

**HYPERION® SYSTEM™ 9 BI+™
APPLICATION BUILDER.NET™**

RELEASE 9.2

DEVELOPER'S GUIDE



Copyright 2004-2006 Hyperion Solutions Corporation.
All rights reserved.

“Hyperion,” the Hyperion logo, and Hyperion’s product names are trademarks of Hyperion. References to other companies and their products use trademarks owned by the respective companies and are for reference purpose only.

No portion hereof may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the recipient’s personal use, without the express written permission of Hyperion.

The information contained herein is subject to change without notice. Hyperion shall not be liable for errors contained herein or consequential damages in connection with the furnishing, performance, or use hereof.

Any Hyperion software described herein is licensed exclusively subject to the conditions set forth in the Hyperion license agreement.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the applicable Hyperion license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (Oct 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14, as applicable.

Hyperion Solutions Corporation
5450 Great America Parkway
Santa Clara, California 95054

Printed in the U.S.A.

Contents

Preface	v
Purpose	v
Audience	v
Document Structure	v
Where to Find Documentation	vi
Conventions	vi
Additional Support	vii
Education Services	vii
Consulting Services	viii
Technical Support	viii
Documentation Feedback	viii
CHAPTER 1 Getting Started with Application Builder.NET	9
Overview	10
Setting Up the Development Environment	10
CHAPTER 2 Advanced Programming with ASP.NET	13
Overview	14
Creating a Basic ASP.NET Application	14
Setting Up Dependencies	14
Creating a Web Form	14
CHAPTER 3 Programming with Base Classes	17
Overview	18
EssOperations	18
EssConnection	19
EssDomain	20
EssProperties	20
CHAPTER 4 Programming with Dialog Classes	23
Overview	24
SignOnDialog	24

MemberSelectionDialog	24
EssTreeview	25
CHAPTER 5 Programming with Adapter Controls	27
Overview	28
Data Server Component	28
Web Grid Adapters	29
Grid Adapter Generic	29
Grid Adapter OWC11	30
Web Metadata Adapters	31
Drop-Down List Adapter	31
List Box Adapter	32
Index	33

Preface

Welcome to the *Application Builder.NET Developer's Guide*. This preface discusses the following topics:

- [“Purpose” on page v](#)
- [“Audience” on page v](#)
- [“Document Structure” on page v](#)
- [“Where to Find Documentation” on page vi](#)
- [“Conventions” on page vi](#)
- [“Additional Support” on page vii](#)
- [“Documentation Feedback” on page viii](#)

Purpose

This guide provides information that you need to create applications programs using Application Builder.NET. It explains Application Builder.NET features and options and contains the concepts, processes, procedures, formats, tasks, and examples that you need to use the software.

Audience

This guide is for software developers and programmers who are responsible for creating applications programs that access and manipulate Hyperion Analytic Services data. This guide assumes that you are familiar with Hyperion Analytic Services and the .NET development environment.

Document Structure

This document contains the following information:

[Chapter 1, “Getting Started with Application Builder.NET,”](#) introduces Application Builder.NET. It describes how to create a basic application within the .NET environment that will access Hyperion Analytic Services data.

Chapter 2, “Advanced Programming with ASP.NET” provides examples of more advanced programming concepts, including programming for ASP.NET.

Chapter 3, “Programming with Base Classes,” describes the most commonly-used base classes and assemblies packaged with Application Builder.NET.

Chapter 4, “Programming with Dialog Classes,” describes how to extend your Application Builder.NET applications using sign-on and member selection dialog boxes.

Chapter 5, “Programming with Adapter Controls,” describes adapter controls packaged with Application Builder.NET.

[Index](#) contains a list of Application Builder.NET terms and their page references.

Where to Find Documentation

All Application Builder.NET documentation is accessible from the following locations:

- The HTML Information Map is located at `\Hyperion\habnet\docs\habnet_info_map.htm`.
- Online help is available by clicking the Online Help link in the Information Map. Context-sensitive help for Application Builder.NET classes, properties, and methods is available within Microsoft Visual Studio .NET.
- The Hyperion Download Center can be accessed from the Hyperion Solutions Web site.

➤ To access documentation from the Hyperion Download Center:

- 1 Go to the **Hyperion Solutions Web site** and navigate to **Services > WorldWide Support > Download Center**.


Note: Your Login ID for the Hyperion Download Center is your e-mail address. The Login ID and Password required for the Hyperion Download Center are different from the Login ID and Password required for Hyperion Support Online through Hyperion.com. If you are not sure whether you have a Hyperion Download Center account, follow the on-screen instructions.

- 2 In the **Login ID** and **Password** text boxes, enter your e-mail address and password.
- 3 In the **Language** list box, select the appropriate language and click **Login**.
- 4 If you are a member on multiple Hyperion Solutions Download Center accounts, select the account that you want to use for the current session.
- 5 To access documentation online, from the Product List, select the appropriate product and follow the on-screen instructions.

