OBA Composition Reference Toolkit Version 2.0

Packaging and Design Guidelines for Components and Adapters

Overview

This document describes the packaging and design guidelines for OBA components and adapters used to integrate OBA components with LOB systems.

The target audiences for this document are developers.

It is recommended that you walk through the OBA Composition Reference Toolkit Overview and User Guide documents before reading this document.

Fundamental Concepts

This section describes the concepts of components and LOB adapters that enable integrating components with line-of-business systems. Subsequent sections will describe the design and packaging guidelines for components and LOB adapters built to integrate with the OBA Composition Reference Toolkit.

Components

Components are the building blocks of OBAs composed and deployed using the OBA Composition Reference Toolkit. There are three types of components that can be registered in an OBA Composition Server and used to compose OBAs:

- **Presentation components**. These are used to generate the user interfaces of OBAs. Examples of presentation components include Microsoft Visual Studio Tools for Office (VSTO) add-ins for the Microsoft Office applications, SharePoint portal pages, Web Parts, and reports.
- **Collaboration components**. These are used to model unstructured artifacts and enable unstructured information worker collaboration. Examples of collaboration components include SharePoint workflows, SharePoint document libraries/lists, and content types.
- **Domain model components**. These are used to model abstracted business data models used for line-of-business application integration and business intelligence. Examples of domain model components include Business Data Catalog (BDC) application definitions, SQL Server Reporting Services Report models, and SQL Server Analysis Services Unified Dimensional Model (UDM) models.

LOB Adapters and Adapter Instances

Line-of-business (LOB) adapters are data source abstractions that enable integrating components with line-of-business data sources and services. One or more LOB adapter instances materialize an LOB adapter to enable integrating with specific semantically compliant instances of an adapter. For example, a customer relationship management (CRM) LOB adapter can establish a common semantic interface for integrating customer relationship data and can have Microsoft Dynamics CRM and Siebel CRM adapter

instances to enable integrating physical CRM data from two different CRM systems that implement the common semantic interfaces.

This is a powerful concept because it enables reusing components against multiple LOB data sources that are semantically similar.

Figure 1 and Figure 2 illustrate the interplay between components, LOB adapters, and LOB adapter instances.

