

NPM Project Development

by

Jacob Wells

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements
for
the Degree of Bachelor of Science
in Information Engineering Technology

© Copyright 2002 Jacob Wells

The information in this document is proprietary and may not be reproduced or distributed
in whole or in part without the permission of the owner.

Jacob Wells

Date

Thomas Wulf, Faculty Advisor

Date

Lawrence Gilligan, Department Head

Date

Table of Contents

Section	Page
Table of Contents	i
List of Figures	iv
Abstract	vi
1. Statement of the Problem	1
2. Description of Solution	1
2.1 User Profiles	2
2.1.1 Nims Personnel	2
2.1.2 Client Users	3
2.2 Design Protocols	3
2.2.1 ASP Pages	3
2.2.2 Data Access Object	4
2.2.3 Database	5
2.2.4 Flash Files	6
2.2.5 XML/XSL	7
2.2.6 Color Scheme	7
2.2.6.1 Intranet Site	8
2.2.6.2 Internet Site	8
2.2.7 Navigation	9
2.2.8 Help	10
3 Deliverables	11
3.1 User Group Management	11
3.2 Presentation Management	11
3.3 Client Presentation Site	12
4 Design and Development	12
4.1 Budget	12
4.2 Timeline	13
4.2.1 First Quarter	13
4.2.2 Second Quarter	14
4.2.3 Third Quarter	16
4.3 Hardware	16
4.4 Software	17
5 Proof of Design	17
5.1 Page Flows	17
5.1.1 Intranet Site	17
5.1.2 Internet Site	18
5.2 User/Group Management	19
5.2.1 User Management Page	19

5.2.1.1	Page Function	19
5.2.1.2	Data Source	20
5.2.1.3	Data Elements	20
5.2.1.4	Available Actions	20
5.2.1.5	Validation	20
5.2.1.6	Page Layout	21
5.2.1.7	Flow Chart	22
5.2.2	Group Management	23
5.2.2.1	Page Function	23
5.2.2.2	Data Source	23
5.2.2.3	Data Elements	23
5.2.2.4	Available Actions	23
5.2.2.5	Validation	24
5.2.2.6	Page Layout	24
5.2.2.7	Flow Chart	25
5.2.3	User Group Association	25
5.2.3.1	Page Function	25
5.2.3.2	Data Source	25
5.2.3.3	Data Elements	26
5.2.3.4	Available Actions	26
5.2.3.5	Page Layout	26
5.2.3.6	Flow Chart	27
5.3	Presentation Management	27
5.3.1	Presentation Creation	27
5.3.1.1	Page Function	28
5.3.1.2	Data Source	28
5.3.1.3	Data Elements	28
5.3.1.4	Available Actions	28
5.3.1.5	Validation	28
5.3.1.6	Page Layout	29
5.3.1.7	Flow Chart	30
5.3.2	Presentation Association	30
5.3.2.1	Page Function	30
5.3.2.2	Data Source	31
5.3.2.3	Data Elements	31
5.3.2.4	Available Actions	31
5.3.2.5	Page Layout	31
5.3.2.6	Flow Charts	32
5.3.3	Content Edit	32
5.3.3.1	Page Function	32
5.3.3.2	Data Source	33
5.3.3.3	Data Elements	33
5.3.3.4	Available Actions	33
5.3.3.5	Page Layouts	34
5.3.3.6	Flow Chart	35

5.3.4	Presentation Download	35
5.3.4.1	Page Function	35
5.3.4.2	Data Source	36
5.3.4.3	Data Elements	36
5.3.4.4	Page Layout	36
5.3.4.5	Flow Chart	36
5.4	Client Presentation Site	37
5.4.1	Client Login Page	37
5.4.1.1	Page Function	37
5.4.1.2	Data Source	37
5.4.1.3	Data Elements	37
5.4.1.4	Validation	37
5.4.1.5	Page Layout	38
5.4.1.6	Flow Chart	38
5.4.2	Client Presentation List Page	38
5.4.2.1	Page Function	38
5.4.2.2	Data Source	39
5.4.2.3	Data Elements	39
5.4.2.4	Available Actions	39
5.4.2.5	Page Layout	39
5.4.2.6	Flow Chart	40
5.4.3	Client Change Password Page	40
5.4.3.1	Page Function	40
5.4.3.2	Data Source	40
5.4.3.3	Data Elements	40
5.4.3.4	Available Actions	41
5.4.3.5	Validation	41
5.4.3.6	Page Layout	41
5.4.3.7	Flow Chart	42
6	Conclusions and Recommendations	43
	Appendix A. Selection from Page Building (default.asp)	45
	Appendix B. Selections from Client Processing (default.asp)	49
	Appendix C. Selections from Server Processing (actions.asp)	51
	Appendix D. Stored Procedure Definitions	53
	Appendix E. Select Code from DB_Connector	55
	References	58