Conventions

The following table shows the conventions that are used in this document:

Table i Conventions Used in This Document

Item	Meaning
	Arrows indicate the beginning of procedures consisting of sequential steps or one-step procedures.
Brackets []	In examples, brackets indicate that the enclosed elements are optional.
Bold	Bold in procedural steps highlights major interface elements.
CAPITAL LETTERS	Capital letters denote commands and various IDs. (Example: CLEARBLOCK command)
Ctrl + 0	Keystroke combinations shown with the plus sign (+) indicate that you should press the first key and hold it while you press the next key. Do not type the plus sign.
Example text	Courier font indicates that the example text is code or syntax.
<i>Courier italics</i>	Courier italic text indicates a variable field in command syntax. Substitute a value in place of the variable shown in Courier italics.
<i>ARBORPATH</i>	When you see the environment variable <i>ARBORPATH</i> in italics, substitute the value of <i>ARBORPATH</i> from your site.
<i>n, x</i>	Italic <i>n</i> stands for a variable number; italic <i>x</i> can stand for a variable number or an alphabet. These variables are sometimes found in formulas.
Ellipses (...)	Ellipsis points indicate that text has been omitted from an example.
Mouse orientation	This document provides examples and procedures using a right-handed mouse. If you use a left-handed mouse, adjust the procedures accordingly.
Menu options	Options in menus are shown in the following format. Substitute the appropriate option names in the placeholders, as indicated. Menu name > Menu command > Extended menu command For example: 1. Select File > Desktop > Accounts .

Additional Support

In addition to providing documentation and online help, Hyperion offers the following product information and support. For details on education, consulting, or support options, click the Services link at the Hyperion Solutions Web site.

Education Services

Hyperion offers instructor-led training, custom training, and e-Learning covering all Hyperion applications and technologies. Training is geared to administrators, end users, and information systems professionals.

Consulting Services

Experienced Hyperion consultants and partners implement software solutions tailored to clients' particular reporting, analysis, modeling, and planning requirements. Hyperion also offers specialized consulting packages, technical assessments, and integration solutions.

Technical Support

Hyperion provides enhanced telephone and electronic-based support to clients to resolve product issues quickly and accurately. This support is available for all Hyperion products at no additional cost to clients with current maintenance agreements.

Documentation Feedback

Hyperion strives to provide complete and accurate documentation. We value your opinions on this documentation and want to hear from you. Send us your comments by clicking the link for the Documentation Survey, which is located on the Information Map for your product.



Getting Started with Application Builder.NET

This chapter introduces Application Builder.NET. It describes how to create a basic application within the .NET environment that will access Analytic Services data.

In This Chapter	Overview	10
	Setting Up the Development Environment	10

Overview

Application Builder.NET provides a comprehensive set of OLAP-aware classes for data navigation, selection, reporting, and visualization to assist you in building custom analytical applications. These controls provide you with the tools to quickly create and deploy applications in stand-alone, client-server, Web, or distributed processing environments.

Application Builder.NET is a component of the Hyperion Business Intelligence Platform that provides an application development workbench for companies wanting to use the Microsoft .NET Framework to create tailored business performance management solutions. Application Builder.NET includes the following key features:

- .NET Framework compatibility
- Web Services-based architecture (SOAP)

Application Builder.NET provides the following functionality:

- Base classes. These classes contain basic OLAP functionality and encapsulate functionality necessary to connect to and perform operations on Analytic Services applications and databases.
- Dialog components. These classes encapsulate the functionality of Analytic Services dialog boxes, including the sign-on and member selection dialog boxes.
- Grid classes: These classes encapsulate functionality enabling data grid controls to display Analytic Services data.

Setting Up the Development Environment

To develop Application Builder.NET applications within Visual Studio for .NET, perform the following steps:

- Install and configure Microsoft Visual Studio for .NET.
- Install Application Builder.NET and note which directory contains the Application Builder.NET assemblies.
- If you are developing an ASP .NET (Active Server Pages) application, install or ensure access to an instance of Microsoft Internet Information Server (IIS).
- If you are developing applications using Microsoft Office Web Components (OWC11), you may need to add Microsoft Office Primary Interop Assemblies and install additional assemblies available from www.microsoft.com.
- Install or ensure access to an Analytic Services server and application.

This guide assumes that you will be using the Sample application and Basic database installed with Analytic Services.

- Ensure that the Application Builder.NET .WAR file has been applied to your Web application server. For information on how to deploy the .WAR file, see the *Application Builder.NET Installation Guide*.
- Ensure that Hyperion® System™ 9 BI+™ Analytic High Availability Services™ is running, and that your application server can access it.

Application Builder.NET requires that you install or have access to an application server. Apache Tomcat is installed with Application Builder.NET. This guide assumes that you are running Tomcat on your local machine.