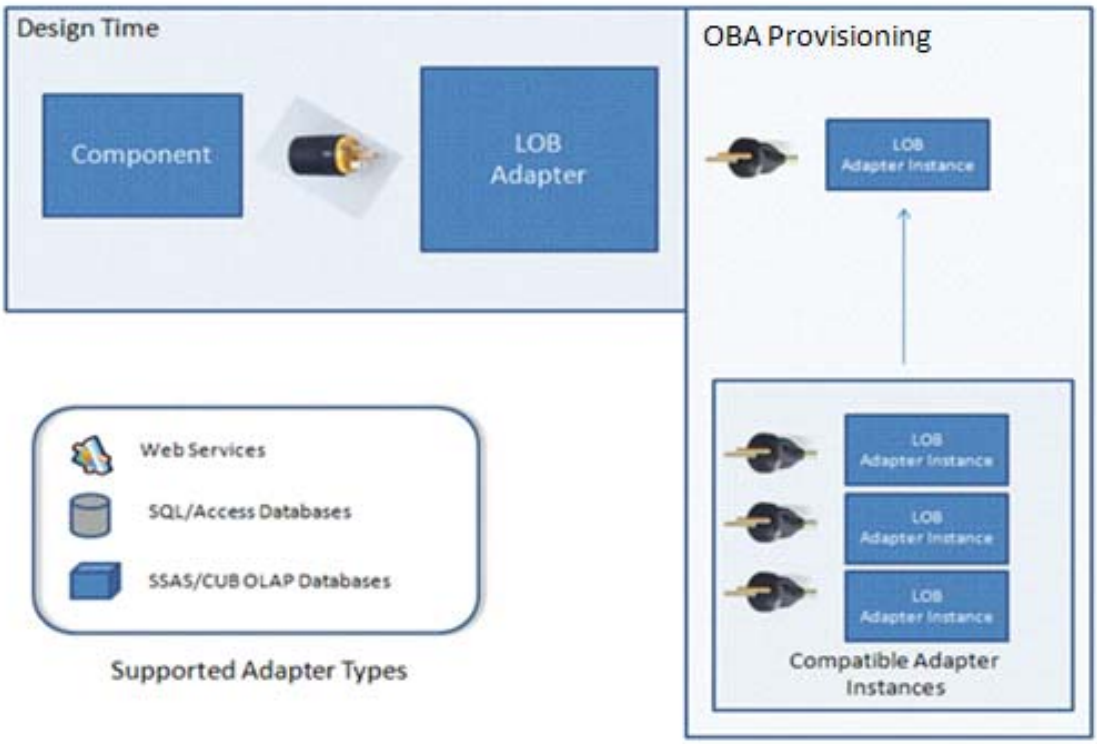

Figure 1: Components and LOB adapters - architecture

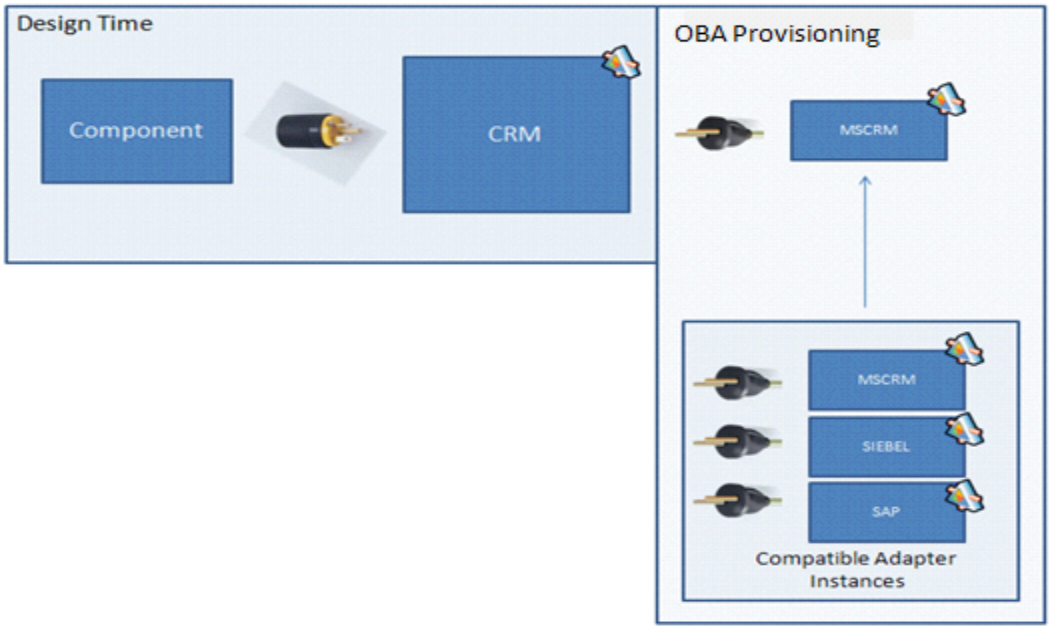

Figure 2: Components and LOB adapters - example

Designing and Packaging LOB Adapter Instances

This section describes the design and packaging guidelines that must be followed to integrate LOB adapters in the OBA Composition Reference Toolkit. These guidelines provide structure to the standard development practices that enable the materialization of composition solution architectures.

Sample packages for each of the supported adapter types are included in the source code/artifacts included in version 2 of the OBA Composition Reference Toolkit.

LOB Web Service (ASMX Services Only)

Package contents	Web site precompiled output
Packaging format	Publish/precompile output of the ASP.NET Web service project delivered as a .zip file.
Supporting adapter(s)	• SQL Server 2005 database • Microsoft Office Access 2007 database
Design guidelines	• Physical data source connections in the service must be resolved from the service's Web.config file. • Sample Web.config: <?xml version="1.0"?> <configuration> <connectionStrings> **<add name="CRM" connectionString="Data Source=localhost;Initial Catalog=AdventureWorks; Integrated Security=True" providerName="System.Data.SqlClient"/>**

	</connectionStrings>
	<system.web>
	<compilation debug="false" />
	<authentication mode="Windows"/>
	<!--
	OBA: Following line is required to deploy
	LOB web services in MOSS
	-->
	<trust level="Full" originUrl=""/>
	</system.web>
	</configuration>
	
	• Connection string name must be the **name of the LOB Adapter**. This will be replaced at run time by provisioning service.
	
	• Sample Web service code to retrieve database connection string:
	
	String crmConnStr =
	`ConfigurationManager.`
	`ConnectionStrings["CRM"].ConnectionString;`
	
	SqlConnection conn = new SqConnection();
	`conn.ConnectionString = crmConnStr;`

Sample	[Installation Folder]\Source code\Seed Components\Insurance Endorsements\InsuranceWS

SQL Server 2005 Database

Package contents	Database objects provisioning script
Packaging format	SQL script to create database objects and insert demo data. Script must be named CreateDBObjects.sql and delivered as a .zip file.
Supporting adapter(s)	Not applicable
Design guidelines	• The script must create a schema (schema name = **LOB adapter name**) within which all database objects must be created. • OBA components that use objects in the database must access the objects using their schema qualified names.
Sample	[Installation Folder]\Source code\Seed Components\Human Life Sciences\LifeScienceDB

Access 2007 Database

Package contents	.mdb or .accdb file
Packaging format	.mdb or .accdb file delivered as a .zip file

Supporting adapter(s)	Not applicable
Design guidelines	None
Sample	[Installation Folder] \Source code\Seed Components\Insurance Endorsements\InsuranceWS

Offline OLAP Database (.cub File Created Using Excel 2007)	
Package contents	.cub file
Packaging format	.cub file delivered as a .zip file
Supporting adapter(s)	Not applicable
Design guidelines	None
Sample	None

SSAS OLAP Database	
Package contents	SQL Server Analysis Server (SSAS) database backup
Packaging format	.abf file (SSAS database backup file) delivered as a .zip file
Supporting adapter(s)	• SQL Server 2005 database • Access 2007 database
Design guidelines	**Data source names in the UDM must match LOB adapter names.**
Sample	None

Designing and Packaging OBA Components

This section describes the packaging guidelines that must be followed to integrate instances of the supported OBA component types in the OBA Composition Reference Toolkit. These guidelines provide structure to the standard development practices that enable the materialization of composition solution architectures.

Sample packages for each of the supported component types are included in the source code/artifacts included in version 2 of the OBA Composition Reference Toolkit.

SharePoint List	
Package contents	• SharePoint feature definition for the list • Compiled assemblies • Updated Config.xml file
Packaging format	Package contents in the following folder structure delivered as a .zip file: < Component Name> - Config.XML - BIN - RELEASE

<table>
<tr><td></td><td>- <Signed Assembly: Receivers, Event Handler, etc>
- TEMPLATE
- CONTROLTEMPLATES
- <Scenario Name>
- <custom field def user controls: .ascx>
- XML
- <field definitions: .xml>
- FEATURES
- <Feature Name>
- feature.xml
- <element manifests: .xml>
- <list name>
- schema.xml
- <list form files: .aspx, .htm>
- LAYOUTS
- <Scenario Name>
- <List supporting ASPX application pages>
- <images></td></tr>
<tr><td>Supported LOB data sources</td><td>• BDC application definitions referenced by BDC list columns (BDC application definitions will be supporting tier 1 components when referenced by a list).
• LOB Web services (event receivers/handlers).</td></tr>
<tr><td>Design guidelines</td><td>• SharePoint solution generator can be used to generate list schemas and associated list forms. The definitions may need to be created manually if the solution generator does not work for a specific list.
• All files included should be required for deployment.
• Feature must have unique GUID and must be scoped to user mashup Web site.