List of Figures

Figure 1. Database Diagram	5
Figure 2. Loading XML File	6
Figure 3. Nims Intranet Color Scheme	8
Figure 4. Nims Internet Color Scheme	9
Figure 5. Navigation Structure	10
Figure 6. Tabbed Navigation	10
Figure 7. Help Dialog	11
Figure 8. Intranet Page Flow	18
Figure 9. Internet Page Flow	19
Figure 10. User Management	21
Figure 11. User Management Flow Chart	22
Figure 12. Group Management	24
Figure 13. Group Management Flow Chart	25
Figure 14. User Group Association	26
Figure 15. User Group Association Flow Chart	27
Figure 16. Presentation Creation	29
Figure 17. Presentation Creation Flow Chart	30
Figure 18. Presentation Association	31
Figure 19. Presentation Association Flow Chart	32
Figure 20. XML Format	33
Figure 21. Content Edit: Master Content	34

Figure 22. Content Edit: Scene Content	34
Figure 22. Content Edit Flow Chart	35
Figure 24. Presentation Download	36
Figure 24. Presentation Download Flow Chart	36
Figure 26. Client Login Page	38
Figure 27. Login Flow Chart	38
Figure 28. Presentation List Page	39
Figure 29. Presentation List Page Flow Chart	40
Figure 30. Client Change Password Page	41
Figure 31. Set Password Page	42
Figure 32. Change and Set Password Flow Chart	42

Abstract

Nims Associates, Inc., an IT services company, needs an effective way to create high quality sales presentations that represent company services to customers with a small investment of time by the sales representative. The NPM Tool was developed to satisfy this need. NPM provides the ability to create high quality customized presentations through a web interface that can be downloaded or published to the Internet for access by the customer. NPM allows the user to choose from presentation styles, select scenes to include in the presentation, customize content, and assign presentations to a client user. Pre-developed content and presentation structure allows presentations to be created in a short time and presents corporate strategies appropriately.

NPM Project Development

1. Statement of the Problem

Corporate strategies and services are not adequately presented in sales presentations that are produced by the sales force of Nims Associates.

Sales Managers do not have an adequate knowledge of corporate services, and tend to insufficiently present the services that our company can provide. Sales Managers create presentations based on their client's needs, but do not have the resources to allow for the presentation of other services that Nims can provide that may further bolster an account. Corporate strategies and services are lost due to lack of knowledge and resources.

Sales Managers have limited knowledge of advanced presentation techniques

Sales Representatives have a general knowledge of programs such as PowerPoint, but have a difficult time with creating dynamic presentations. The sales force primary focus is not in multimedia or design, and due to this fact, sales presentations are less than sophisticated.

2. Description of the Solution

The NPM tool, a Web based application, utilizes Flash 5 and XML to provide the ability to create high quality presentations in a timely manner.

Centralized library of presentation templates

By providing a centralized library of customizable presentation templates, the sales force will be able to appropriately portray the company and the services that can be provided by Nims.

Internet availability of Nims Presentations

Client presentations will live past the initial delivery through Internet availability. In addition, because the presentations are hosted on the Nims Internet site, additional information found on the site could potentially spawn new client interest. Clients can visit the site at any time to review the presentation without contacting the sales representative.

Simple Web based application for creation of sales presentations

The presentation maintenance piece of NPM shortens the time involved in creating presentations, and in turn, gives more time for other sales tasks such as meeting with the client. Additionally, through the use of Flash, high-level presentations can be created without training the sales force on Flash development. The sales representative will simply input basic information about the client and choose from the library of sales presentation templates to best fit the needs of their client, and the application will create a customized presentation.