Note: Hyperion provides Apache Tomcat on the installation media for convenience if you want to use it for your deployment. Hyperion does not own or maintain the Apache Tomcat application server and is not responsible for problems that you may encounter with its functionality. Hyperion, however, does fully support the use of Apache Tomcat in its products. In deployments where customers require high availability or failover, Hyperion recommends that you deploy a commercially supported application server where these capabilities are supported.



Advanced Programming with ASP.NET

This chapter guides you through developing more advanced Application Builder.NET applications. It introduces programming for ASP.NET and programming with other frequently-used Application Builder.NET classes.

In This Chapter	Overview	14
	Creating a Basic ASP.NET Application	14

Overview

Creating an application with Application Builder.NET for .NET provides the flexibility of creating a number of different types of applications in any .NET language. This section shows how to create an ASP.NET application and then add enhanced sign-on and display functionality to it.

Creating a Basic ASP.NET Application

Follow the procedures in this topic to create a .NET application to retrieve and manipulate data from an Analytic Services sample application and database.

- To set up a sample application:
 - 1 Launch Microsoft Visual Studio for .NET.
 - 2 Click **New Project**.
 - 3 Select the **Visual C# Projects** or **Visual Basic Projects** folder and the **Web Application** template.
 - 4 Optionally, rename the project file.
 - 5 Click **OK**.

Setting Up Dependencies

To use Application Builder.NET methods and properties, you must add references to the Application Builder.NET assemblies in your program.

- To add Application Builder.NET adapter components to the Toolbox:
 - 1 Right-click on the Toolbox and choose **Add/Remove Items**.

The Customize Toolbox dialog box opens.
 - 2 Click **Browse** and navigate to the directory containing the Application Builder.NET assemblies. The default directory is `c:\Hyperion\habnet\bin`.
 - 3 Select the `Hyperion.Objects.Essbase.Components.Web.dll` file and click **Open**.
 - 4 Click **OK**.

Creating a Web Form

- To create a web form for an ASP.NET application using Application Builder.NET:
 - 1 Launch Visual Studio .NET 2003 and start a new project.
 - 2 Choose ASP.NET Web Application and the programming language that you want to use.
 - 3 To add a Drop-Down List:

- a. Drag-and-drop a Drop-down List into the form in the designer window.
- b. Double-click the DataServer component from the Toolbox to add it to the component tray and select the DataServer in the component tray.
- c. Click the **Sign On** link in the designer verb collection area.
- d. Enter Essbase login credentials in the Sign-on dialog box and click **Sign-on**.
- e. Select the cube from the Select Cube dialog box and click **OK**.
- f. Double-click the DropDownListAdapter component from the Toolbox to add it to the component tray.
- g. Select the DropDownListAdapter component in the component tray.
- h. In the Properties window, select values for ControlToBind and DataServer properties.
- i. Click **SelectMembers** in the designer box collection and select the Members from the Member Selection dialog box.
- j. Double-click on the form to open the code window and add the following lines before the namespace declaration:
 - o If you are programming in C#:


```
using Hyperion.Objects.Essbase.Common.Grid.WebDataGrid;
```
 - o If you are programming in Visual Basic for .NET:


```
Imports Hyperion.Objects.Essbase.Common.Grid.WebDataGrid;
```
- k. Create the following variables in the variables declaration section of your form:
 - o If you are programming in C#:


```
private GridWebDataGrid EssbaseGrid= new GridWebDataGrid();
```
 - o If you are programming in Visual Basic for .NET:


```
private EssbaseGrid as new GridWebDataGrid();
```

The EssOperations class encapsulates all information necessary to execute operations on an Analytic Services application. Declare the EssOperations class at the class level or as a public variable; alternatively, declare the other objects within your form.
- l. Add code to the Form1_Load method that will enable your application to connect to Analytic Services and to perform operations on a cube.
 - o If you are programming in C#:


```
///plug the physical grid into the essbase grid object
EssbaseGrid.DataGrid = this.DataGrid1;
this.dataServer1.EssOperations.Grid = EssbaseGrid;
```
 - o If you are programming in Visual Basic for .NET:


```
EssbaseGrid.DataGrid = Me.DataGrid1
Me.DataServer1.EssOperations.Grid = EssbaseGrid
```

4 To add a List Box:

- a. Drag-and-drop a List Box and a Button into the form in the designer window.