• Sample feature definition:
<Feature Id="{}"
Title="Sample Feature Definition"
Scope="Web"
Hidden="FALSE"
…
xmlns="http://schemas.microsoft.com/sharepoint/">
…
</Feature>

• Recommended <ScenarioName>.<Component Name> as the feature name.</td></tr>
<tr><td>Coding guidelines for LOB data access</td><td>• Connection string information for LOB adapters will be available in the SharePoint Web.config file.
• Connection string key names used in the config file must match the names of the LOB adapters used by the component.

• Sample Web.config file:</td></tr>
</table>

	<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
	<configuration>
	…
	<appSettings>
	…
	<add key="CRM" value="http://obadevwks/CRM/Service.asmx" />
	</appSettings>
	…
	</configuration>
	
	• Sample code to retrieve connection string information from event handlers:
	String svcUrl = ConfigurationManager.AppSettings["CRM"];
	
	CRMService svc = new CRMService ();
	svc.Url = svcUrl;
	svc.Credentials = System.Net.CredentialCache.DefaultCredentials;
	
	CustomersDS customers = svc.GetCustomers();

Configuration file	Configuration file (Config.xml) must be included to deploy custom assemblies to the global assembly cache. Sample Config.xml file: <?xml version="1.0" encoding="utf-8" ?> <Install> <Gac> <Assembly> <FileName>ScnearioName.ComponentName.dll</FileName> </Assembly> </Gac> </Install>
Configuration file for page display	List view Web parts can be added to the pre-existing SharePoint content pages using WebPart tag in the SitePages tag. Sample Config.xml file: <?xml version="1.0" encoding="utf-8" ?> <Install> <SitePage Url="default.aspx"> <WebPart ListName="ComponentName" ListViewName="All Items" Title="All uploaded documents" ZoneIndex="0" Type="ListView" ZoneId="Right" /> </WebPart> </SitePage> </Install>
Sample	[Installation Folder]\Source code\Seed Components\Insurance

	Endorsements\MindTree.Tasks

SharePoint Document Library	
Package contents	• SharePoint feature for the document library • Compiled assemblies • Updated Config.xml file
Packaging format	SharePoint feature files in the following folder structure delivered as a .zip file: <Component Name> - Config.XML - BIN - RELEASE - <Signed Assembly: Receivers, Event Handlers, etc> - WSS - TEMPLATE - CONTROLTEMPLATES - <Scenario Name> - <custom field def user controls: .ascx> - XML - <field definitions: .xml> - FEATURES - <Feature Name> - feature.xml - <element manifests: .xml> - SAMPLE DOCUMENTS - < files: .docx, .xml, etc> - <list name> - schema.xml - <list form files: .aspx, .htm> - LAYOUTS - <Scenario Name> - <Supporting ASPX application pages> - <images>
Supported LOB data sources	• BDC application definitions referenced by BDC list columns (BDC application definitions will be supporting tier 1 components when referenced by a list). • LOB Web services (event receivers/handlers).
Design guidelines	• SharePoint solution generator can be used to generate list schemas and associated list forms. The definitions may need to be created manually if the solution generator does not work for a specific list. • All files included should be required for deployment. • Feature must have unique GUID and must be scoped to user mashup Web site.

	• Sample feature definition:
	`<Feature Id="{}"`
	`  Title="Sample Feature Definition"`
	`  Scope="Web"`
	`  Hidden="FALSE"`
	`…`
	`  xmlns="http://schemas.microsoft.com/sharepoint/">`
	`…`
	`</Feature>`
	
	• Recommended <ScenarioName>.<Component Name> as the feature name.

Coding guidelines for LOB data access	• Connection string information for LOB adapters will be available in the SharePoint Web.config file.
	• Connection string key names used in the config file **must match the names of the LOB adapters** used by the component.
	
	• Sample Web.config file:
	<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
	<configuration>
	…
	<appSettings>
	…
	<add key="CRM" value="http://obadevwks/CRM/Service.asmx" />
	</appSettings>
	…
	</configuration>
	
	• Sample code to retrieve connection string information from event handlers code:
	String svcUrl = ConfigurationManager.AppSettings["CRM"];
	
	CRMService svc = new CRMService ();
	svc.Url = svcUrl;
	svc.Credentials = System.Net.CredentialCache.DefaultCredentials;
	
	CustomersDS customers = svc.GetCustomers();

Configuration file	Configuration file (Config.xml) must be included to deploy custom assemblies to the global assembly cache. Sample Config.xml file: <?xml version="1.0" encoding="utf-8" ?> <Install> <Gac> <Assembly> <FileName>ScenarioName.ComponentName.dll</FileName> </Assembly> </Gac>

	</Install>
Configuration file for page display	List view Web parts can be added to the pre-existing SharePoint content pages using WebPart tag in the SitePages tag. Sample Config.xml file: <?xml version="1.0" encoding="utf-8" ?> <Install> <SitePage Url="default.aspx"> <WebPart ListName="ComponentName" ListViewName="All Items" Title="All uploaded documents" ZoneIndex="0" Type="ListView" ZoneId="Right" /> </WebPart> </SitePage> </Install>
Sample	[Installation Folder]\Source code\Seed Components\Insurance Endorsements\MindTree.SharedDocuments

SharePoint Content Types

Package contents	• SharePoint feature for the content type • Compiled assemblies • Updated Config.xml file
Packaging format	SharePoint feature files in the following folder structure delivered as a .zip file: <Component Name> - Config.XML - BIN - RELEASE - <Signed Assembly: Receivers, Event Handlers, etc> - WSS - TEMPLATE - CONTROLTEMPLATES - <Scenario Name> - <custom field def user controls: .ascx> - XML - <field definitions: .xml> - FEATURES - <Feature Name> - feature.xml - <element manifests: .xml> - <feature/template files: .dotx, .docx, etc>

<table>
<tr><td></td><td>- LAYOUTS
- <Scenario Name>
- <Supporting ASPX application pages>
- <images></td></tr>
<tr><td>Supported LOB data sources</td><td>LOB Web services (event receivers/handlers)</td></tr>
<tr><td>Design guidelines</td><td>• Content type should be used for document templates.
• All files included should be required for deployment.
• Feature must have unique GUID and must be scoped to user mashup Web site.

