



SharePoint File System Replication Tool

Formation Technology, 2006

SharePoint File System Replication Tool	3
What is it:	3
Problem	3
Environment.....	3
Technology	4
Initialising the application.....	5
Creating an Area	6
Deleting an Area.....	7
Bulk Migration of Directory Tree to SharePoint	7
SharePoint Templates	9
About Formation Technology.....	10

SharePoint File System Replication Tool

By Formation Technology

What is it?

This tool provides a quick way for SharePoint Administrator's to build and populate a portal by replicating a file structure and its contents.

This tool was developed by Formation Technology, Melbourne Australia.

Problem

We needed a way to migrate a large folder structure into a SharePoint Portal. Each folder need to appear in SharePoint as separate "area" and the contents of that folder needed to appear a document library within the corresponding area.

To present the information appropriately, a custom template was required to publish a document library webpart, and an area details webpart.

Our initial assessment of the project was that it take days of effort to manually configure and populate the portal.

Environment

The application (henceforth referred to as the File System Replicator for SharePoint or FSR) needed to communicate with SharePoint Portal Server 2003 running on Windows Server 2003 in a domain authenticated environment.

FSR needed to be run on both Windows XP SP 2 and on the actual portal server (Windows Server 2003).

As access to the SharePoint Administration Services was restricted, the application needed to connect to the SharePoint server as an authenticated user.

Technology

FSR for SharePoint was written as a Windows Forms application in Visual Studio.Net 2003.

The tool provides for the following configurable properties:

- DefaultTemplate
Default template is the name of the template that should be pre-selected in the template list.

- AreaServiceUrl
Area Service URL is the URL used to create an instance of the Area Service web service proxy; as every portal site's URL varies this needs to be configurable.
- SitesServiceUrl
Sites Service URL is the URL used to create an instance of the Sites web service proxy.
- UriRootPath
Url Root Path is the URL path to the root of the service. This is used by the file uploading code to correctly place files in the area's document library.
- DocumentLibraryPath
Document Library Path is the directory name of the document library. SharePoint does not require that the document library is named "Document Library". This name is configurable in the template. In our example we named the document library "Area Contents" which was the title of the document library also the path (ie. /Area Contents/AllItems.aspx).

To interact with SharePoint Portal Server the FSR for SharePoint does most of its operations via the inbuilt services being:

1. Area Service (<http://server/vti/bin/AreaService.aspx>)

This service is used to build the tree of areas for the user interface. It also provides the facility to create and delete areas.

See: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/spptsdk/html/cAreaService_SV01071465.asp

2. Sites (<http://server/vti/bin/Sites.aspx>)

Using the Sites web service a list of templates is populated.

See: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/spptsdk/html/cAreaService_SV01071465.asp

To instantiate the services two methods (one for each web service) were created to return an instance of the respective web service proxy using the correct URL (as defined in the application configuration file).

```
static AreaService CreateAreaService(){
    AreaService service = new AreaService();
    service.Credentials = System.Net.CredentialCache.DefaultCredentials;
    service.Url =
System.Configuration.ConfigurationSettings.AppSettings["AreaServiceURL"];
    return service;
}

static Sites CreateSiteService(){
    Sites service = new Sites();
    service.Credentials = System.Net.CredentialCache.DefaultCredentials;
    service.Url =
System.Configuration.ConfigurationSettings.AppSettings["SitesServiceURL"];
    return service;
}
```

Figure 1

As shown in Figure 1, an instance of the proxy is created and the credentials of the configured server URL is set.

As it's written the "DefaultCredentials" are used which means the user running the application requires administrative access to the SharePoint site.

Initialising the application

The first step is to initialise the tree within the root area.

```
void RefreshAreaList() {
    areaList.Nodes.Clear();

    using (AreaService service = CreateAreaService()) {
        Guid homeAreaId = service.GetHomeAreaID();
        AreaData data = service.GetAreaData(homeAreaId);

        TreeNode homeNode = new TreeNode(data.AreaName);
        homeNode.Tag = homeAreaId.ToString();

        areaList.Nodes.Add(homeNode);

        RecursiveTreeSearch(homeAreaId, homeNode, service);

        homeNode.Expand();
    }
}

void RecursiveTreeSearch(Guid parentId, TreeNode parentNode, AreaService service) {
    Guid[] firstTierAreaIds = service.GetSubAreas(parentId);

    for (int i = 0; i < firstTierAreaIds.Length; i++) {
        AreaData data = service.GetAreaData(firstTierAreaIds[i]);
        TreeNode tierNode = new TreeNode(data.AreaName);
        tierNode.Tag = firstTierAreaIds[i].ToString();

        parentNode.Nodes.Add(tierNode);
        RecursiveTreeSearch(firstTierAreaIds[i], tierNode, service);
    }
}
```

Figure 2

Figure 2 demonstrates some of nuances in using the SharePoint web services. It is not possible to request a complete tree from the Area Service. A call to GetSubAreas on the Area Service accepts an area ID and returns an array of GUIDs which identify the children of the provided area. With that ID you can then make a call to GetAreaData providing the ID of the area which you require more information on.

