Insight
Americas 07

BUSINESS OBJECTS WEB SERVICES INTEGRATION

Michael Brown, Devon Energy
Scot Nesom, Devon Energy

COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

AGENDA

1. Review business requirements
2. How to set up the Web.Config
3. Authenticating and connecting to the Business Objects Web Services
4. Consuming the data from the Web Services
5. Building the tree view navigation
6. Conclusion
7. Q&A

COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

BUSINESS REQUIREMENTS

- The objective of our implementation of this web service, was to allow multiple applications already embedded with our corporate environment the ability to leverage the business objects reporting platform.

- Project/Business specific applications delivered through various user interfaces both purchased and built.
  - Web applications
  - Windows applications
  - Microsoft Office Add-in’s

COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

BI&T OBJECTIVES

(Business Information & Technology)

- Build an internal web service application that is extendable while seamlessly integrating the enterprise reporting platform within the business context or job function.
  - Our deployment strategy is completely project based.
  - Core foundations: .NET, Business Objects
  - Preferred delivery platform: web applications, web services, Server and User controls

COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

BI&T API Review

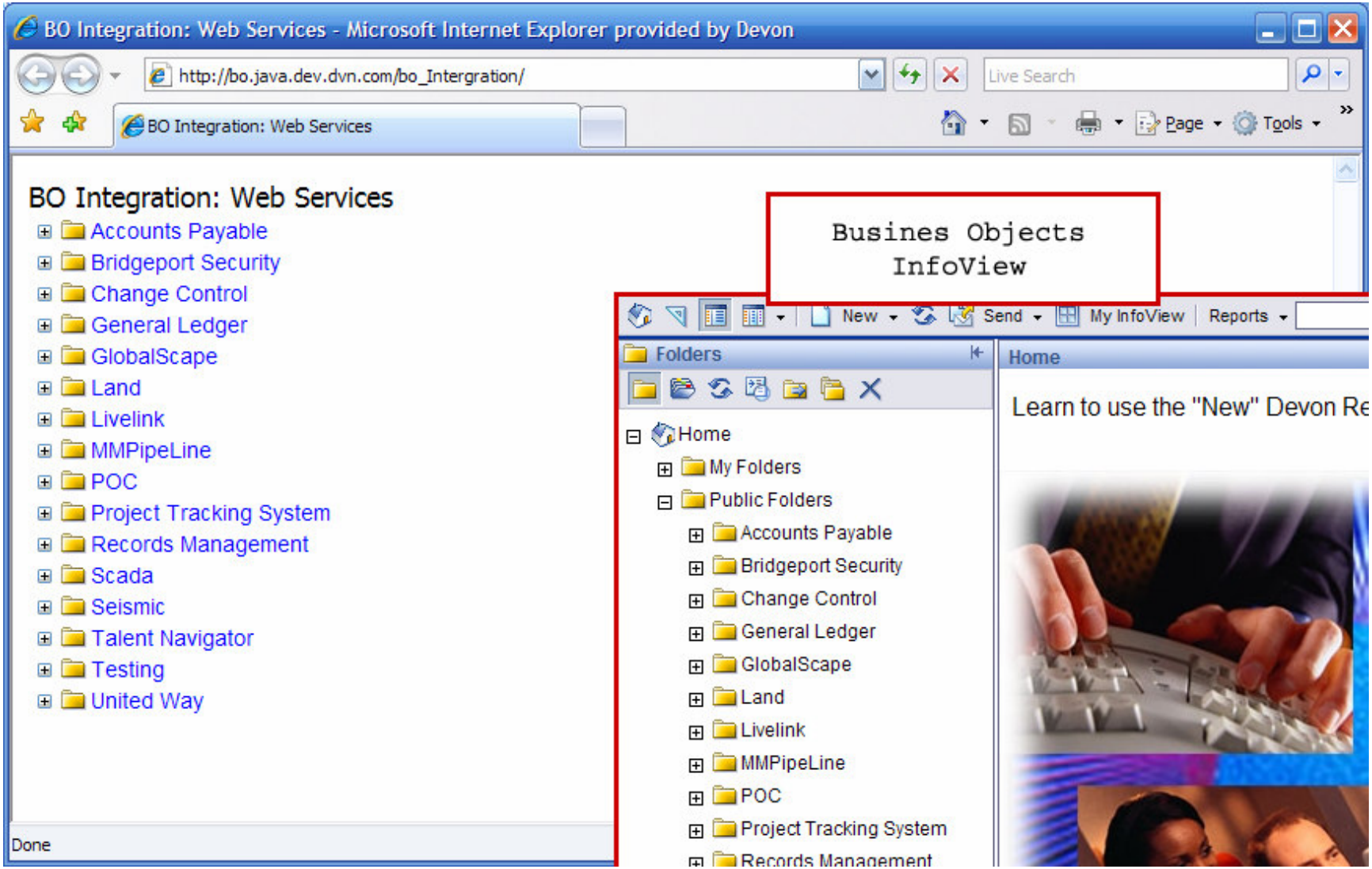
(Business Information & Technology)