• Sample feature definition:
<pre><Feature Id="{}"
  Title="Sample Feature Definition"
  Scope="Site"
  Hidden="FALSE"
…
  xmlns="http://schemas.microsoft.com/sharepoint/">
…
</Feature></pre>
• Recommended <ScenarioName>.<Component Name> as the feature name.</td></tr>
<tr><td>Coding guidelines for LOB data access</td><td>• Connection string information for LOB adapters will be available in the SharePoint Web.config file.
• Connection string key names used in the config file must match the names of the LOB adapters used by the component.

• Sample Web.config file:
<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<configuration>
…
 <appSettings>
…
  <add key="CRM" value="http://obadevwks/CRM/Service.asmx" />
 </appSettings>
…
</configuration></pre>
• Sample code to retrieve connection string information from event handlers code:
<pre>String svcUrl = ConfigurationManager.AppSettings["CRM"];

CRMService svc = new CRMService ();
svc.Url = svcUrl;
svc.Credentials = System.Net.CredentialCache.DefaultCredentials;

CustomersDS customers = svc.GetCustomers();</pre></td></tr>
<tr><td>Configuration file</td><td>Configuration file (Config.xml) must be included to deploy custom</td></tr>
</table>

	assemblies to the global assembly cache. Sample Config.xml file: <?xml version="1.0" encoding="utf-8" ?> <Install> <Gac> <Assembly> <FileName>ScenarioName.ComponentName.dll</FileName> </Assembly> </Gac> </Install>
Sample	[Installation Folder]\ Source code\Seed Components\Human Life Sciences\Drug Research Analysis

SharePoint Workflows	
Package contents	• SharePoint feature for the workflow • Compiled assemblies • Updated Config.xml file
Packaging format	SharePoint feature files in the following folder structure delivered as a .zip folder: <Component Name> - Config.XML - BIN - RELEASE - <Signed Assembly: Workflow, Receivers, etc> - WSS - TEMPLATE - FEATURES - <Feature Name> - feature.xml - <workflow: .xml> - <feature files: InfoPath - .xsn, etc> - <supporting folders> - <feature files: InfoPath - .xsn, Resources: .resx etc> - LAYOUTS - <Scenario Name> - <images> - <Workflow ASPX pages: Initiation, Association, Modification , Status, Task Display, Task Edit, etc>
Supported LOB data sources	LOB Web services (event receivers/handlers)
Design guidelines	• All files included should be required for deployment. • Feature must have unique GUID and must be scoped to user OBA home site.

<table>
<tr><td></td><td>
Sample feature definition:
<pre>
<Feature Id="{}"
  Title="Sample Feature Definition"
  Scope="Site"
  Hidden="FALSE"
…
  xmlns="http://schemas.microsoft.com/sharepoint/">
…
</Feature>
</pre>
Sample workflow definition:
<pre>
<Workflow Id="{}"
  Name="Sample Workflow"
…
  xmlns="http://schemas.microsoft.com/sharepoint/">
…
</Workflow>
</pre>
Recommended <ScenarioName>.<Component Name> as the feature name.
</td></tr>
<tr><td>Coding guidelines for LOB data access</td><td>
Connection string information for LOB adapters will be available in the SharePoint Web.config file.
Connection string key names used in the config file must match the names of the LOB adapters used by the component.
Sample Web.config file:
<pre>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<configuration>
…
 <appSettings>
…
  <add key="CRM" value="http://obadevwks/CRM/Service.asmx" />
 </appSettings>
…
</configuration>
</pre>
Sample code to retrieve connection string information from event handlers code:
<pre>
String svcUrl = ConfigurationManager.AppSettings["CRM"];

CRMService svc = new CRMService ();
svc.Url = svcUrl;
svc.Credentials = System.Net.CredentialCache.DefaultCredentials;

CustomersDS customers = svc.GetCustomers();
</pre>
</td></tr>
<tr><td>Configuration file</td><td>Configuration file (Config.xml) must be included to deploy custom assemblies to the global assembly cache.

Sample Config.xml file:</td></tr>
</table>

	<?xml version="1.0" encoding="utf-8" ?> <Install> <Gac> <Assembly> <FileName>ScenarioName.ComponentName.dll</FileName> </Assembly> </Gac> </Install>
Sample	[Installation Folder] \Source code\Seed Components\Expense Reporting\ER.ApprovalWorkflow

VSTO Document Add-in

Package contents	VSTO ClickOnce output
Packaging format	ClickOnce publish output delivered as a .zip file
Supported LOB data sources	LOB Web service
Design guidelines	Visual Studio 2008 must be used to created VSTO document add-ins.
Coding guidelines for LOB data access using OBADataCOnnections.xml	• The OBA Data Connections file consists of provisioned OBA configuration in the order they were provisioned and the adapter information. The OBA configuration can be used in add-ins to connect to the OBA, and adapter information can be used to connect to the data sources. • LOB connection string or Web Services Description Language (WSDL) information must be resolved from < My Data Sources >\OBADataConnections.xml file. OBADataConnections.xml must be downloaded by the user from the Data Connections Library in his/her Mashups Site Collection root site and copied into his/her **My Data Sources** folder. • The LOB adapter name must be used in the implementation to resolve the LOB adapter connection string (obtained from OBADataConnections.xml). • Sample OBAComponents.xml file: <OBADataConnections> <Mashups> <Mashup> <Name>OBA 1</Name> <SiteUrl>http://obadevwks/OBA1</SiteUrl> <VSTOAddins /> </Mashup> <Mashup> <Name>OBA 2</Name> <SiteUrl>http://obadevwks/OBA2</SiteUrl> <VSTOAddins /> </Mashup>

