-либо из следующего списка.

- Напрямую вызывает методы слоя доступа к данным из Корпоративного портала.
- Напрямую вызывает методы доступа к метаданным.
- Использует прокси-классы для вызова методов X++.

Следующий код показывает, как использовать класс *AxControlExceptionHandler* в блоке *try-catch* для обработки исключений.

```
try
{
    // Code that may encounter exceptions goes here.
}
catch (System.Exception ex)
{
    AxEExceptionCategory exceptionCategory;

    // Determine whether the exception can be handled.
    if (AxControlExceptionHandler.TryHandleException(this, ex, out exceptionCategory) == false)
    {
        // The exception was fatal and cannot be handled. Rethrow it.
        throw;
    }
    if (exceptionCategory == AxEExceptionCategory.NonFatal)
    {
        // Application code to properly respond to the exception goes here.
    }
}
```

AxControlExceptionHandler пытается обработать исключения Microsoft Dynamics AX на основе трех категорий исключений, которые обсуждались выше. Он возвращает *true* в случае, если исключение относится к категории *NonFatal*.

Контроль доступа

В Корпоративном портале подсистема контроля доступа Microsoft Dynamics AX функционирует, опираясь на используемые продукты и технологии, такие как SharePoint и IIS. Для обеспечения безопасности и контроля доступа к сайтам Корпоративного портала, доступным из Интернета, также важны настройки firewall и обеспечение защищенной передачи данных.

Существуют две конфигурации в определении сайта Корпоративного портала. Первая, **Microsoft Dynamics Public**, позволяет клиентам или потенциальным клиентам, осуществляющим доступ через Интернет, просматривать каталоги продукции, запрашивать заведение клиентских учетных записей и т.д. Вторая конфигурация, **Microsoft Dynamics Enterprise Portal**, представляет собой полнофункциональный портал для сценариев самообслуживания с участием пользователей, осуществляющих доступ из корпоративной или внешней сети и являющихся сотрудниками компании, поставщиками или клиентами.

В конфигурации **Microsoft Dynamics Public** разрешена анонимная аутентификация как в IIS, так и в SharePoint, так что получить доступ к сайту может кто угодно. Для подключения к **Microsoft Dynamics AX** в этой конфигурации используется встроенная учетная запись пользователя **Microsoft Dynamics AX**, именуемая **Guest**. Учетная запись **Guest** входит в группу пользователей **Enterprise Portal Guest**, у которой есть ограниченный доступ к компонентам **Microsoft Dynamics AX**, необходимым для функционирования общедоступного сайта.

Конфигурация **Microsoft Dynamics Enterprise Portal** использует либо интегрированную **Windows-аутентификацию**, либо базовую аутентификацию, а доступ осуществляется с использованием **SSL (Secure Sockets Layer)**, разрешенным как в IIS, так и в SharePoint. Доступ к такому защищенному сайту имеют лишь пользователи с учетными записями в **Active Directory**, которые также настроены как пользователи **Microsoft Dynamics AX** и имеют разрешение на доступ к сайту, настроенное системным администратором **Microsoft Dynamics AX**. Для настройки пользователей в качестве сотрудников, поставщиков, деловых партнеров или клиентов используется форма **Администрирование системы > Настройка > Пользователи > Пользовательские связи**, доступная в клиенте **Microsoft Dynamics AX**. После настройки связей вы можете предоставить пользователям доступ к тому или иному сайту Корпоративного портала через группу сайтов, в которую входит данный сайт.

Для подключения к **AOS** в обеих конфигурациях сайтов Корпоративного портала используется учетная запись **.NET Business Connector проху**. Чтобы сайты обоих типов могли функционировать, пул приложений **SharePoint** должен быть настроен на использование доменной учетной записи и эта учетная запись должна быть настроена в **Microsoft Dynamics AX** в качестве **.NET Business Connector проху**. После установки соединения для задействования подсистемы безопасности **Microsoft Dynamics**

AX Корпоративный портал использует либо *LogonAsGuest*, либо *LogonAs*, в зависимости от типа сайта Корпоративного портала, к которому пользователь осуществляет доступ. Microsoft Dynamics AX предоставляет разнообразные способы и средства ограничения доступа пользователя, такие как размещение ограничений на отдельных таблицах и полях и управление доступностью функционала приложения через конфигурационные ключи и конфигурационные веб-ключи, как показано на рис. 7-12. На уровне пользователя контроль доступа также может осуществляться за счет использования ролей, обязанностей и привилегий.