2.1 User Profile

2.1.1 Nims Personnel

Primary User

Primary Function will be the creation of presentations through the presentation creation wizard. This type of format (wizard) should be familiar to the user as many of the applications used by this group use a similar type of approach.

Sales representatives have a great deal of experience in creating presentations, so the development of content to insert into the presentations should require little additional knowledge.

This group has a medium level of technical literacy and should be comfortable navigating this application. This application will use the Nims Intranet Style and navigation structure, so there will be little or no learning curve associated with this application in regards to navigation as many of the Nims applications are currently leveraging the Intranet Site.

2.1.2 Client Users

Secondary Users

This user will vary in regards to technical literacy. The portion of the application that will be viewed by the client will require very little technical literacy. The client user will access the application through the Nims Internet site, and once logged in, will be presented with a list of available presentations. This will require the user to be accustomed to navigating a Web site. A help link will be available to give information on how to open presentations, or how to contact a Nims Representative if any problem occurs.

2.2 Design Protocols

2.2.1 ASP Pages

The main user interface will be built using Active Server Pages. All pages will use some or all of the following technologies:

- VBScript (server)
- JavaScript (client)
- Data Access Component – VB written wrapper of ADO
- XML/XSL

All pages will use a base template file to allow for consistency across all pages. Style will be maintained in a Cascading Style Sheet and so all style definitions should be included in this file. ASP pages will build a string, which is referenced by the base template file (see Appendix A).

Pages will utilize the XMLHTTP class in the MSXML 3 extensively to provide an enhanced user experience. By using this feature of the XML object, there is a perceived performance gain. The XMLHTTP class allows posting of XML from the client (See Appendix B) to an ASP page which can process it on the server side (See Appendix C), and respond back to the client. This feature allows the client to interact with the server without the need to reload the ASP page, which, in addition to the perceived performance gain, allows a Web application to resemble a traditional client server application.

2.2.1 Data Access Object

- VB Component, wraps ADO connection, command, and recordset objects
- Functions
 - RunSQLString
 - RunSQLStringReturnRS
 - RunSQLStringReturnXML (see Appendix E)

2.2.2 Database

The backend is a SQL Server 2000 Database comprised of six tables (see Figure 1). All table names are prefixed with “t_” and are named based upon their function.

- t_styles – holds all styles which can be used as templates
- t_prstn – holds all presentations which have been created through NPM
- t_grp – holds all presentation groups
- t_usr – holds all client users
- t_grpprstn – group presentation join table
- t_grpusr – user group join table