- b. Double-click the DataServer component from the Toolbox to add it to the component tray and select the DataServer in the component tray.
- c. Click the **Sign On** link in the designer verb collection area.
- d. Enter Essbase login credentials in the Sign-on dialog box and click **Sign-on**.
- e. Select the cube from the Select Cube dialog box and click **OK**.
- f. Double-click the ListBoxAdapter component from the Toolbox to add it to the component tray and select the ListBoxAdapter component in the component tray.
- g. In the Properties window, select values for ControlToBind, DataServer and Button properties.
- h. Click **SelectMembers** in the designer box collection and select the Members from the Member Selection dialog box.
- i. Double-click on the form to open the code window and add the following lines before the namespace declaration:
 - o If you are programming in C#:


```
using Hyperion.Objects.Essbase.Common.Grid.WebDataGrid;
```
 - o If you are programming in Visual Basic for .NET:


```
Imports Hyperion.Objects.Essbase.Common.Grid.WebDataGrid;
```
- j. Create the following variables in the variables declaration section of your form:
 - o If you are programming in C#:


```
private GridWebDataGrid EssbaseGrid= new GridWebDataGrid();
```
 - o If you are programming in Visual Basic for .NET:


```
private EssbaseGrid as new GridWebDataGrid();
```

The EssOperations class encapsulates all information necessary to execute operations on an Analytic Services application. Declare the EssOperations class at the class level or as a public variable; alternatively, declare the other objects within your form.
- k. Add code to the Form1_Load method that will enable your application to connect to Analytic Services and to perform operations on a cube.
 - o If you are programming in C#:


```
///plug the physical grid into the essbase grid object
EssbaseGrid.DataGrid = this.DataGrid1;
this.dataServer1.EssOperations.Grid = EssbaseGrid;
```
 - o If you are programming in Visual Basic for .NET:


```
EssbaseGrid.DataGrid = Me.DataGrid1
Me.DataServer1.EssOperations.Grid = EssbaseGrid
```

3

Programming with Base Classes

This chapter explains the most frequently-used Application Builder.NET base classes. It discusses many of the most important properties and methods within the classes.

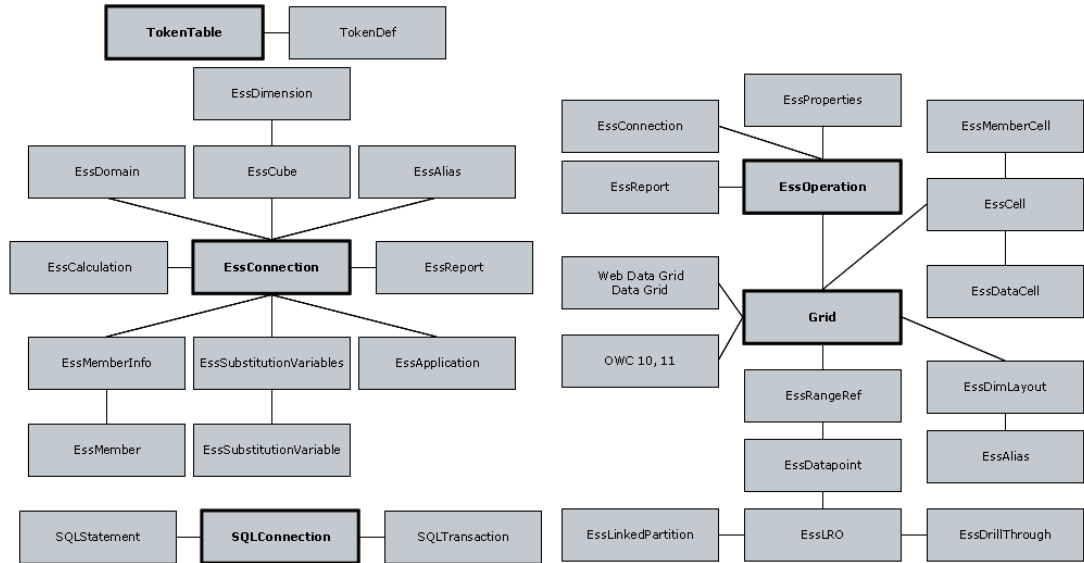
In This Chapter	Overview	18
	EssOperations	18
	EssConnection	19
	EssDomain	20
	EssProperties	20

Overview

Application Builder.NET includes many object classes encapsulating the functionality of the Analytic Services API. These object classes, plus the dialog classes, enable you to create applications that access Analytic Services functionality with fewer lines of code than in previous releases of Application Builder.NET, or in the Analytic Services API.

Figure 1 shows base Application Builder.NET classes and how they relate to one another.

Figure 1 Base Object Classes



The topics in this section describe the most commonly-used base classes. For more information about all base classes, refer to the online help and the Visual Studio Object Explorer.

EssOperations

The EssOperations class encapsulates the functionality of common Analytic Services operations, including Pivot, Zoom In, Zoom Out, Keep Only, Lock Cells, and Unlock Cells. The EssOperations class is contained in the Hyperion.Objects.Essbase.Common.Grid assembly.