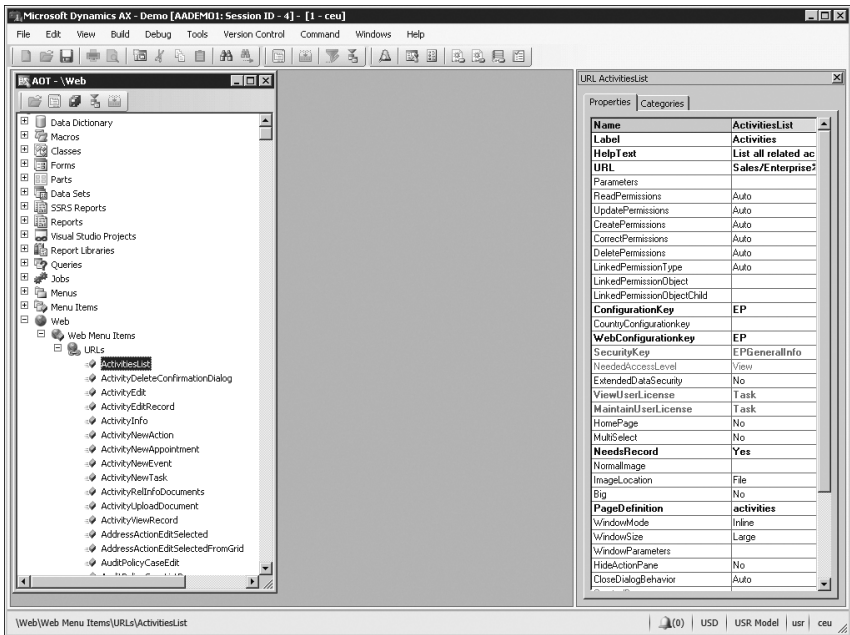


Рис. 7-12. Назначение веб-пункту меню конфигурационного ключа и конфигурационного веб-ключа

Контроль доступа в Корпоративном портале основан на ролях. Это означает, что вы можете легко сгруппировать задачи, связанные с той или иной бизнес-функцией, в роль, такую как менеджер по продажам или консультант, и назначить эту роль пользователям, чтобы предоставить им необходимые права доступа к объектам Microsoft Dynamics AX для выполнения соответствующих задач в Корпоративном портале. Чтобы дать пользователям доступ к большому объему функциональности, вы можете

назначить им несколько ролей. Более подробную информацию о ролях вы можете найти в главе 11.

Контроль доступа к веб-элементам

Для контроля доступа к веб-элементам управления в веб-частях Share-Point вы можете использовать привилегии, например создать новую или использовать одну из уже существующих. Чтобы контролировать, какие пользователи могут осуществлять доступ к управляемому веб-контенту (Web\Web Content\Managed), веб-пунктам меню, ссылающимся на URL (Web\Web Menu Items\URLs) или на какое-либо действие (Web\Web Menu Items\Actions), **вы можете добавить эти элементы приложения в привилегии в качестве точек входа.** Не забывайте, что контроль доступа необходим как к веб-пунктам меню, так и к управляемому веб-контенту. Если вы ограничите доступ только к веб-пункту меню (рис. 7-13), управляемый веб-контент, например веб-элемент управления, все еще останется доступен пользователю, и тот сможет добавить его на страницу, чтобы получить беспрепятственный доступ к нему.

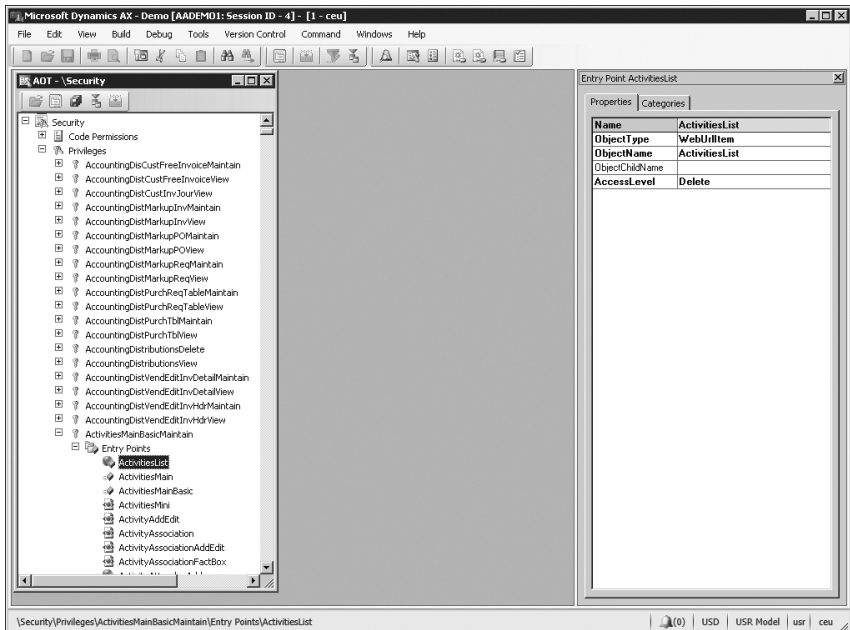


Рис. 7-13. Добавление в привилегию в качестве точки входа веб-пункта меню, ссылающегося на URL

Во время входа в систему роль пользователя определяет, что ему будет доступно. Если пользователь не имеет доступа к какому-либо веб-пункту меню, то данный пункт не появится в веб-меню пользователя во время выполнения. Если ссылка на недоступный веб-пункт меню находится на других веб-элементах управления, к которым доступ у пользователя имеется, то вместо гиперссылки пользователю будет отображаться только текст метки данного веб-пункта меню.