Our approach in building this web service application was a simple one.

- Keep with the Service Oriented Architecture model
- Leverage any existing public API that will keep our risk of code duplication and code maintenance low.
- Review and implementation of the Business Objects published API made the most sense. It provided.
  - Accessibility
  - Extendable
- Ease of implementation while keeping our project deployment strategy approach the same.

COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

Tree View Navigation



COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

WEB.CONFIG SETTINGS

COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

WEB.CONFIG SETTINGS

- Allows the Business Objects Web Services to use current logged on user for authentication.

```
<system.web>
   <authentication mode="Windows"/>
   <identity impersonate="true"/>
</system.web>
```


COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

WEB.CONFIG SETTINGS

- AppSettings
  - These settings are used to retrieve the report catalog information.

```
<add key="CMS_SERVER"
      value="okcmsdap20.CORP.DVN.COM"/>
<add key="ROOT_LEVEL" value="23"/>
```

  - These settings are used to build the tree navigation control.

```
<add key="REPORT_REFRESH" value="N"/>

<add key="EXCLUDE_TYPES" value="analysis,
      analytic,afdashboardpage,objectpackage,
      fullclient"/>
```


COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

WEB.CONFIG SETTINGS

- AppSettings (Cont.)
  - These settings are used to build the tree navigation control.

```
<add key="REPORT_NAV_URL"
      value="http://bo.java.dev.dvn.com/
      businessobjects/enterprise115/
      desktoplaunch/opendoc/
      openDocument.jsp?"/>

<add key="REPORT_NAV_TARGET" value="_blank"/>
```


COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

WEB.CONFIG SETTINGS

- Web References
  - Session Web Service

```
<add key="bo.java.services.session"
    value="http://bo.java.dev.dvn.com/dswsbobje/
    services/session"/>
```

  - BICatalog Web Service

```
<add key="bo.java.services.bicatalog"
    value="http://bo.java.dev.dvn.com/dswsbobje/
    services/bicatalog"/>
```


COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

AUTHENTICATION AND CONNECTION TO WEB SERVICE

COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

AUTHENTICATION AND CONNECTION TO WEB SERVICE

- Business Objects Enterprise Session Web Service provides access to the user's details, and objects that are subject to the rights of the user.
- Before you create a Web Service Session you need to create a token using the SessionMgr.Logon.
- Next you create a Web Services Session with the token to retrieve the SessionID of which you will use to get the catalog data from the BICatalog Web Service.

COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

AUTHENTICATION AND CONNECTION TO WEB SERVICE

- Global.asax
  - Session_Start
  - Add References: CyrstalDecisions.Enterprise.Framework

```
using (CrystalDecisions.Enterprise.SessionMgr sessionMgr =
       new CrystalDecisions.Enterprise.SessionMgr())
{
   CrystalDecisions.Enterprise.EnterpriseSession
       enterpriseSession =
       sessionMgr.Logon(string.Empty, string.Empty,
       ConfigurationManager.AppSettings["CMS_SERVER"],
       "secWinAD");
```


COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

AUTHENTICATION AND CONNECTION TO WEB SERVICE

- Global.asax (Cont.)

```
bo.java.services.session.Session boSession =
     new bo.java.services.session.Session();

boSession.UseDefaultCredentials = true;

bo.java.services.session.SessionInfo boSessionInfo =
     boSession.loginWithToken(enterpriseSession.
     LogonTokenMgr.DefaultToken, "en_US",
     enterpriseSession.TimeZone.ToString());

Session["boSessionID"] = boSessionInfo.SessionID;

} //end of the using
```


COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

CONSUMING DATA FROM BICATALOG WEB SERVICE

COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

CONSUMING DATA FROM BICATALOG WEB SERVICE

- Page Load

```
boCatalog.Credentials =
   System.Net.CredentialCache.DefaultCredentials;

if (!Page.IsPostBack)
{
 try{
   boCatalogObject = boCatalog.getCatalog(
       Session["boSessionID"].ToString(),
       objUID, levelDepth, mySort, objType,
       resourceRightNames, objectPropertyNames,
       instanceRetrievalType);
```


COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

getCatalog Parameters

Source:
http://devlibrary.businessobjects.com/BusinessObjectsXIR2/en/en/WS_SDK/wssdk_consumer/doc/wssdk_net_consumer_apiRef/html/wslrfBusinessObjectsDSWSBICatalogBICatalogGetCatalogTopic.htm

- *objectUID* either from Category or Folder or Document. To get the root level, put an empty or null string.
- *depth* Number of level you want to retrieve. 0 is the current level, 1 is current plus child level, and -1 you get all levels.
- *sort* The return object is sorted with this sort list. First Sort is the first item of the table. No sort is required

COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

getCatalog Parameters (cont.)

- *bicatalogObjectType* This is the bicatalogObjectType that you want to retrieve. Null retrieves all object types. If you pass a non-handled type, the GetCatalog method will returned null.

- *resourceRightNames* List of Resource Right names you want to retrieve. If you pass a non-handled Resource Right name, the GetCatalog method will ignore it. Pass null and no Resource Rights are returned.

- *objectPropertyNames* List of Object Property names you want to retrieve. If you pass a non-handled Object Property name, the GetCatalog method will ignore it.

- *instanceRetrievalType* Indicate the Document instance type to filter on. The filter only impacts Document object, and not Folder or Category.

COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

CONSUMING DATA FROM BICATALOG WEB SERVICE

- Page Load (Cont.)

```
    if (boCatalog != null) {
        RecursivePopulateFolders(boCatalogObject,
        this.TreeCtrl.Nodes);
    } else {
        Response.Write("No items found catalog.");
    }

} catch (Exception ex) {
        Response.Write("Error occured: <br/>" +
        ex.ToString());
}
} //end Page.IsPostBack
```


COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

BUILDING THE TREE VIEW NAVIGATION CONTROL

COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

BUILDING THE TREE VIEW NAVIGATION CONTROL

- ASPX Page
  - Inserted the ASP Tree View Control
  - Set the ExpandDepth to zero
  - Set the OnTreeNodeExpanded Event

```
<asp:TreeView
   ID="TreeCtrl"
   runat="server"
   ExpandDepth="0"
   OnTreeNodeExpanded="TreeCtrl_TreeNodeExpanded">
</asp:TreeView>
```


COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

BUILDING THE TREE VIEW NAVIGATION CONTROL

- Code Behind Page
  Two Methods
  One Event Method

- RecursivePopulateFolders
  - Populates the specified tree node collection with all folders and documents retrieved from the web service GetCatalog method.

- Create Node
  - Builds the tree node for the tree node collection with the display text, navigation URL, and icon.

- TreeCtrl_TreeNodeExpanded
  - On tree node expand event retrieve the node data from Web Service and create child tree nodes.

COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

Create Node Method

```
private TreeNode CreateNode(string text,
   string path, string type, string tip, string id)
     {
          TreeNode n = new TreeNode();
          n.Text = text;
          n.ToolTip = tip;

          if (type == "Folder")
          {
               n.Value = id;
               n.ImageUrl = "images/folder.gif";
               n.SelectAction =
                    TreeNodeSelectAction.Expand;
```


COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

Create Node Method (cont.)

```
        TreeNode dummyNode = new TreeNode("*");
                n.ChildNodes.Add(dummyNode);

}   else   {          // if type is not a Folder

   n.NavigateUrl = path;

   n.Target   = ConfigurationManager.
       AppSettings["REPORT_NAV_TARGET"].ToString();

   switch (type.ToLower())
   {
       case "analytic":
       case "analysis":
```

 COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

Create Node Method (cont.)

```
            case "folder":
            case "webi":
            case "crystalreport":
                    n.ImageUrl = "images/" + type.ToLower()
                                 + ".gif";
                    n.Text = n.Text;
                    break;
            default:
                    n.ImageUrl = "images/unknown.gif";
                    n.Text = n.Text + "(" +
                                 type.ToLower() + ")";
                    break;
            }
        }
            return n;     //return the tree node
    }
```


COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

Recursive Populate Folders Method

```
private void RecursivePopulateFolders(
        BICatalogObject[] catalogObjects,
        TreeNodeCollection nodes)
{
   foreach (BICatalogObject catalogObject in
              catalogObjects)
{

   if (catalogObject is Document) {

        Document doc = catalogObject as Document;

        StringBuilder pathBuilder =
                    new StringBuilder();
```


COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

Recursive Populate Folders Method (cont.)

```
pathBuilder.Append(ConfigurationManager.
      AppSettings["REPORT_NAV_URL"].ToString());
pathBuilder.Append("&iDocID=" + doc.UID);
pathBuilder.Append("&sType=" + doc.FileType);
pathBuilder.Append("&sWindow=Same");
pathBuilder.Append("&sRefresh=" +
   ConfigurationManager.AppSettings["REPORT_REFRESH"]
   .ToString());

   nodes.Add(this.CreateNode(doc.Name,
       pathBuilder.ToString(),
       doc.ObjectType, doc.Description,
       doc.UID.ToString()));

   } else {   //not a document object type
```


COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

Recursive Populate Folders Method (cont.)

```
            nodes.Add(this.CreateNode(
                  catalogObject.Name,
                  string.Empty,
                  catalogObject.ObjectType,
                  catalogObject.Description,
                  catalogObject.UID.ToString()));
      }
      } //ends Foreach catalogObject
}
```


COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

TreeCtrl_TreeNodeExpanded Event Method

```
protected void TreeCtrl_TreeNodeExpanded(
        object sender, TreeNodeEventArgs e)
{
          TreeNode nodeSelected;
          nodeSelected = e.Node;

   if (nodeSelected.ChildNodes.Count > 0 &&
       nodeSelected.ChildNodes[0].Text == "*")
   {
       nodeSelected.ChildNodes.Clear();
```


COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

TreeCtrl_TreeNodeExpanded Event Method (cont.)