	</Mashups>
	<OBADataConnection>
	<Adapter>CRM</Adapter>
	<AdapterType>LOBService</AdapterType>
	<ConnectionString></ConnectionString>
	</OBADataConnection>
	<OBADataConnection>
	<Adapter>SCM</Adapter>
	<AdapterType>SQLDB</AdapterType>
	<ConnectionString></ConnectionString>
	</OBADataConnection>
	</OBADataConnections>
	
	• Sample code to retrieve SQL Server connection string from add-in:
	string path = System.Environment.GetFolderPath(
	Environment.SpecialFolder.MyDocuments);
	path = Path.Combine(path, @"\My Data Sources\OBADataConnections.xml");
	
	XmlDocument config = new XmlDocument();
	config.Load(path);
	
	XmlNode svcUrl = config.SelectSingleNode(
	"/OBADataConnections/
	OBADataConnection[Adapter='CRM']/ConnectionString").InnerText
	;
	
	CRMService svc = new CRMService ();
	svc.Url = svcUrl;
	svc.Credentials = System.Net.CredentialCache.DefaultCredentials;
	
	CustomersDS customers = svc.GetCustomers();

Coding guidelines for LOB data access using Metadata Web service	• The metadata Web service methods for OBA data connections consist of provisioned OBA configuration and the adapter information. The OBA configuration can be used in add-ins to connect to the OBA and adapter information can be used to connect to the data sources. • Steps involved to use metadata web methods to retrieve OBA configuration and adapter information: 1. Get registry value **Manifest** from the **HKEY_CURRENT_USER\Software\Microsoft\Office\Word\Addins\<AddIn Name>** registry key. 2. Using the manifest URL, access the meta data services. 3. Use OBA Data Connections Web methods to retrieve adapter connection strings and OBA information. • Sample code: using (RegistryKey key = Registry.CurrentUser.OpenSubKey(@"Software\Microsoft\Office\Word\Addins\<addin name>")) {

	``
	  Uri manifestUri = new
	Uri(Convert.ToString(key.GetValue("Manifest")));
	  string metadataSvcUrl =
	string.Format("http://{0}:{1}/CompositionServer/Service.asmx",
	  manifestUri.Host, manifestUri.Port);
	  MetadataService.Service svc = new
	OBADataAccess.MetadataService.Service();
	  svc.Url = metadataSvcUrl;
	  svc.UseDefaultCredentials = true;
	
	  // To get all adapter connections
	  MetadataService.AdapterConnectionInfo[] connections =
	svc.GetAllAdapterConnections();
	
	  // To get connection information for an adapter
	  string connectionString =
	svc.GetAdapterConnectionString("AdapterName");
	
	  // To get all Mashup/OBA connections
	  MetadataService.MashupConnectionInfo[] mashupConnections
	= svc.GetAllMashupConnections();
	
	  // To get all Mashup/OBA connections
	  string lastMashupUrl = svc.GetLastMashupConnectionUrl();
	}
	``

Sample	[Installation Folder]\ Source code\Seed Components\Human Resources\NEH.OfferLetterTemplate (incomplete)

VSTO Application Add-in

Package contents	VSTO ClickOnce output
Packaging format	ClickOnce publish output delivered as a .zip file
Supported LOB data sources	LOB Web service
Design guidelines	• Visual Studio 2008 must be used to created VSTO document add-ins. • Recommended <ScenarioName>.<Component Name> as the project name.
Coding guidelines for LOB data access using OBADataCOnnections.xml	• The OBA Data Connections file consists of provisioned OBA configuration in the order they were provisioned and the adapter information. The OBA configuration can be used in add-ins to connect to the OBA and adapter information can be used to connect to the data sources. • LOB connection string or WSDL information must be resolved from < My Data Sources >\OBADataConnections.xml file. OBADataConnections.xml must be downloaded by the user from the Data Connections Library in his/her Mashups Site Collection root site and copied into his/her **My Data Sources** folder. • The LOB adapter name must be used in the implementation to resolve the LOB Adapter Connection String (obtained from

	OBADataConnections.xml).
	
	• Sample OBAComponents.xml file:
	<OBADataConnections>
	<Mashups>
	<Mashup>
	<Name>OBA 1</Name>
	<SiteUrl>http://obadevwks/OBA1</SiteUrl>
	<VSTOAddins />
	</Mashup>
	<Mashup>
	<Name>OBA 2</Name>
	<SiteUrl>http://obadevwks/OBA2</SiteUrl>
	<VSTOAddins />
	</Mashup>
	</Mashups>
	<OBADataConnection>
	<Adapter>CRM</Adapter>
	<AdapterType>LOBService</AdapterType>
	<ConnectionString></ConnectionString>
	</OBADataConnection>
	<OBADataConnection>
	<Adapter>SCM</Adapter>
	<AdapterType>SQLDB</AdapterType>
	<ConnectionString></ConnectionString>
	</OBADataConnection>
	</OBADataConnections>
	
	• Sample code to retrieve SQL Server connection string from add-in:
	string path = System.Environment.GetFolderPath(
	Environment.SpecialFolder.MyDocuments);
	path = Path.Combine(path, @"\My Data Sources\OBADataConnections.xml");
	
	XmlDocument config = new XmlDocument();
	config.Load(path);
	
	XmlNode svcUrl = config.SelectSingleNode(
	"/OBADataConnections/
	OBADataConnection[Adapter='CRM']/ConnectionString").InnerText
	;
	
	CRMService svc = new CRMService ();
	svc.Url = svcUrl;
	svc.Credentials = System.Net.CredentialCache.DefaultCredentials;
	