Если пользователь не имеет доступа к какому-либо веб-контенту, то данный контент не будет отображаться на веб-странице. Свойства веб-части также ограничивают перечень пунктов, доступных в выпадающем списке, в зависимости от прав доступа пользователя к соответствующим объектам. Более того, типы действий, которые может выполнять пользователь с данными объектами, зависят от уровня доступа, установленного для этих объектов в ролях, которые назначены пользователю.

Контекст записи и шифрование

Контекст записи – это интерфейс передачи информации через строку запроса странице веб-части для выборки записи из Microsoft Dynamics AX. Корпоративный портал использует контекст записи для поиска записи в базе данных Microsoft Dynamics AX и отображения ее в веб-форме для просмотра и редактирования. Некоторые из параметров строки запроса, передаваемых странице веб-части Корпоративного портала в качестве контекста записи, приведены ниже.

- **WTID.** Идентификатор таблицы (Table ID).
- **WREC.** Идентификатор записи (Rec ID).
- **WKEY.** Уникальный ключ записи (идентификатор поля и значение этого поля для той записи, которую необходимо найти).

Эти параметры передаются либо через строку запроса, либо через данные, отправляемые из веб-формы. Чтобы обеспечить безопасность Корпоративного портала, Microsoft Dynamics AX по умолчанию использует хэш-параметр. За счет этого URL, сгенерированный для одного пользователя, не может быть использован для какого-либо другого пользователя. В целях отладки и веб-разработки, администратор может отключить шифрование данных (использование хэш-параметра) в Корпоративном портале, сняв соответствующий флажок на вкладке Разное формы администрирования веб-узлов, которая доступна из меню Администрирование системы

> Настройка > Корпоративный портал > Веб-сайты. Если используется безопасность на уровне записей или другие меры ограничения доступа на уровне данных, то в случае отсутствия угроз безопасности отключение шифрования может повысить производительность системы. Однако настоятельно рекомендуется оставить шифрование включенным.

Интеграция с SharePoint

Корпоративный портал создан на платформе SharePoint и пользуется преимуществами некоторых полезных возможностей и функциональностей, предлагаемых SharePoint для реализации сценариев совместной работы и управления контентом.

Навигация по сайту

Сайты Корпоративного портала используют элементы навигации и объектную модель SharePoint для отображения навигационных элементов Microsoft Dynamics AX, хранящихся в АОТ. Чтобы в качестве навигационных элементов сайта SharePoint в верхней и левой частях страницы можно было отображать веб-меню из АОТ, установщик Корпоративного портала добавляет провайдеров навигации *DynamicsLeftNavProvider* и *DynamicsTopNavProvider*. Для редакций SharePoint Standard и Enterprise вместо *DynamicsTopNavProvider* добавляется *DynamicsMOSSTopNavProvider*. Эти провайдеры навигации замещают используемых по умолчанию провайдеров *TopNavigationDataSource* и *QuickLaunchDataSource*.

Веб-модули определяют сайты и дочерние сайты SharePoint в Корпоративном портале (например, Продажи, Персонал и т.д.). Они также используются при создании верхней навигационной панели. Если вы хотите скрыть ссылку на модуль из верхней навигационной панели, то можете установить свойство веб-модуля *ShowLink* в значение *No*.

Веб-меню представляют собой наборы веб-пунктов меню, ссылающихся на URL или на действия. Вы можете использовать эти элементы для определения структуры меню быстрого запуска для веб-модуля, указав в его свойстве *QuickLaunch* соответствующее веб-меню (рис. 7-14). Также для этих целей вы можете использовать веб-часть Quick launch. Ссылки автоматически отображаются или скрываются в зависимости от прав доступа пользователя. Веб-пункты меню помогают создавать гибкие и динамичные сайты. Вы также можете использовать веб-пункты меню вместе с веб-частью Left navigation, чтобы на странице или в пользова-

тельном веб-элементе управления предоставить пользователю ссылки для навигации.

С точки зрения реализации, инфраструктура использует класс *WebLink* для генерации гиперссылок. У этого класса есть все необходимые свойства и методы для передачи информации между сервером и браузером. И, что более важно, он содержит метод, который возвращает ссылку URL для отображения в браузере. Помимо этого, класс *WebLink* содержит несколько методов для передачи данных о записи таблицы.