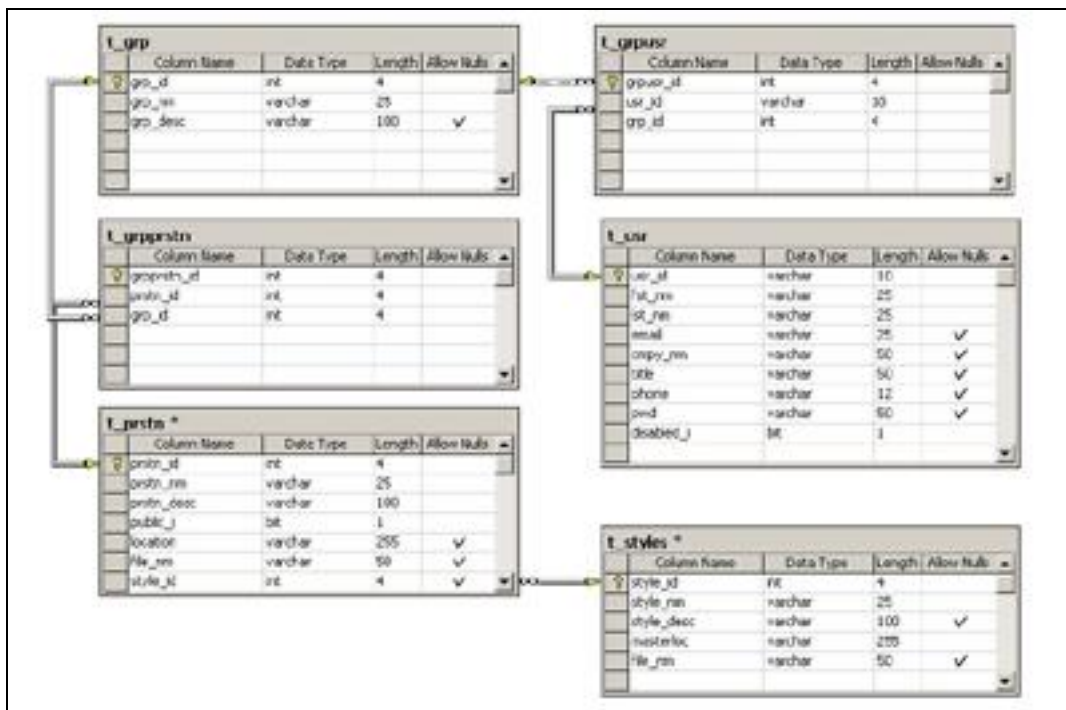


Figure 1. Database Diagram

All data access is done through stored procedures (see Appendix D), which are accessed from the ASP user interface through one of the methods contained in the Data Access Component. All Stored procedures are prefixed with “p_” and are named based upon their function (see Appendix D).

2.2.3 Flash Files

Flash 5 was chosen because of its ability to parse XML and integrate the contents of a XML Dom into the content of the Flash Scene. Through the use of Flash’s Active Script language, the presXML.xml file is opened at run time, parsed, and integrated into the applicable scene. The Flash SWF file is dependent upon the following three files:

- presXML.xml – contains the contents of the presentations
- navigation.as – built dynamically to provide navigation within the presentation
- common.as – common functions that are used in the Active Scripting within the SWF file. Includes code from “XMLNitro” (1) which quickens the parsing of XML and allows the parser to ignore white space in the DOM.

The initial page of all presentations will open the xml file and assign nodes to various constants. Other scenes in the presentation use these constants to pull data from the XML Dom in order to populate the dynamic content. The following code illustrates the opening and assigning of the master content variables.

```
#include "common.as"
#include "navigation.as"
var loaded
function myLoad()
{
    loaded = true
    XMLNode = this.firstChild
    mstNode = XMLNode.childNodes[SelectNodeByName("mastercontent",
XMLNode)]
    scenesNode = XMLNode.childNodes[SelectNodeByName("scenes",
XMLNode)]
}
```

```

        strPresTitle =
mstNode.childNodes[SelectNodeByID("strPresTitle",
mstNode)].attributes["value"]
        strPresDate =
mstNode.childNodes[SelectNodeByID("strPresDate",
mstNode)].attributes["value"]
        strPresenter =
mstNode.childNodes[SelectNodeByID("strPresenter",
mstNode)].attributes["value"]
        strCompanyName =
mstNode.childNodes[SelectNodeByID("strCompanyName",
mstNode)].attributes["value"]

        scenesNode = XMLNode.childNodes[SelectNodeByName("scenes",
XMLNode)]

    }

if(loaded != true)
{
    objXML = new XML()
    objXML.ignoreWhite = true;
    objXML.onLoad = myLoad;
    objXML.load("presXML.xml");
}

stop ();

```

Figure 2. Loading XML File

2.2.4 XML/XSL

To promote code reuse, the XML data store (presXML.xml) is transformed in the user interface using XSL. Additionally, the user interface also uses the Data Access Object's RunSQLStringReturnXML method to return recordsets in XML, which are also transformed and displayed in the user interface.

2.2.5 Color Scheme

The Color Scheme will be representative of the color schemes currently found on the Nims Internet and Intranet Sites.

2.2.5.1 Intranet Site

The Nims Intranet Site is the primary host of the NPM Application. The application will follow the current navigation and style conventions established to assure continuity (see Figure 3).