	CustomersDS customers = svc.GetCustomers();

Coding guidelines for LOB data access using	• The metadata Web service methods for OBA Data Connections consist of provisioned OBA configuration and the adapter

Metadata Web service	information. The OBA configuration can be used in add-ins to connect to the OBA and adapter information can be used to connect to the data sources.
	• Steps involved to use metadata Web methods to retrieve OBA configuration and adapter information:
	1. Get registry value **Manifest** from the **HKEY_CURRENT_USER\Software\Microsoft\Office\Word\Addins\<AddIn Name>** registry key.
	2. Using the manifest URL, access the metadata services.
	3. Use OBA Data Connections Web methods to retrieve adapter connection strings and OBA information.
	• Sample code:
	`using (RegistryKey key = Registry.CurrentUser.OpenSubKey(`
	`  @"Software\Microsoft\Office\Word\Addins\<addin name>"))`
	`{`
	`  Uri manifestUri = new`
	`Uri(Convert.ToString(key.GetValue("Manifest")));`
	`  string metadataSvcUrl =`
	`string.Format("http://{0}:{1}/CompositionServer/Service.asmx",`
	`  manifestUri.Host, manifestUri.Port);`
	`  MetadataService.Service svc = new`
	`OBADataAccess.MetadataService.Service();`
	`  svc.Url = metadataSvcUrl;`
	`  svc.UseDefaultCredentials = true;`
	
	`  // To get all adapter connections`
	`  MetadataService.AdapterConnectionInfo[] connections =`
	`svc.GetAllAdapterConnections();`
	
	`  // To get connection information for an adapter`
	`  string connectionString =`
	`svc.GetAdapterConnectionString("AdapterName");`
	
	`  // To get all Mashup/OBA connections`
	`  MetadataService.MashupConnectionInfo[] mashupConnections`
	`= svc.GetAllMashupConnections();`
	
	`  // To get all Mashup/OBA connections`
	`  string lastMashupUrl = svc.GetLastMashupConnectionUrl();`
	`}`

Sample	[Installation Folder]\Source code\Seed Components\Insurance Endorsements\MindTree.DocumentManagementAddIn

BDC Application Definitions

Package contents	Application definition XML file
Packaging format	Application definition XML file delivered as a .zip file
Supporting adapter(s)	• LOB Web service • SQL Server

Design guidelines	**LOBSystemInstance name in the BDC application definition must match the internal name of the source LOB adapter.**
Sample	[Installation Folder]\Source code\Seed Components\Plant Floor Operations\Statera.PlantFloorOpsBDC

InfoPath Forms

Package contents	• InfoPath form published using administrator-approved form template method • Universal Data Connection (UDCX) files (generated by converting form connections to UDCX and by choosing the relative link type location option) • Config.xml that includes mapping of UDCX files and the adapters/SharePoint lists
Packaging format	The following files delivered as a .zip file: • .xsn file • UDCX files • Config.xml
Supported LOB data sources	• LOB Web service • SharePoint lists/document libraries
Design guidelines	• InfoPath form must be published as "administrator-approved form template." • Form connections must be converted to UDCX files and by selecting the relative link type location option. • InfoPath form must be browser-enabled. • InfoPath form must be deployed and tested from within a SharePoint site and must be verified to ensure that it is able to resolve its data connections using the UDCX files in the Data Connections library in the parent site collection's root site.
Coding guidelines for LOB data access	• All LOB access must be made using UDCX data connection files. • UDCX files must be stored in a data connections library named "Data Connections" in the root site of the site collection to which the form template is activated. • Update Config.xml with mapping of UDCX files and supporting adapters/SharePoint lists or document libraries. • Config.xml sample: <?xml version="1.0" encoding="utf-8" ?> <Install> <DataConnections> <DataConnection> <FileName>Fine_ERP_ER_CreateManager.udcx</FileName> <AdapterType>**LOB**</AdapterType> <AdapterName>Fine_ERP.WebService</AdapterName> </DataConnection> <DataConnection>

	<FileName>Fine_ERP_ER_GetDepartments.udcx</FileName> <AdapterType>**LOB**</AdapterType> <AdapterName>Fine_ERP.WebService</AdapterName> </DataConnection> <DataConnection> <FileName>SP_ER_SubmitExpenseReport.udcx</FileName> <AdapterType>**SPList**</AdapterType> <AdapterName>ER.ExpenseReportLib</AdapterName> </DataConnection> </DataConnections> </Install> • Supported adapter type values: **LOB** and **SPList** • AdapterName corresponding to LOB AdapterType must be **Adapter Internal Name.** • AdapterName corresponding to SharePoint list or document library must be **List or Document Library Component Internal Name.** • **The SharePoint List or Document Library must be selected as Supporting DataSource in Component metadata.**
Sample	[Installation Folder]\Source code\Seed Components\Expense Reporting\ER.ExpenseReportLib

## Web Parts	
Package contents	• SharePoint feature for the Web Parts • Compiled assemblies • Updated config.xml file
Packaging format	SharePoint feature files in the following folder structure delivered as a .zip file: <Component Name> - Config.XML - BIN - RELEASE - <Signed Assembly: Receivers, Webparts, etc> - WSS - TEMPLATE - CONTROLTEMPLATES - <Scenario Name> - <Webpart hosted user controls: .ascx> - FEATURES - <Feature Name> - feature.xml - <element manifests: .xml> - DWP

<table>
<tr><td></td><td>- <web parts definitions : .dwp>
- LAYOUTS
- <Scenario Name>
- <images>
- <Application Pages: .aspx files></td></tr>
<tr><td>Supported LOB data sources</td><td>• LOB Web service
• SQL Server
• Access database</td></tr>
<tr><td>Design guidelines</td><td>• All files included should be required for deployment.
• Feature must have unique GUID and must be scoped to user mashup Web site.