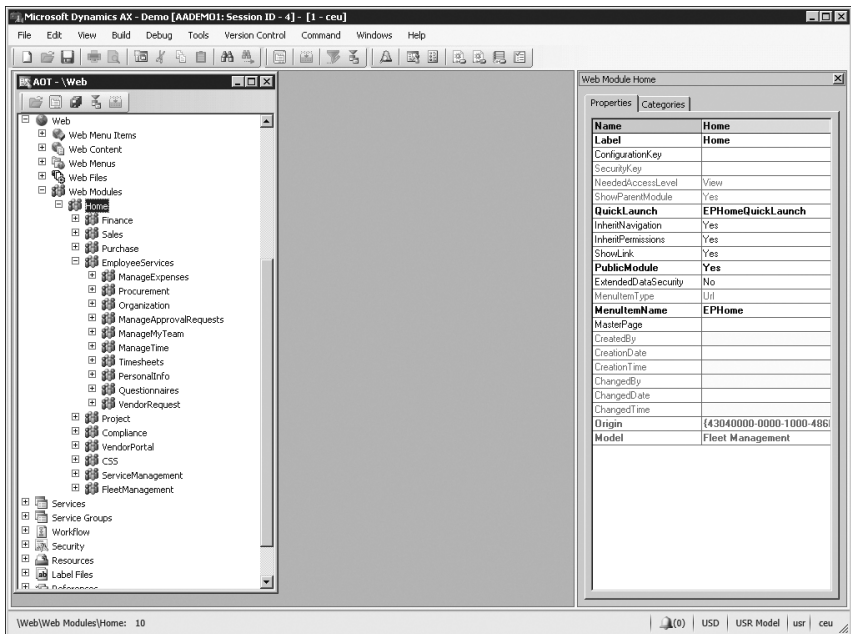


Рис. 7-14. Веб-модули определяют сайты, дочерние сайты и ссылки для быстрого перехода, отображаемые на странице

Определения сайтов, шаблоны страниц и веб-части

Сайты SharePoint можно модифицировать с помощью определений сайтов и настраиваемых шаблонов, созданных на основе существующих определений сайтов. Определение сайта содержит большое количество файлов, расположенных в файловой системе каждого из веб-серверов; эти файлы определяют структуру и схему сайта. Новые определения сайтов можно

создавать путем копирования файлов существующих определений и их изменения под требования новых сайтов. Настраиваемые шаблоны создаются с помощью имеющегося интерфейса пользователя для изменения существующих сайтов и сохранения их в качестве шаблонов.

Файлы определения сайта Корпоративного портала и настраиваемые шаблоны хранятся в AOT, в узле Web\Web Files\Static Files. При установке Корпоративного портала эти файлы развертываются из AOT в файловую систему веб-сервера и SharePoint.

Корпоративный портал содержит одно определение сайта по умолчанию, для которого есть две конфигурации: одна для аутентифицированных пользователей и другая для общего доступа пользователей через Интернет. Определение сайта развертывается в каталог <drive>:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\TEMPLATE\SiteTemplates\AXSITEDEF. Шаблоны страниц с веб-частями развертываются в каталог определения сайта, зависящий от языка: <drive>:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\TEMPLATE\<lcid>\AXSITEDEF.

Развертывание Корпоративного портала осуществляется в виде развертывания набора из четырех функциональных возможностей к SharePoint. Сайт SharePoint представляет собой модульную, размещаемую на сервере кастомизацию на основе набора файлов, содержащую элементы, которые могут быть установлены и активированы в среде SharePoint. Определения функциональных возможностей развертываются в <drive>:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\TEMPLATE\FEATURES. Для Корпоративного портала они включают следующие функциональности.

- **DynamicsAxEnterprisePortal.** Обеспечивает выполнение основных шагов установки Корпоративного портала, такие как развертывание шаблона страницы и других файлов и компонентов, настройка провайдеров навигации и регистрация Microsoft Dynamics AX. Эта функциональность предназначена для среды SharePoint Foundation.
- **DynamicsAxEnterprisePortalMOSS.** Включает шаги (с учетом среды, на которой производится установка) для развертывания в среде редакций SharePoint Standard и Enterprise.
- **DynamicsSearch.** Делает возможным поиск по данным Microsoft Dynamics AX и SharePoint с помощью окна поиска на сайтах Корпоративного портала.

- **DynamicsAxWebParts.** Служит для развертывания различных веб-частей Microsoft Dynamics AX.

Файлы, необходимые для функционирования Корпоративного портала, хранятся в AOT, в узле Web\Web Files\Static Files. В этом узле хранятся и другие файлы, связанные с инфраструктурой Корпоративного портала: файлы .aspx, которые используются для импорта/экспорта определений страниц и списков; файлы инфраструктуры управления документами; шаблон страницы; общие страницы ASP.NET; изображения; таблицы стилей и файлы конфигурации.