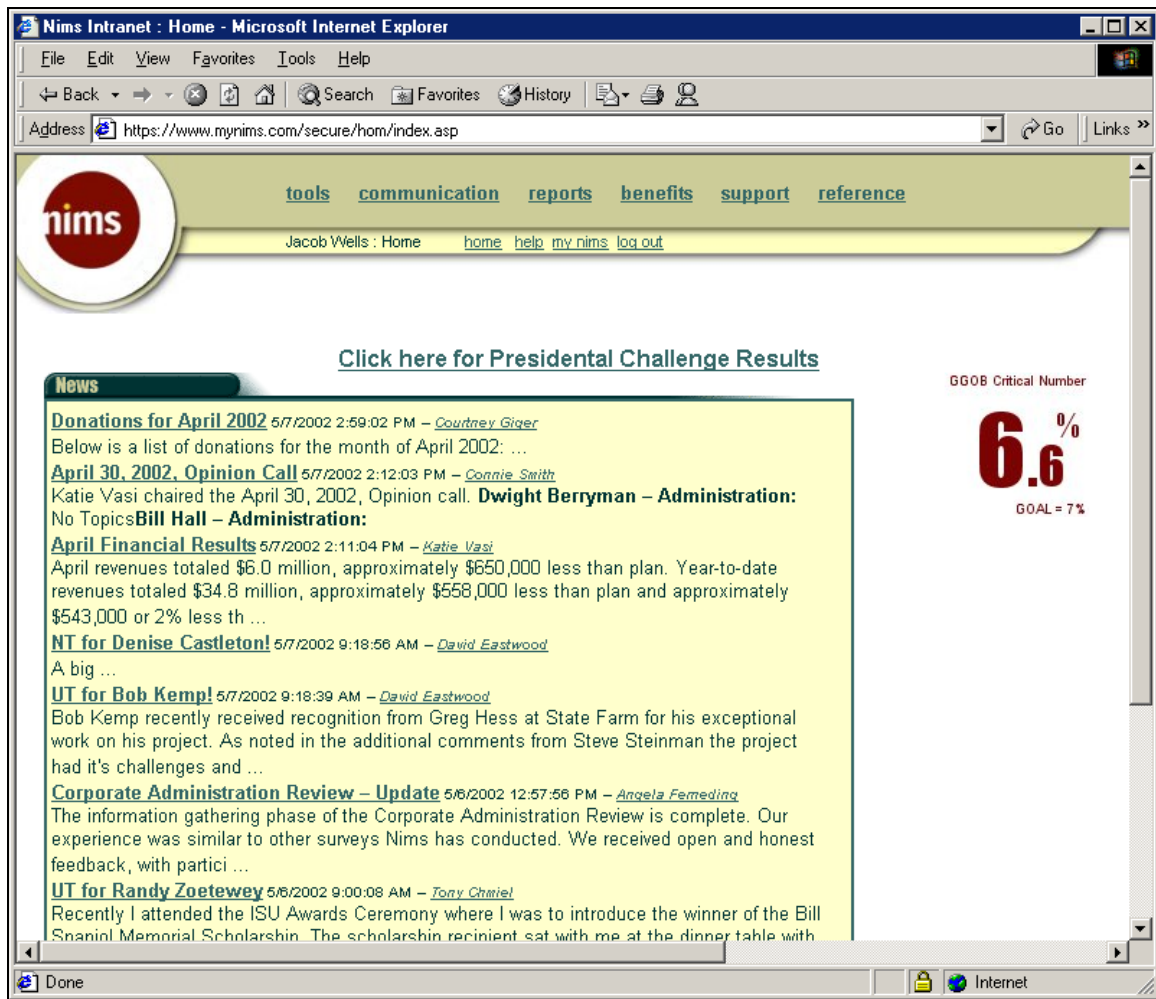


Figure 3. Nims Intranet

2.2.6.2 Internet Site

The Nims Internet Site is the secondary host of the NPM Application. The client presentation list pages will follow the current navigation and style conventions established to assure continuity (Figure 2).



Figure 4. Nims Internet Site

2.2.7 Navigation

Navigation will follow the currently developed navigation scheme represented on the Nims Internet and Intranet sites. The Intranet portion will be found in the menu structure (see Figure 5).