Carry on recursively going through this process for each area until all area information has been extracted from the SharePoint Portal. For convenience and accuracy I use the dotnet Tree Control to represent the area structure.

Next, initialise the template list.

```
using (Sites sites = CreateSiteService()) {
    Template[] templates;
    sites.GetSiteTemplates(1033, out templates);

    templateList.ValueMember = "Name";
    for (int i = 0; i < templates.Length; i++)
        if (templates[i].Name.StartsWith("SPS"))
            templateList.Items.Add(templates[i].Name);

    string defaultTemplate =
System.Configuration.ConfigurationSettings.AppSettings["DefaultTemplate"];

    for (int i = 0; i < templateList.Items.Count; i++)
        if (templateList.Items[i].ToString() == defaultTemplate) {
            templateList.SelectedIndex = i;
        }

    if (templateList.SelectedIndex == -1 && templateList.Items.Count > 0)
        templateList.SelectedIndex = 0;
}
```

Figure 3

Figure 3 shows the initialisation procedure for the template list. The Sites Service is used to return a list of templates from the SharePoint site (This is the only function of the Sites service).

We simply enumerated through the returned template array and add each template to the template list combo box control. Only templates with names beginning with SPS are included in the list as this is the convention we follow at Formation Technology. Each item is checked before it is selected to see if it is the nominated default template.

Creating an Area

The Area Service is used to create an area. The service requires a parent area id (GUID), area name and the name of the template to use.

```
if (areaList.SelectedNode == null)
    return;

if (newAreaName.Text.Length == 0)
    return;

using (AreaService service = CreateAreaService()) {
    Guid selectedArea = new Guid(areaList.SelectedNode.Tag.ToString());
    service.CreateArea(selectedArea, newAreaName.Text,
templateList.SelectedItem.ToString());
}
```

Figure 4

To create an area the user nominates an area from the area list tree control to be used as the new area's parent. The user then types the name of the area in the provided field and clicks Create Area.

At this point the code verify's a parent node has been chosen and that a name has been entered, the service is then initialised and the CreateArea command is issued.

Deleting an Area

To delete an area, you will only require the ID of the area to be deleted.

```
if (areaList.SelectedNode == null)
    return;

DialogResult chk = MessageBox.Show("Are you sure you want to permanently delete this
area?", "Delete Area", MessageBoxButtons.YesNo);

if (chk == DialogResult.No)
    return;

using (AreaService service = CreateAreaService()) {
    Guid homeAreaId = service.GetHomeAreaID();
    Guid selectedArea = new Guid(areaList.SelectedNode.Tag.ToString());

    if (homeAreaId != selectedArea)
        service.DeleteArea(selectedArea);
}
```

Figure 5

To delete an area the user nominates an area from the area list tree control and then clicks Remove Area. We verify that the user isn't trying to delete the Home area. Then we call DeleteArea and pass the ID of the area to be deleted.

Bulk Migration of Directory Tree to SharePoint

To enable the bulk migration of a directory into Sharepoint follows these steps:

1. Select a template
2. Select a portal site area (for all the new data to appear under)
3. Click on "Begin Insert"

4. Select a directory to migrate.
5. Select "OK"

The time to run the application will depend on the number and size of you selected folder and files. Once the migration has completed, a dialog will appear with "Success!".

```
DialogResult result = recursionPoint.ShowDialog();

if (result != DialogResult.OK)
    return;

DirectoryInfo rootRecursionFolder = new
DirectoryInfo(recursionPoint.SelectedPath);
Guid selectedAreaId = new Guid(areaList.SelectedNode.Tag.ToString());

using (AreaService service = CreateAreaService()) {
    RecurseFolder(rootRecursionFolder, service, selectedAreaId);

    MessageBox.Show("Success!");
}

RefreshAreaList();
```