EPSetupParams – это файл XML, используемый при автосоздании сайта в ходе установки Корпоративного портала с целью определения значений по умолчанию для параметров сайта, таких как заголовок, описание и URL.

Шаблон страницы Корпоративного портала автоматически добавляет веб-части Page title, Quick launch и Infolog. Когда в Корпоративном портале создается веб-страница, эти веб-части уже доступны на ней, делая работу со всеми страницами Корпоративного портала единообразной и согласованной, а также позволяя ускорить разработку веб-приложения. На рис. 7-15 показаны некоторые из ключевых файлов, составляющих определение сайта, и их расположение на веб-сервере.

Веб-части Корпоративного портала хранятся в AOT, в узле Web\Web Files\Static Files. При необходимости компании-партнеры и клиенты могут добавлять в этот узел свои веб-части – Корпоративный портал автоматически обеспечит развертывание их файлов в GAC на веб-сервере и добавит записи о них как о безопасных элементах управления в файл Web.config.

Страницы с веб-частями отображают одну или несколько веб-частей, которые предоставляют простой способ создания функционально мощных веб-страниц, показывающих разнообразную информацию, от списка записей данных Microsoft Dynamics AX в текущем сайте до внешних данных, представленных специально разработанными веб-частями. Вы можете создавать страницы с веб-частями в SharePoint, используя Internet Explorer. Просто перетаскивайте веб-части на страницы и устанавливайте их свойства на основе предлагаемых списков значений. Редактировать страницы с веб-частями вы можете в SharePoint Designer либо в Internet Explorer. Internet Explorer может использоваться для изменения набора веб-частей на странице, их расположения, а также значений их свойств. SharePoint Designer может использоваться для вставки логотипов и других

графических элементов, изменения библиотек документов или списков, применения тем и стилей оформления, изменения шаблона страниц и т.д. Однако имейте в виду, что вы не сможете импортировать в АОТ страницы, измененные с помощью SharePoint Designer.

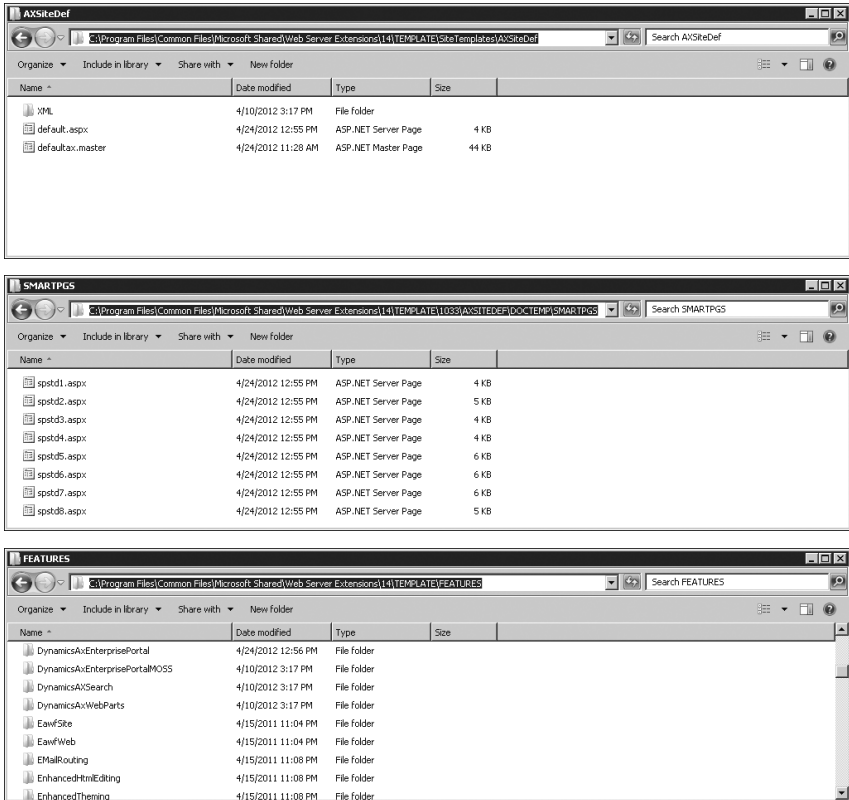


Рис. 7-15. Файлы определения сайта Корпоративного портала

Страницы с веб-частями, созданные на сайте Корпоративного портала в SharePoint, можно импортировать в АОТ в качестве определений страниц, используя пункт контекстного меню Импортировать страницу, доступный для веб-пунктов меню типа URL. Определения страниц хранятся в АОТ, в Web\Web Files\Page Definitions.