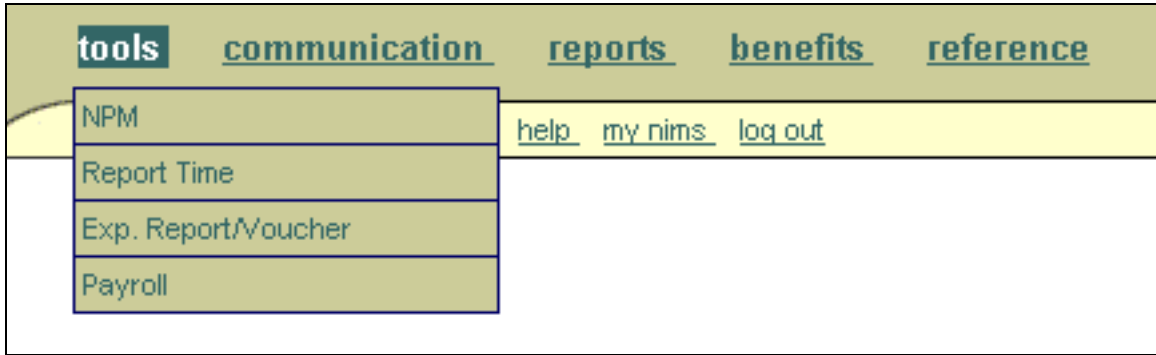


Figure 5. Navigation Structure

At the Application level, users will navigate the application using a tabbed approach (see Figure 6). Each main section will have a tab, when clicked, the application will navigate to that page. Active tabs are indicated by having a black background.



Figure 6. Tabbed Navigation

2.2.8 Help

Each page will have help available, describing the function of the page, the available controls and buttons and actions of the buttons (Figure 7).

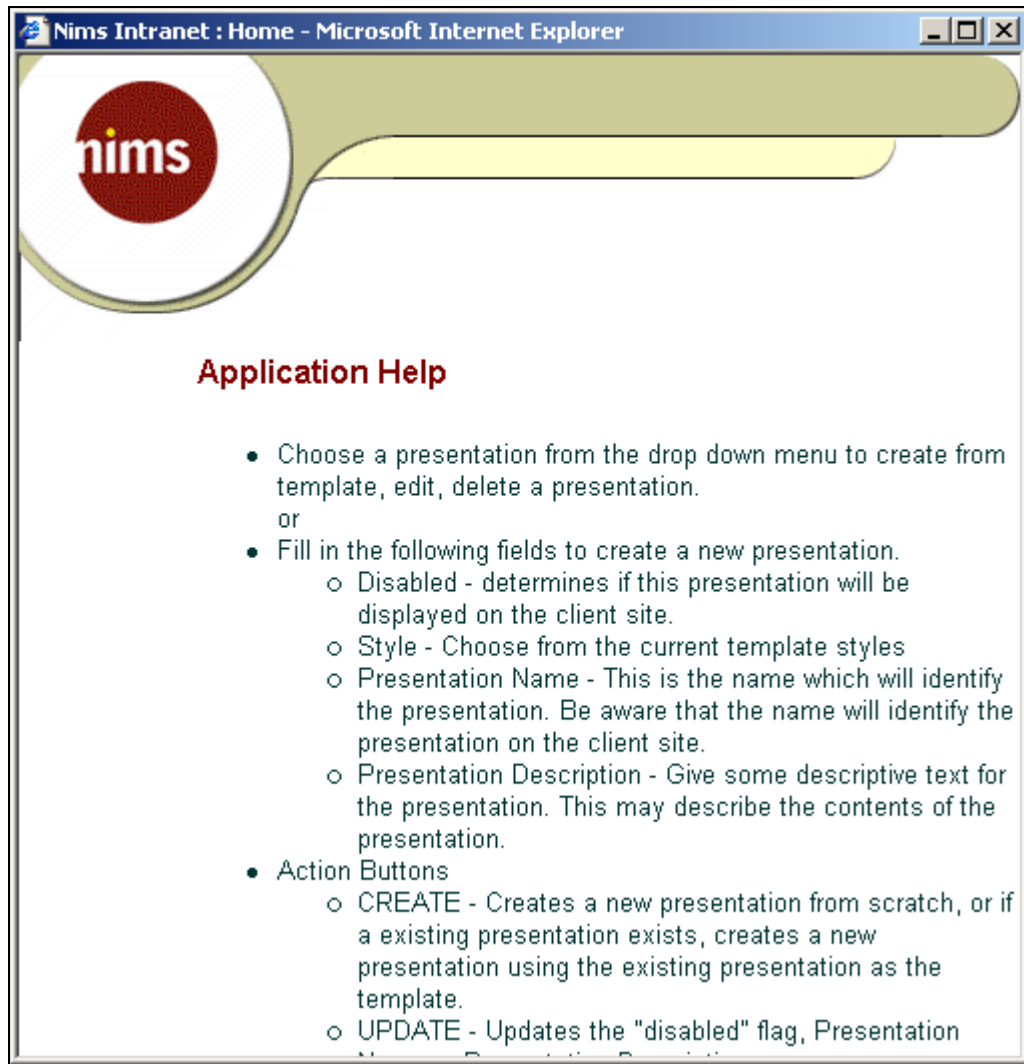


Figure 7. Help Dialog

3. Deliverables

3.1 User/Group Management

- Nims Users will be able to create, edit or remove users
- Nims Users will be able to create, edit or remove groups
- Nims Users will be able to add/remove users from groups