Some of the key properties and methods of this class are listed in the following table:

Table 2 Key Properties and Methods of EssOperations

Property or Method	Description
EssConnection property	Sets or returns the essConnection object used by the current essOperations object to encapsulate connection information for Analytic Server databases and applications currently being used.
EssProperties property	Sets or returns the essProperties object used by the current essOperations object to specify properties for Analytic Services operations.
EssReports property	Sets or returns the essReport object used by the current essOperations object for generating reports.
Grid property	Sets or returns the Grid object (which has implemented IEssGrid) used by the current EssOperations object.
KeepOnly method	Performs a KeepOnly operation on the active cell range in the Grid object associated with the current essOperations object.
Pivot method	Performs a Pivot operation on the currently active cell in the Grid object associated with the current essOperations object.
Retrieve method	Performs a Retrieve operation to retrieve all data from the connected Analytic Server database corresponding to the contents of the Grid object associated with the current essOperations object.
ZoomIn method	Performs a ZoomIn operation to zoom-in on data contained in the Grid object associated with this essOperations object.

EssConnection

The EssConnection class encapsulates the information and actions used to connect to an Analytic Services application and database via Analytic High Availability Services. The EssConnection class is contained in the Hyperion.Objects.Essbase.Common assembly.

Some of the key properties and methods of this class are listed in the following table:

Table 3 Key Properties and Methods of EssConnection

Property or Method	Description
Application property	Sets or returns the name of the Analytic Services application.
Connected method	Returns TRUE if the application is connected to an Analytic Services database; otherwise FALSE.
CubeViewName property	Sets or returns the server cube that will be used.
Database property	Sets or returns the name of the Analytic Services database.
OLAPServer property	Sets or returns the Analytic Server name.
EssApplication property	Sets or returns the EssApplication object associated with the current connection.

Table 3 Key Properties and Methods of EssConnection

Property or Method	Description
EssDomain property	Sets or returns the EssDomain object associated with the current connection.
Connect method	Connects to the Analytic Services database using the values of the Application, Database, and EssDomain properties.

EssDomain

The EssDomain class encapsulates the information and actions used to sign on to the Analytic High Availability Services and access the server, application, and database hierarchy. The EssDomain class is contained in the Hyperion.Objects.Essbase.Common assembly.

The following table lists some of the key properties and methods of the EssDomain class.

Table 4 Key Properties and Methods of EssDomain

Property or Method	Description
EDSServer property	Sets or returns the Analytic High Availability Services server network address.
Domain property	Sets or returns the Analytic High Availability Services domain name.
EncryptedLogin property	Sets or returns the flag that determines whether the connection username and password are encrypted before being sent to the server.
Password property	Sets or returns the login password.
Username property	Sets or returns the username used to log onto Analytic Services.
SignOn method	Signs on to Analytic Services.

EssProperties

The EssProperties class encapsulates properties associated with the current Analytic Services session. The EssProperties class is contained in the Hyperion.Objects.Essbase.Common.Grid assembly.

The following table lists some of the key properties of the EssProperties class.

Table 5 Key Properties of essProperties

Property or Method	Description
DrillLevel property	Sets or returns the default drill level to be applied to the ZoomIn and ConditionalZoomIn methods of an essOperations object.
HybridAnalysisEnabled property	Sets or returns a flag that indicates if Hybrid Analysis is to be enabled for any Analytic Services operation.
IndentStyle property	Sets or returns the indentation style to be applied to data obtained from the Analytic Services server when it is added to the Grid object associated with an essOperations object.

Table 5 Key Properties of essProperties

Property or Method	Description
MissingTextString property	Sets or returns the Analytic Services missing text string to use on the Grid object associated with an essOperations object in data retrieved from an Analytic Services database.
SendZerosAsMissing property	Sets or returns TRUE if the Grid object associated with an essOperations object should treat data that is zero as missing data in retrieves from an Analytic Services database; otherwise FALSE.
SuppressMissing property	Sets or returns TRUE if the Grid object associated with an essOperations object should suppress the retrieval of rows from an Essbase server when the rows contain all #Missing values; otherwise FALSE.
UpdateMode property	Sets or returns TRUE if an essOperations object automatically locks the corresponding database area when it retrieves data; otherwise FALSE.



Programming with Dialog Classes

This chapter describes the most commonly used dialog classes in Application Builder.NET.

In This Chapter	Overview	24
	SignOnDialog	24
	MemberSelectionDialog	24
	EssTreeView	25

Overview

Application Builder.NET dialog classes encapsulate the functionality of Analytic Services dialog boxes, including the sign-on and member selection dialog boxes. Dialog boxes can be displayed within an application by using the `ShowDialog()` method of the specific class.

In order to add a dialog box component to an application, add it to the Toolbox and double-click the component. This will add the component to the component tray.

SignOnDialog

The `SignOnDialog` class encapsulates the functionality necessary to display the Sign On dialog box. It uses an `EssConnection` object to connect to a Analytic Services application.