• Sample feature definition:
<Feature Id="{}"
Title="Sample Feature Definition"
Scope="Web"
Hidden="FALSE"
…
xmlns="http://schemas.microsoft.com/sharepoint/">
…
</Feature></td></tr>
<tr><td>Configuration file</td><td>• Configuration file (Config.XML) must be included to deploy custom assemblies to the global assembly cache and to register Web Parts to the SharePoint safe controls list.

• Sample Config.xml file:
<?xml version="1.0" encoding="utf-8" ?>
<Install>
<Gac>
<Assembly>
<FileName>ScenarioName.ComponentName.dll</FileName>
</Assembly>
</Gac>
<SafeControls>
<SafeControl Assembly=" Litware. CustomerFeedbackResults, Version=1.0.0.0, Culture=neutral, PublicKeyToken=224a41c8397cea5c" Namespace=" Litware.OBA.CustomerFeedback " TypeName="*" Safe="True" />
</ SafeControls >
</Install></td></tr>
<tr><td>Coding guidelines for LOB data access</td><td>• Connection string information for LOB adapters will be available in the SharePoint Web.config file.
• Connection string key names used in the config file must match the names of the LOB adapters used by the component.

• Sample Web.config file:
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<configuration></td></tr>
</table>

	…
	<appSettings>
	…
	<add key="CRM" value="http://obadevwks/CRM/Service.asmx" />
	</appSettings>
	…
	</configuration>
	
	• Sample code to retrieve connection string information from Web Part:
	String svcUrl = ConfigurationManager.AppSettings["CRM"];
	
	CRMService svc = new CRMService ();
	svc.Url = svcUrl;
	svc.Credentials = System.Net.CredentialCache.DefaultCredentials;
	
	CustomersDS customers = svc.GetCustomers();

Sample	[Installation Folder]\Seed Components\Plant Floor Operations\BSIL.AjaxBdcListWebPart

Application Pages

Package contents	• SharePoint feature for the application pages • Compiled assemblies • Updated config.xml file
Packaging format	SharePoint feature files in the following folder structure delivered as a .zip file: <Component Name> - Config.XML - BIN - RELEASE - <Signed Assembly: Receivers, Layout base pages, etc> - WSS - CONTROLTEMPLATES - <Scenario Name> - <Page embedded user controls: .ascx> - TEMPLATE - FEATURES - <Feature Name> - feature.xml - <element manifests: .xml> - LAYOUTS - <Scenario Name> - <images> - <Application Pages: .aspx files>

<table>
<tr><td>Supported LOB data sources</td><td>

- LOB Web service
- SQL Server
- Access database

</td></tr>
<tr><td>Design guidelines</td><td>

- All files included should be required for deployment.
- Each feature should have unique GUID.
- Recommended <Scenario Name>.<Component Name> as the feature name or a friendly name.
- Application pages are always scoped at the farm. Because of this, it is recommended to scope the feature to the farm.
- Sample feature definition:

```
<Feature Id="{}"
  Title="Sample Feature Definition"
  Scope="Farm"
  Hidden="FALSE"
…
  xmlns="http://schemas.microsoft.com/sharepoint/">
…
</Feature>
```

</td></tr>
<tr><td>Configuration file</td><td>

- Configuration file (Config.xml) must be included to deploy custom assemblies to the global assembly cache.
- Sample Config.xml file:

```
<?xml version="1.0" encoding="utf-8" ?>
<Install>
 <Gac>
  <Assembly>
   <FileName>Litware.<AssemblyName>.dll</FileName>
  </Assembly>
 </Gac>
</Install>
```