Figure 6

Figure 6 shows the process where a user is prompted to nominate a directory to migrate. This top level directory will not be created on the SharePoint server however it's files will be migrated to the server. RecurseFolder is then called to traverse and migrate the entire directory tree. RecurseFolder receives the service instance, current directory information and matching parent area id.

```
void RecurseFolder(DirectoryInfo folderInfo, AreaService service, Guid parentId){
    DirectoryInfo[] subFolders = folderInfo.GetDirectories();

    for (int i = 0; i < subFolders.Length; i++) {
        Guid areaId = service.CreateArea(parentId, subFolders[i].Name,
templateList.SelectedItem.ToString());
        RecurseFolder(subFolders[i], service, areaId);
    }

    AreaData areaData = service.GetAreaData(parentId);
    FileInfo[] fileInfo = folderInfo.GetFiles();
    string rootPath =
System.Configuration.ConfigurationSettings.AppSettings["UrlRootPath"],
        documentLibraryPath =
System.Configuration.ConfigurationSettings.AppSettings["DocumentLibraryPath"];

    for (int i = 0; i < fileInfo.Length; i++) {
        HttpWebRequest request = (HttpWebRequest)WebRequest.Create(rootPath +
areaData.WebUrl + "/" + documentLibraryPath + "/" + fileInfo[i].Name);
        request.Credentials = CredentialCache.DefaultCredentials;
        request.Method = "PUT";
        Stream requestStream = request.GetRequestStream();
        StreamWriter writer = new StreamWriter(requestStream);
        FileStream sourceFile = fileInfo[i].OpenRead();
        byte[] buffer = new byte[sourceFile.Length];
        sourceFile.Read(buffer, 0, buffer.Length);
        writer.Write(buffer);
        sourceFile.Close();
        writer.Close();

        try {
            WebResponse response = request.GetResponse();
            response.Close();
        } catch (WebException) {
            MessageBox.Show("File: " + request.RequestUri.ToString() +
"\n\nCould not be created.");
        }
    }
}
```

Figure 7

In Figure 7 you can see the workings of RecurseFolder. First it loops through the selected folders / subfolders, creating an area for a folder and then calling RecurseFolder to process the sub folders.

After looping through the sub folders all the files in the folder are then deployed to the area document library. The name of the document library is configurable.

Files are deployed to the document library by performing a PUT http request against the SharePoint portal server. The file data is read and then written out to the request stream. This method is simplistic and doesn't allow any meta-data to be defined for any custom columns that may or may not be defined for the document library.

SharePoint Templates

SharePoint templates are typically stored in this folder: \Program Files\Common Files\Microsoft Shared\web server extensions\60\TEMPLATE\1033\. To create a new template, copy an existing template folder (such as SPSNEWS or SPSHOME) and then name it (SPSTEST for example). Next go to the XML folder and edit WEBTEMPSPS.XML. Look for the Template tag with a Name attribute that matches the your copied template. Copy it and change the name Name and ID (use a number greater than 10000). Feel free to experiment with the other attributes.

```
<Template Name="SPSKWPAGE" ID="10000">
  <Configuration ID="0" Title="Testing Area Template for Acme " Type="0" Hidden="TRUE"
  ImageUrl="..\images/spshome.gif" Description="Area Template."><!-- _locID@Title="webtemp_title_spsnews0"
  _locComment="{StringCategory=HTX}" --> <!-- _locID@Description="webtemp_desc_spsnews0"
  _locComment="{StringCategory=HTX}" --></Configuration>
</Template>
```

Figure 8

Figure 8 shows an example of what your Template tag should look like. Save WEBTEMPSPS.xml and reset the W3SVC service on your Portal server. Note that when u create an area using a template, subsequent changes to that template are not reflected in the areas you have already created. So it's important to get it right first time, or you will need to regenerate your area tree (or manually apply changes).

In your template folder, the key files you can work with are the default.aspx and the xml files in the XML folder. In the lists folder you'll find a folder for the different types of lists you can include in your template. In the DOCLIB folder for example you'll find files allowing you to completely customise the look of all aspects of the document library. You can also edit Schema.xml to control the different views of the document library. Editing this file is hard work and the sheer size of it (a bit under 200K) can make it easy to get lost.

Under Views look at the View tags and the ID tag. The value of the ID tag should match what is on your *SPSTEMPLATE/XML/Onet.xml* file under Modules. See Figure 9.

```
<Module Name="Default" Url="" Path="">
  <File Url="default.aspx" Type="Ghostable">
  <View List="101" BaseViewID="20" WebPartZoneID="MiddleLeftZone" WebPartOrder="1" />
  </File>
</Module>
```

Figure 9

The BaseViewID value should match the ID attribute value of the View element in Schema.xml. The List attribute value should match the ID attribute value of the ListTemplate element for the matching list on Onet.xml (see Figure 10).

```
<ListTemplate Name="doclib" DisplayName="Document Library" Type="101" BaseType="1"
OnQuickLaunch="TRUE" SecurityBits="11" Description="Create a document library when you have a collection of
documents or other files that you want to share. Document libraries support features such as sub-folders, file
versioning, and check-in/check-out." Image="/_layouts/images/itdl.gif" DocumentTemplate="101" />
```

A more detailed explanation of the configuration of SharePoint templates is outside the scope of this article.

Be sure to send us an email with your feedback.

EMAIL: SharePoint@formation.com.au

About Formation Technology

Formation Technology is a professional services company providing innovative IT business solutions. We have a team of MCP's are trained and dedicated to deal with complex business challenges. We have built our reputation on a no-nonsense, "on time and on budget" approach that has seen the successful deployment of: solutions that support thousands of concurrent users, portals that improve collaboration and knowledge worker productivity, and expert internet and intranet systems.

To find out more, visit: www.formation.com.au

Formation Technology

EMAIL: sales@formation.com.au