Some of the more useful properties and methods of this class are summarized in the table below:

Table 6 Selected Properties and Methods of the `SignOnDialog` class

Property or Method	Description
<code>EssConnection</code> property	Sets or returns the name of the associated <code>EssConnection</code> object used to connect to an Analytic Services database.
<code>ShowSelectCubeDialog</code> property	Sets or returns the flag that indicates whether the <code>SelectCubeDialog</code> should be displayed automatically when the user clicks the Sign On button and successfully signs on.
<code>ShowDialog</code> method	Displays the Sign On dialog .

MemberSelectionDialog

The `MemberSelectionDialog` class encapsulates the functionality of the Analytic Services Member Select dialog box. You can add this functionality to your .NET application by using this class in your programs. The `MemberSelectionDialog` class is part of the `Hyperion.Objects.Essbase.Windows.Forms` assembly.

Some of the more useful properties and methods of this class are summarized in the following table:

Table 7 Selected Properties and Methods of the `MemberSelectionDialog` class

Property or Method	Description
<code>MaxGenExpandable</code> property	Sets or returns the maximum generation number expandable in the Members tree.
<code>AllowChangeDimension</code> property	Sets or returns the flag that enables or disables the Dimension combo box.
<code>AllowLevelNames</code> property	Sets or returns the flag that enables or disables the "By Level Name" View Option button.
<code>DefaultDimension</code> property	Sets or returns the name of the dimension represented in the Members tree when the Member Selection dialog is displayed.

Table 7 Selected Properties and Methods of the MemberSelectionDialog class

Property or Method	Description
ListCount property	Returns the number of members selected from the last call to GenerateMemberList.
GetMemberList method	Generates and returns a collection of members using the selection rules defined for the dimension.
GenerateMemberList method	Generates an internal list of members using the selection rules defined for the dimension

EssTreeView

The EssTreeView class encapsulates information and actions necessary to display Analytic Services members in a tree view control.

Some of the more useful properties and methods of this class are summarized in the following table:

Table 8 Selected Properties and Methods of the EssTreeView class

Property or Method	Description
EssConnection property	Sets or returns the name of the associated EssConnection object used to connect to an Analytic Services database.
UseAliases property	Sets or returns the flag that indicates if member names or Aliases are displayed in the treeview.
PreFillDescendants property	Sets or returns the flag that signifies if the descendants should be pre-filled in the treeview.
AddMembers method	Adds a list of members as root nodes to the treeview.
Add method	Adds a member as a root node to the treeview. The PreFillDescendants and MaxGenExpandable properties determine the members represented in the treeview.

5

Programming with Adapter Controls

This chapter explains the Application Builder.NET drag-and-drop controls: data server component, web grid adapter, and metadata adapters. It also defines the key properties and methods within each of the controls.

In This Chapter	Overview	28
	Data Server Component	28
	Web Grid Adapters	29
	Web Metadata Adapters	31

Overview

Application Builder.NET uses the DataServer component to maintain the current state of an application and performs all communication with an Analytic Services server. Web grid adapters and Web metadata adapters contain metadata and data. The adapters synchronize their metadata and data with the Data Server and populate the associated control with the information.

To add the DataServer component, Web grid adapters, and Web metadata adapters to an application, add them to the Toolbox and double-click or drag-and-drop the component into the component tray.

Data Server Component

The Data Server component for the web (ASP.NET) is a central component in the development of .NET applications. It consists of all the .NET properties, such as the sign-on dialog, server connection, and so on.

Some of the key properties and methods of this class are listed in the following table:

Table 9 Key Properties and Methods of DataServer Component

Property or Method	Description
Name	Sets or returns the name of the data server component.
AutoConnect property	Set to TRUE to connect if not currently connected.
ClearPasswordOnRun property	Clears the password required to connect to the data server. By default, the flag is set to TRUE.
EssConnection property	Sets or returns the essConnection object used by the current DataServer object to encapsulate connection information for Analytic Server databases and applications currently being used.
EssDatapoint property	Returns the EssDataPoint object defined.
EssOperations property	Sets the Analytic Services operations, such as retrieving, zooming, pivoting, and updating data.
EssProperties property	Sets or returns the essProperties object used by the current dataServer object to specify properties for Analytic Services operations.
ID	Sets or returns the ID for this DataServer object.
OlapMode property	Sets to TRUE to connect to the Analytic Server in the OLAP mode.
ReportLocal property	Sets to TRUE if local reporting is required; otherwise, FALSE.
ReportObject property	Sets the reporting object.
ShowLoginForm property	Sets to TRUE to display the login form; otherwise, FALSE.
Caller	Set to the object making a request to the Data Server to execute an operation (e.g. Retrieve). By default this is set to the Data Server control itself. Should be used only by the controls attaching themselves to the Data Server.