</td></tr>
<tr><td>Coding guidelines for LOB data access</td><td>

- Connection string information for LOB adapters will be available in the SharePoint Web.config file.
- Connection string key names used in the config file **must match the names of the LOB adapters** used by the component.
- Sample Web.config file:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<configuration>
…
 <appSettings>
…
  <add key="CRM" value="http://obadevwks/CRM/Service.asmx" />
 </appSettings>
…
</configuration>
```

- Sample code to retrieve connection string information from

</td></tr>
</table>

	Web Part:
	String svcUrl = ConfigurationManager.AppSettings["CRM"];
	
	CRMService svc = new CRMService ();
	svc.Url = svcUrl;
	svc.Credentials = System.Net.CredentialCache.DefaultCredentials;
	
	CustomersDS customers = svc.GetCustomers();

Sample	None

Site Pages

Package contents	• SharePoint feature for the site pages • Compiled assemblies • Updated config.xml file
Packaging format	SharePoint feature files in the following folder structure delivered as a .zip file: <Component Name> - Config.XML - BIN - RELEASE - <Signed Assembly: Receivers, Page Handlers, etc> - WSS - TEMPLATE - CONTROLTEMPLATES - <Scenario Name> - <Page embedded user controls: .ascx> - FEATURES - <Feature Name> - feature.xml - <element manifests: .xml> - <Pages Folder> - <Pages: .aspx, .master> - LAYOUTS - <Scenario Name> - <images> - <Application Pages: .aspx files>
Supported LOB data sources	• LOB Web service • SQL Server • Access database
Design guidelines	• All files included should be required for deployment. • Each feature should have unique GUID. • Recommended <ScenarioName>.<Component Name> as the feature name. • Feature must be scoped to user mashup Web site.

	• Sample feature definition:
	`<Feature Id="{}"`
	`  Title="Sample Feature Definition"`
	`  Scope="Web"`
	`  Hidden="FALSE"`
	`…`
	`  xmlns="http://schemas.microsoft.com/sharepoint/">`
	`…`
	`</Feature>`

Configuration file	• Configuration file (Config.xml) must be included to deploy custom assemblies to the global assembly cache. • Sample Config.xml file: <?xml version="1.0" encoding="utf-8" ?> <Install> <Gac> <Assembly> <FileName>CompanyName.ComponentName.dll</FileName> </Assembly> </Gac> <SafeControls> <SafeControl Assembly=" <Assembly Name>, Version=1.0.0.0, Culture=neutral, PublicKeyToken=224a41c8397cea5c" Namespace="<Web part control namespace>" TypeName="*" Safe="True" /> </ SafeControls > </Install>
Coding guidelines for LOB data access	• Connection string information for LOB adapters will be available in the SharePoint Web.config file. • Connection string key names used in the config file **must match the names of the LOB Adapters** used by the component. • Sample Web.config file: <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <configuration> … <appSettings> … <add key="CRM" value="http://obadevwks/CRM/Service.asmx" /> </appSettings> … </configuration> • Sample code to retrieve connection string information from Web Part: String svcUrl = ConfigurationManager.AppSettings["CRM"]; CRMService svc = new CRMService (); svc.Url = svcUrl;

	svc.Credentials = System.Net.CredentialCache.DefaultCredentials; CustomersDS customers = svc.GetCustomers();
Sample	[Installation Folder]\Seed Components\Human Life Sciences\HLS.HLSSitePage

Themes

Package contents	• Theme content • Updated Config.xml file
Packaging format	SharePoint feature files in the following folder structure delivered as a .zip file: <Component Name> - Config.XML - WSS - TEMPLATE - IMAGES - <Vendor Name> - <preview images> - THEMES - <Theme folder> - <Theme files>
Design guidelines	None
Configuration file	• Configuration file (Config.xml) must include theme ID, name, description and preview images. • TemplateID in Config.xml must be same as theme folder name in the folder structure. • Sample Config.xml file: <?xml version="1.0" encoding="utf-8" ?> <Install> <SPThemes> <Templates> <TemplateID>ObaCentral</TemplateID> <DisplayName>OBA Central</DisplayName> <Description>OBA Central Theme</Description> <Thumbnail>images/litware/obacentral.gif</Thumbnail> <Preview>images/litware/obacentral.gif</Preview> </Templates> </SPThemes> </Install>
Sample	None

SSRS Reports

Package contents	SSRS report (.rdl) file
Packaging format	.rdl file delivered as a .zip file
Supporting adapter(s)	• SQL Server 2005 database • Access 2007 database
Design guidelines	• .rdl file must contain embedded named data sources. • Data source names in the .rdl file must match the names of the LOB adapters used by the report.
Design guidelines to access supporting adapters as data source	• The embedded data source name for supporting adapters **must be internal name of the adapter**. • Sample .rdl file using SharePoint list Web service: <DataSources> <DataSource Name="**Statera.PlantFloorOpsDB**"> <rd:DataSourceID> eb031746-f22f-4244-b4b8-b2439e138f79 </rd:DataSourceID> <ConnectionProperties> <DataProvider>SQL</DataProvider> <ConnectString> Data Source=localhost;Initial Catalog=PlantFloorOps </ConnectString> <IntegratedSecurity>true</IntegratedSecurity> </ConnectionProperties> </DataSource> </DataSources>
Design guidelines to access SharePoint list or document library as data source	• The embedded data source name for current SharePoint list Web service **must be "SiteListService"**. • Sample .rdl file using SharePoint list Web service: <DataSources> <DataSource Name="**SiteListService**"> <rd:DataSourceID> 2b7c589b-8fe9-4cf1-9af3-e1524185b872 </rd:DataSourceID> <ConnectionProperties> <DataProvider>XML</DataProvider> <ConnectString> http://obadevwks:34738/ER/_vti_bin/lists.asmx </ConnectString> <IntegratedSecurity>true</IntegratedSecurity> </ConnectionProperties> </DataSource> </DataSources>
Sample report that access supporting adapter as data	[Installation Folder]\Seed Components\Plant Floor Operations\Statera.ProductionPerfReport

source	
Sample report that access SharePoint list as data source	[Installation Folder]\Seed Components\ Expense Reporting\ER.CostAnalysisReport

Excel Services Report (or) Dashboard

Package contents	• Excel workbook (.xlsx) file • (Optional) Config.xml file to add Excel Web Part to SharePoint pages
Packaging format	Delivered as a .zip file: • .xlsx file • (Optional) Config.xml
Supporting adapter(s)	• SQL Server 2005 database • SSAS database • .cub file
Design guidelines	• The Report Data Source connection must be embedded in the .xlsx file. • The connection name must match the name of the LOB adapter used by the report. • Example: The following is the serialized connection information from an .xlsx file. The use of the adapter name as the connection name and the connection string of the source SSAS database referenced by the connection are highlighted in bold font: <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <connections xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main"> <connection id="1" keepAlive="1" name="**CRM**" type="5" refreshedVersion="3" background="1" saveData="1"> <dbPr connection="**Provider=MSOLAP.3;Integrated Security=SSPI;** **Persist Security Info=True; Initial Catalog=CRMOLAP;** **Data Source=karserv;MDX Compatibility=1;Safety Options=2;** **MDX Missing Member Mode=Error**" command="CRM" commandType="1"/> <olapPr sendLocale="1" rowDrillCount="1000"/> </connection> </connections> • The Mashup provisioning engine will look a connection in the report .xlsx file with the related **adapter internal name** and update the connection string based on the location to which the corresponding adapter instance is provisioned.
Guidelines to display Excel Web part in the Web pages	• Configuration file (Config.xml) supports provisioning of Excel Web Part to the SharePoint content pages.

	• Sample Config.xml to include Web Part: <?xml version="1.0" encoding="utf-8" ?> <Install> <SitePages> <SitePage Url="default.aspx"> **<WebPart** **ReportUrl="Reports\Pipeline by Area.xlsx"** **Title="Pipeline by Area"** **ZoneIndex="2"** **Type="ExcelReport"** **ZoneId="Right" />** </SitePage> </SitePages> </Install>
Sample	[Installation Folder]\Seed Components\Human Life Sciences\HLS.PipelinebyAreaChart