Определения страниц, импортированные в АОТ, автоматически создают страницы, когда на основе определения сайта Корпоративного пор-

тала создается новый сайт. Свойство *PublicPage* узла определения страницы указывает, должна ли страница создаваться на общедоступном сайте, для которого разрешен анонимный доступ. На сайте, создаваемом для доступа аутентифицированных пользователей, создаются все страницы. Если вы используете свойство *Title* определения страницы, то в нем должна содержаться метка, чтобы страница при доступе с другими региональными настройками отображала локализованный заголовок.

Импорт и развертывание страницы с веб-частями

Когда вы создаете новую страницу с веб-частями в Корпоративном портале, ее нужно импортировать в АОТ. Затем вы сможете провести развертывание этой страницы на другой сайт Корпоративного портала, кроме того, при создании нового сайта Корпоративного портала система также автоматически выполнит развертывание этой страницы. Чтобы импортировать страницу в АОТ, **создайте веб-пункт меню, ссылающийся на страницу**, щелкните по нему правой кнопкой мыши и из контекстного меню выберите **Импортировать страницу**. Импортированное определение страницы будет сохранено в `\Web\Web Files\Page Definitions`. После импорта страницы могут использовать для заголовков метки Microsoft Dynamics AX, за счет чего одно и то же определение страницы можно будет использовать для сайтов на различных языках.

Для создания или развертывания сайта Корпоративного портала или его отдельных элементов, таких как веб-модули, страницы, веб-элементы управления и т.д., вы можете использовать утилиту командной строки *AxUpdatePortal*. Она также поддерживает удаленное развертывание, так что вам не придется самостоятельно подключаться к каждому серверу Корпоративного портала, чтобы выполнить на нем развертывание.

Утилита *AxUpdatePortal* располагается либо в каталоге `C:\Program Files\Microsoft Dynamics AX\60\Setup` **на компьютере, где установлен Корпоративный портал**, либо в каталоге `C:\Program Files\Microsoft Dynamics AX\60\EnterprisePortalTools` **на компьютере, где установлен клиент Microsoft Dynamics AX**.

В табл. 7-3 перечислены параметры командной строки, поддерживаемые *AxUpdatePortal*.

Табл. 7-3. Параметры утилиты *AxUpdatePortal*

Параметр	Описание
<i>Listvirtualservers</i>	Выводит список всех виртуальных серверов (веб-приложений SharePoint и веб-сайтов IIS) на сервере
<i>-deploy</i>	Развертывает новый виртуальный сервер (веб-приложение SharePoint) на веб-сервере IIS, где уже установлен Корпоративный портал
<i>-createsite</i>	Создает веб-сайт Корпоративного портала на существующем виртуальном сервере Корпоративного портала
<i>-updateall</i>	Обновляет все веб-компоненты на веб-сайте Корпоративного портала (коллекции сайтов SharePoint)
<i>-proxies</i>	Обновляет все прокси-сборки на веб-сайте Корпоративного портала (коллекции сайтов SharePoint)
<i>-images</i>	Обновляет все изображения на веб-сайте Корпоративного портала (коллекции сайтов SharePoint)
<i>-updatewebcomponent</i>	Обновляет веб-компонент на веб-сайте Корпоративного портала (коллекции сайтов SharePoint)
<i>-websiteurl</i> <value>	URL веб-сайта Корпоративного портала (коллекции сайтов SharePoint)
<i>-treenodepath</i> <value>	Путь к узлу AOT развертываемого веб-компонента
<i>-updatewebsites</i>	Обновляет все веб-сайты Корпоративного портала в ходе повторного развертывания
<i>-iisreset</i>	Останавливает и перезапускает веб-сервер IIS после завершения операции по развертыванию

Для получения более подробной информации по этим параметрам запустите *AxUpdatePortal /?*.

Вот некоторые примеры использования *AxUpdatePortal* для выполнения различных действий над веб-сайтом, доступным по адресу: *http://ServerName/site/DynamicsAX*.

Создать и развернуть новый веб-сайт Корпоративного портала:

```
AxUpdatePortal -deploy -createsite -websiteurl "http://ServerName/site/DynamicsAx"
```

Обновить все компоненты веб-сайта Корпоративного портала:

```
AxUpdatePortal -updateall -websiteurl "http://ServerName/site/DynamicsAx"
```

Развернуть все прокси-сборки на веб-сайте Корпоративного портала:


```
AxUpdatePortal -proxies -websiteurl "http://ServerName/site/DynamicsAx"
```

Развернуть веб-элемент управления Customers на веб-сайте Корпоративного портала:

```
AxUpdatePortal -updatewebcomponent -treenodepath "\\Web\Web Files\Web Controls\  
Customers" -websiteurl "http://ServerName/site/DynamicsAx"
```

Корпоративный поиск

Корпоративный поиск в Microsoft Dynamics AX 2012 позволяет пользователям осуществлять поиск в данных, метаданных и содержимом документов, прикрепленных к записям. Этот поиск доступен как в Корпоративном портале, так и в клиенте Microsoft Dynamics AX. Корпоративный поиск для сбора данных и метаданных Microsoft Dynamics AX использует сервисы Metadata и Query из Microsoft Dynamics AX. Для индексирования и выполнения поисковых запросов Корпоративный поиск использует SharePoint Business Connectivity Services (BCS).