Table 9 Key Properties and Methods of DataServer Component

Property or Method	Description
OlapState property	Sets the OLAP state.
KeepOnly method	Performs a WebServices KeepOnly Operation on the active cell range in the Grid object associated with this EssOperations object.
LockCells	Locks the selected cells in the Analytic Server database.
Pivot	Overloaded. Pivots data in the attached grid object from the Analytic Server database defined in the EssConnection object, based on properties defined in the EssProperties object and the currently active cell in the attached grid object.
RemoveOnly	Performs a WebServices RemoveOnly Operation on the active cell range in the Grid object associated with this EssOperations object.
Report	Overloaded. Runs an Analytic Services report script represented in an EssReport object and places the results into a grid.
ReportFile	Overloaded. Runs an Analytic Services report script represented in an EssReport object and places the results into a file.
Retrieve	Overloaded. Retrieves data into the attached grid object from the Analytic Server database defined in the EssConnection object, based on properties defined in the EssProperties object.
RetrieveForState	Performs a Retrieve for State operation (the internal grid is changed to match the Olap State).
Update	Overloaded. Performs a WebServices Update Operation to update the contents of the connected Analytic Server database with values corresponding to the contents of the Grid object associated with this EssOperations object.
ZoomIn	Overloaded. Zooms in on data in the attached grid object from the Analytic Server database defined in the EssConnection object, based on properties defined in the EssProperties object.
ZoomOut	Zooms out on data in the attached grid object from the Analytic Server database defined in the EssConnection object based, on properties defined in the EssProperties object.

Web Grid Adapters

The Web grid adapters enables you to select the data server controls and exchange data from different data grid types and controls the information exchanged between them.

Grid Adapter Generic

The GridAdapterGeneric class is used to pass information between grids that have a data source property supporting a dataset.

Some of the key properties and methods of this class are listed in the following table:

Table 10 Key Properties and Methods of DropDownListAdapter

Property or Method	Description
Name	Sets or returns the name of the grid adapter component.
DataServer property	Sets or returns the name of the Data Server control used by the current control.
InitializeGrid method	Initialized to retrieve grid information into a dataset.

Grid Adapter OWC11

The Office Web Components (OWC) 11 class is used to pass information between grids that have a data source property supporting a dataset.

Some of the key properties and methods of this class are listed in the following table:

Table 11 Key Properties and Methods of OWC 11 Grid Adapter Component

Property or Method	Description
InitializeSpreadSheet	Initializes the internal DataGrid to hold the Metadata information.
ClientOWC11SpreadsheetName	Gets or Sets the name of the client side OWC11 object. This name is used to populate the spreadsheet with the retrieved data. Case sensitive and must be preceded with the form name.
DataServer	The Data Server control to which this adapter is attached.
UseStyles	Sets or returns True if the grid control displays the font styles for linked objects, linked partitions, and Integration Server Drill-Through. Set the property to True to use font styles.
StyleFontLinkedURL	Sets or returns the font assigned to grid cells that contain data that is linked to a URL. Set the 'UseStyles' to true as a Pre requisite.
Retrieve	Performs a basic Retrieve Operation.
ZoomIn	Performs a Zoom In Operation. Displays the Zoomed in data for the selected Dimension in the grid.
ZoomOut	Performs a Zoom Out Operation. Displays the Zoomed out data for the selected Dimension in the grid.
KeepOnly	Performs a KeepOnly Operation. Keeps the selected Dimensions and removes all the others from the grid.
RemoveOnly	Performs a RemoveOnly Operation. Removes only the selected dimensions from the grid.
LockCells	Performs a LockCells operation on the current active cell range.
UnlockCells	Performs an UnlockCells operation on the current active cell range.

Table 11 Key Properties and Methods of OWC 11 Grid Adapter Component

Property or Method	Description
Pivot	Executes a Pivot Operation on the currently active cell. You use the pivot method to change the orientation of data like Move a row group to a column group, Move a column group to a row group, Transpose the order of row groups and Transpose the order of column groups.
Update	Performs an Update Operation. You can change the value in the grid cells and update them in the database.

Web Metadata Adapters

The Web metadata adapter components enables you to exchange metadata between Data Servers and the list.

Drop-Down List Adapter

The DropDownListAdapter class is used to access the Data Server controls using the drop-down list. The DropDownListAdapter class encapsulates the information and actions used to add and remove Analytic Services members to or from a drop-down list.

Some of the key properties and methods of this class are listed in the following table:

Table 12 Key Properties and Methods of DropDownListAdapter

Property or Method	Description
ControlToBind property	Sets the ID of Control to Bind.
DataServer property	Sets or returns the name of the Data Server control used by the current control.
Dimension property	Sets or returns the name of the database dimension.
DisplayMode property	Sets or returns the type of members in the list.
EventEnabled property	Set to TRUE if the SelectedIndexChanged event is handled by DropDownListAdapter; otherwise, FALSE.
Layout property	Sets or returns the layout of the dimension described by the DimState object.
MaxItem property	Sets or returns the maximum number of items to be displayed in the list.
Add method	Adds a member to the member list collection.
Clear method	Clears the data from the member list collection.
Contains method	Determines whether the specified member is in the member list collection.
CopyTo	Copies the elements of the collection to an array, starting at a particular array index.
IndexOf	Returns the index of the member.
Insert	Inserts a member into the collection at the specified index.