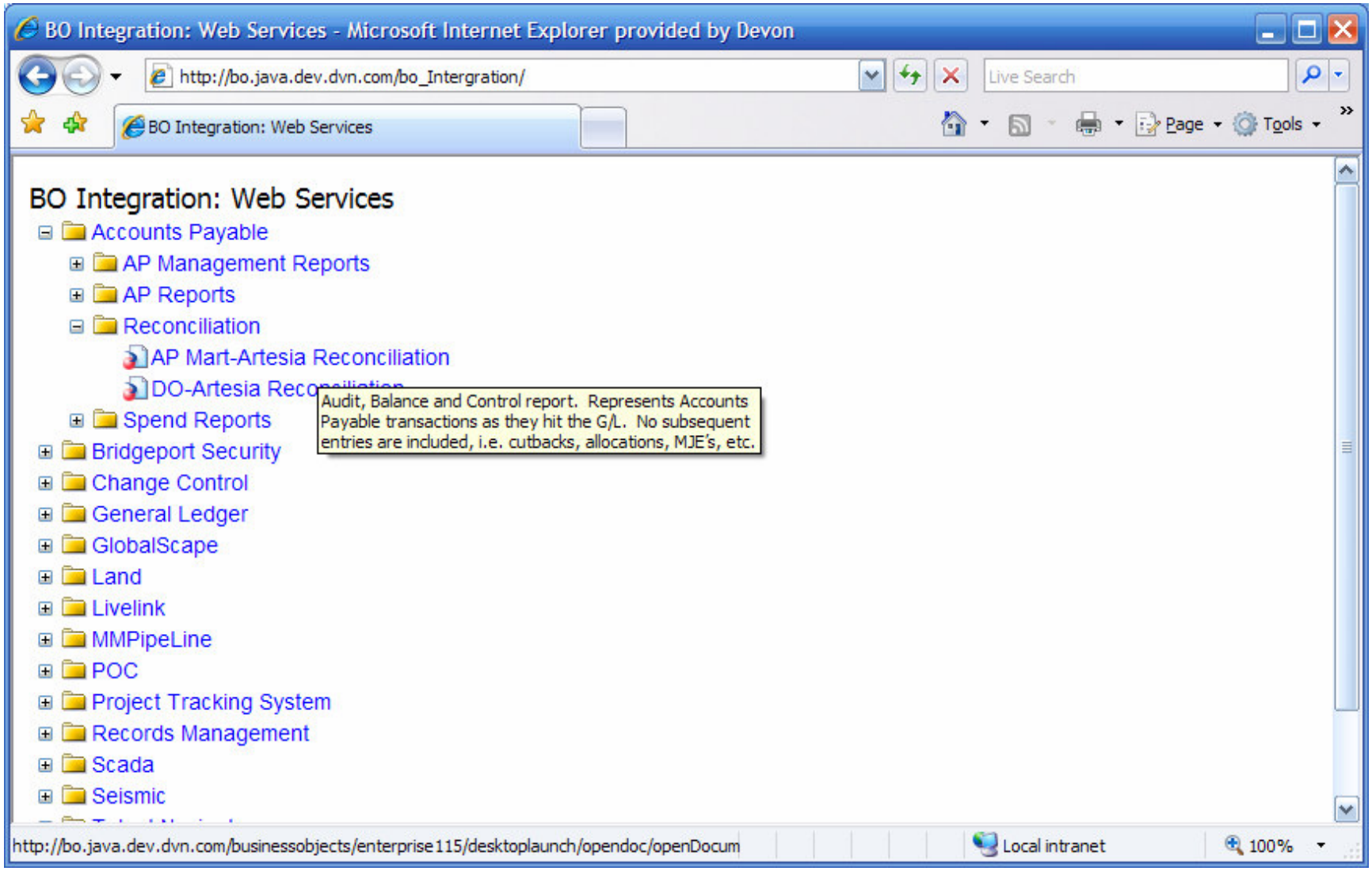
```
    boCatalogObject = boCatalog.getCatalog(
        Session["boSessionID"].ToString(),
        nodeSelected.Value.ToString(), levelDepth,
        mySort, objType, resourceRightNames,
        objectPropertyNames, instanceRetrievalType);

    if (boCatalogObject != null){
        TreeNodeCollection tnc =
              nodeSelected.ChildNodes;

        RecursivePopulateFolders(
              boCatalogObject, tnc);
    }
  }
}
```


COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

Tree View Navigation Expaned



 COPYRIGHT © 2007 BUSINESS OBJECTS S.A.

CONCLUSION

- Web.Config
  - Identity impersonate = true
- Authentication and connection to Web Service
  - Authentication and connect to the CMS Server to receive a token
  - Use token from logon to create the session with the Web Service to get the SessionID
- Consuming the data from the Web Services
  - GetCatalog method with the SessionID
- Building the tree view navigation
  - Integrating through the catalog objects to build the nodes in the tree view control.

COPYRIGHT © 2007 BUSINESS OBJECTS S.A.