Чтобы сделать доступной эту поисковую возможность, вы должны установить с помощью установщика Microsoft Dynamics AX компонент Enterprise Search. Если вы используете редакции SharePoint Standard или Enterprise, то никакие дополнительные компоненты для работы Корпоративного поиска устанавливать не нужно, поскольку в эти редакции уже встроены поисковые возможности. Однако в случае SharePoint Foundation перед установкой Корпоративного поиска вам понадобится предварительно установить Microsoft Search Server Express.

Корпоративный поиск использует запросы, чтобы сделать данные в Microsoft Dynamics AX доступными для поиска. После установки Корпоративный поиск индексирует запросы по умолчанию и запускает полный сбор данных и метаданных.

Если вы хотите сделать доступными для поиска дополнительные данные, необходимо выполнить следующие шаги.

1. В АОТ либо найдите запрос, который выбирает нужные данные, либо создайте такой запрос.
2. Установите свойство запроса *Searchable* в *Yes*.
3. Скомпилируйте запрос и убедитесь, что при этом не выявлено никаких нарушений Best Practice.
4. В клиенте Microsoft Dynamics AX запустите Мастер конфигурации Корпоративного поиска (Администрирование системы > Настройка

> Поиск > Мастер конфигурации), показанный на рис. 7-16. Мастер отобразит список всех запросов в АОТ, чье свойство *Searchable* установлено в *Yes*.

По умолчанию все запросы, чье свойство *Searchable* установлено в *Yes*, будут выбраны для публикации в **Microsoft Business Connectivity Services (BCS)**. **Вы можете снять отметку у тех запросов, которые не хотите публиковать.**

5. Если необходимо, выберите опцию **Выбрать поля**, чтобы исключить те или иные поля из индексирования.
6. Установите галочку для запуска полной обработки данных. Вы также можете использовать **Центр администрирования SharePoint для запуска** полной или инкрементной обработки данных.
7. Щелкните **Далее** и затем **Готово**.

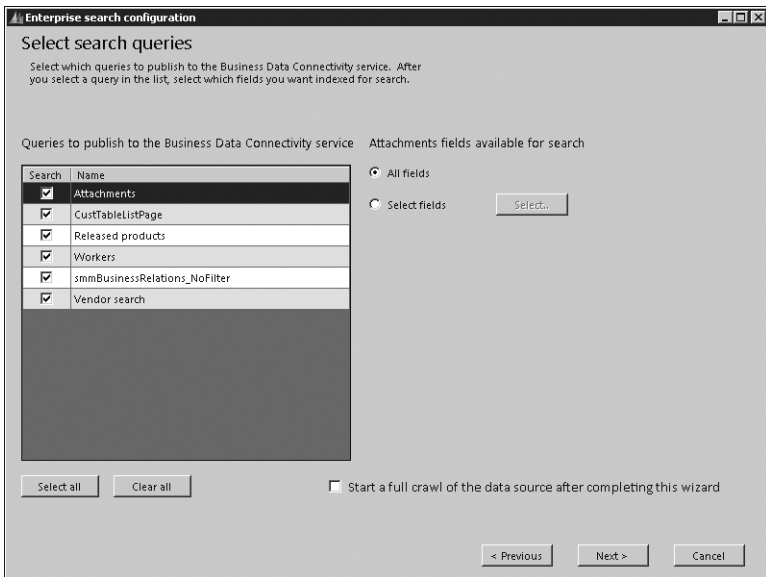


Рис. 7-16. Мастер конфигурации Корпоративного поиска

Мастер конфигурации Корпоративного поиска использует учетную запись агента данных для индексирования данных и публикации запросов в BCS. **Вы также можете увидеть своих опубликованные запросы в Центре администрирования SharePoint, в меню Управление приложениями >**

Управление служебными приложениями > Служба Business Data Connectivity Service (Application Management > Manage Service Applications > Business Data Connectivity Service).

Изменения в метаданных (таких как веб-меню и т.д.) достаточно редки, поэтому Корпоративный поиск по умолчанию выполняет полную обработку метаданных только один раз во время установки. Изменения же в данных (таких как заказы на продажу и т.п.), напротив, происходят часто. По умолчанию Корпоративный поиск выполняет полную обработку данных единожды при установке и инкрементную обработку – каждый день в полночь.

Если вы опубликовали новый запрос, то можете запустить полную обработку данных непосредственно из Мастера конфигурации Корпоративного поиска, как указывалось ранее. Вы также можете запустить полную или инкрементную обработку данных вручную из Центра администрирования SharePoint, выполнив следующие шаги.