Table 12 Key Properties and Methods of DropDownListAdapter

Property or Method	Description
Remove	Removes a member from the member list collection.
RemoveAt	Removes the member at the specified index.

List Box Adapter

The ListBoxAdapter class is used to access the Data Server controls using the list box. The ListBoxAdapter class encapsulates the information and actions used to add and remove Analytic Services members to or from a list box.

Some of the key properties and methods of this class are listed in the following table:

Table 13 Key Properties and Methods of ListBoxAdapter

Property or Method	Description
ControlToBind property	Sets the ID of Control to Bind.
DataServer property	Sets or returns the name of the Data Server control used by the current control.
Button	Button to be bound to the adapter to submit the page after the selection is made to the list box.
Dimension property	Sets or returns the name of the database dimension.
DisplayMode property	Sets or returns the type of members in the list.
EventEnabled property	Set to TRUE if the SelectedIndexChanged event is handled by ListBoxAdapter; otherwise, FALSE.
Layout property	Sets or returns the layout of the dimension described by the DimState object.
MaxItem property	Sets or returns the maximum number of items to be displayed in the list box.
Add method	Adds a member to the list box collection.
Clear method	Clears the data from the list box collection.
Contains method	Determines whether the specified member is in the list box collection.
CopyTo	Copies the elements of the collection to an array, starting at a particular array index.
IndexOf	Returns the index of the member.
Insert	Inserts a member into the collection at the specified index.
Remove	Removes a member from the list box.
RemoveAt	Removes the member at the specified index.

Index

A

aliases, 25
Application property, 19
ASP.NET
 creating a basic application, 14
audience for this guide, v

B

Base classes
 EssConnection, 19
 EssDomain, 20
 EssOperations, 18
 EssProperties, 20
 Overview, 17

C

Connect method, 20
Connected method, 19
consulting services, viii
CubeViewName property, 19

D

Database property, 19
Dialog boxes, creating
 member selection, 24
 sign-on, 24
documents
 conventions used, vi
 structure of, v
documents, accessing
 Hyperion Download Center, vi
 Hyperion Solutions Web site, vi
 Information Map, vi

 online help, vi
Domain property, 20

E

EDSServer property, 20
education services, vii
EncryptedLogin property, 20
EssApplication property, 19
EssConnection class
 Application property, 19
 Connect method, 20
 Connected method, 19
 CubeViewName property, 19
 Database property, 19
 EssApplication property, 19
 EssDomain property, 20
 OLAPServer property, 19
EssConnection property, 24 to 25
EssDomain class
 Domain property, 20
 EDSServer property, 20
 EncryptedLogin property, 20
 Password property, 20
 SignOn method, 20
 Username property, 20
EssDomain property, 20
EssOperations class
 about, 18
 EssConnection property, 19
 EssProperties property, 19
 EssReports property, 19
 Grid property, 19
 KeepOnly method, 19
 Pivot method, 19

Retrieve method, [19](#)

ZoomIn method, [19](#)

EssTreeView class

Add method, [25](#)

AddMembers method, [25](#)

EssConnection property, [25](#)

overview, [25](#)

PreFillDescendants property, [25](#)

UseAliases property, [25](#)

F

Forms

creating a Web grid form, [14](#)

H

Hyperion Consulting Services, [viii](#)

Hyperion Education Services, [vii](#)

Hyperion Essbase

KeepOnly operation, [19](#)

Pivot operation, [19](#)

Retrieve operation, [19](#)

ZoomIn operation, [19](#)

Hyperion Solutions Web Site

accessing documents, [vi](#)

Hyperion Technical Support, [viii](#)

Hyperion.Objects.Essbase.Windows.Forms assembly, [24](#)

K

KeepOnly operation, [19](#)

M

members, [25](#)

MemberSelectionDialog class

AllowChangeDimension property, [24](#)

AllowLevelNames property, [24](#)

DefaultDimension property, [24](#)

GenerateMemberList method, [25](#)

GetMemberList method, [25](#)

ListCount property, [25](#)

MaxGenExpandable property, [24](#)

overview, [24](#)

O

OLAPServer property, [19](#)

P

Password property, [20](#)

Pivot operation, [19](#)

prerequisites for using this guide, [v](#)

R

Retrieve operation, [19](#)

S

SignOn method, [20](#)

SignOnDialog class

EssConnection property, [24](#)

ShowDialog method, [24](#)

ShowSelectCubeDialog property, [24](#)

T

technical support, [viii](#)

U

Username property, [20](#)

W

Web form, creating, [14](#)

Z

ZoomIn operation, [19](#)