1. Запустите Центр администрирования SharePoint.
2. Перейдите в Управление приложениями > Управление служебными приложениями > Служба поиска (Application Management > Manage Service Applications > Search Service Application).
3. На навигационной панели выберите Сканирование > Источники контента (Crawling > Content Sources). **Вы увидите два источника контента:** один – для данных, другой – для метаданных.
4. Щелкните по нужному источнику и затем нажмите либо Запустить полное сканирование (Start Full Crawl), либо Запустить инкрементное сканирование, как показано на рис. 7-17. Помните, что этот процесс может занять длительное время, в зависимости от того, насколько много данных или метаданных необходимо проиндексировать.

После того как данные и метаданные просканированы и проиндексированы, они публикуются в службе BCS. Пользователи могут выполнять поисковые запросы в окне поиска, расположенного в правом верхнем углу страницы Корпоративного портала. Результаты усекаются во время поиска на основе информации о роли, назначенной пользователю, региональных и других настроек, так что пользователи видят лишь те данные, которые им доступны и применимы к ним.

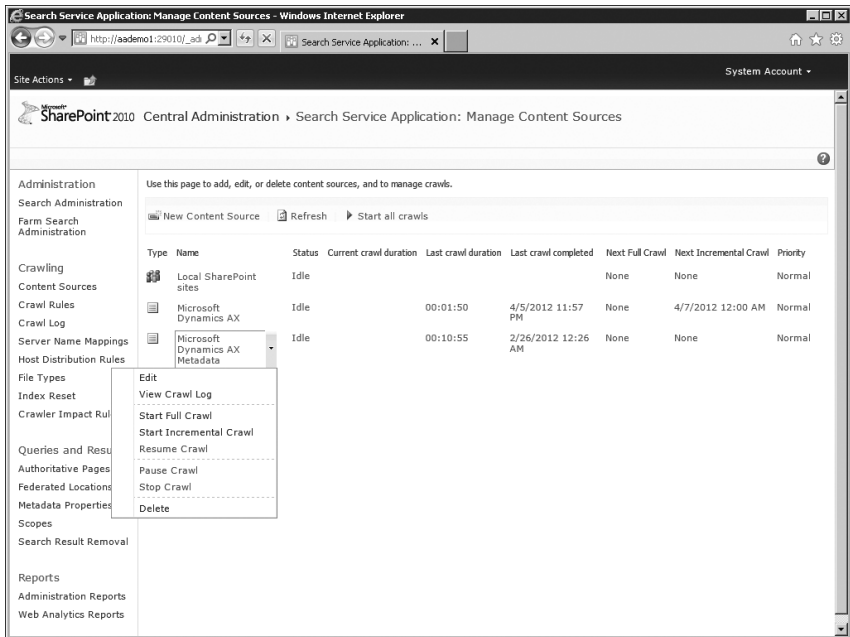


Рис. 7-17. Запуск индексации данных для поиска из Центра администрирования SharePoint

Темы

Корпоративный портал поддерживает темы SharePoint. Вы можете применить существующую тему SharePoint к сайту Корпоративного портала для изменения его внешнего вида точно так же, как применяете их к любому другому сайту SharePoint. Партнеры и клиенты также могут создавать новые темы и изменять или расширять таблицы стилей Корпоративного портала для приведения их в соответствие с новой темой.

Корпоративный портал использует пять файлов с таблицами стилей. AXEP.css является основным файлом. AXEP_RTL.css используется для языков, в которых направление письма – справа налево, и основывается на AXEP.css. Эти два файла располагаются на веб-сервере в каталоге <drive>\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\TEMPLATE\LAYOUTS\<lcid>\STYLES\Themable. Таблицы стилей AXEP_CRC.css, AXEP_CRC_RTL.css и AXEP_WebPart_Padding.css используются для ролевых центров при отображении их в Windows-клиенте

Microsoft Dynamics AX. Эти файлы располагаются на веб-сервере в каталоге `<drive>:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\TEMPLATE\LAYOUTS\ep\Stylesheets`.

Шаблонная страница Корпоративного портала ссылается на эти таблицы стилей. `AXEP.css` и `AXEP_RTL.css` содержат директивы для тем SharePoint, и потому располагаются в специальном каталоге, где SharePoint сможет их найти.

Когда тема SharePoint применяется к сайту Корпоративного портала, SharePoint проводит синтаксический разбор директив и вносит изменения, отражающие новую тему. Эти изменения включают замену цветов и начертаний шрифтов и даже изменение палитры некоторых изображений. Затем измененные таблицы стилей и изображения сохраняются в базе данных содержимого SharePoint. После этого шаблон страницы уже ссылается на эти новые таблицы стилей, чтобы примененная тема нашла отражение во внешнем виде Корпоративного портала.