3.2 Presentation Management

- Nims Users will be able to choose from various styles of presentations

- Nims Users will be able to add scenes from a library of scenes associated with a particular style
- Nims Users will be able to make inputs to the presentations to “personalize” the presentations
- Nims Users will be able to assign presentations a group, or make it public
- Nims Users will be able to remove/disable presentations
- Nims Users will be able to download presentation files from the site
- Nims Users will be able to preview presentations throughout the creation process

3.3 Client Presentation Site

- Client Users will be able to accessed a “personalized” site
- Client Users will be able to view a list of presentations that he has rights to based on user id, groups membership or public associations
- Client Users will be able to view presentations on the site

4. Design and Development

4.1 Budget

The following table represents the costs associated with this project.

Item	Use	Cost
PowerApp 120 (2 processors) incl. OS	Intranet Web Server	\$3,305.00 dell.com
PowerApp 120 (2 processors) incl. OS	Internet Web Server	\$3,305.00 dell.com
Dell PowerEdge 6400 (2 processors) incl. OS	Database Server	\$15,100.00 dell.com
Microsoft SQL Server 2000 Enterprise Edition (25-client)	DBMS	\$11,199.99 amazon.com
Visual Studio Enterprise 6.0 with Plus Pack	ASP Development, DLL Development	\$1,429.95 buy.com
Flash 5	SWF Development	\$338.95 buy.com

4.2 Timeline

The following sections illustrate the general timeline followed throughout the three quarters of Senior Design. Each table represents a month and each block represents the a week ending on the date noted in the column header.

4.2.1 First Quarter

	Oct-00			
Task	7	14	21	28
Research				
Project Planning				

Key Dates

- Oct. 12, 2000 - Problem/Area of Inquiry
- Oct. 23, 2000 - Project Planning began

	Nov-00			
Task	4	11	18	25
Research				
Project Planning				

Key Dates

- Nov. 30, 2000 - Final Proposal

	Dec-00				
Task	2	9	16	23	30
Project Planning					

Key Dates

- Dec. 7, 2000 - Oral Presentation

4.2.2 Second Quarter

	Jan-02			
Task	5	12	19	26
High Level Design	█	█	█	
Detail Level Design		█	█	█
Resource Gathering		█	█	█
Server Setup	█			
Database Setup	█			
Site Development				█
Database Development				█
Flash Development				█

Key Dates

- Jan 3, 2002
 - Begin Resource Gathering
 - Server Setup and Database Setup Begins
- Jan 12, 2002
 - Server Setup and Database Setup Complete
- Jan 17, 2002
 - High Level Design Document Completed
 - Begin work on Detail Level Design Document
- Jan 27, 2002
 - Site Development Begins
 - Database Development Begins
 - Flash Development Begins

	February 2002			
Task	2	9	16	23
Detail Level Design				
Resource Gathering				
Site Development				
Database Development				
Flash Development				
Unit Testing				

Key Dates

- Feb 2, 2002 - Detail Level Design Document Completed
- Feb 7, 2002 – Review/Discussion of DLD
- Feb 9, 2002 - Resource Gathering completed
- Feb 23, 2002 - Unit Testing Begins
- Feb 24, 2002 -Work on Final Written Report Begins

	Mar-02				
Task	2	9	16	23	30
Site Development					
Database Development					
Flash Development					
Unit Testing					
Integration Testing					
Prototype Completion					

Key Dates

- Mar 2, 2002 - Prototype Development Completed
- Mar 7, 2002
 - Design Freeze Report Due
 - Prototype Testing Completed
- Mar 14, 2002
 - Oral Presentation of Proof of Concept/Working Quick Prototype

4.2.3 Third Quarter

	Apr-02				
Task	6	13	20	27	
Site Development					
Database Development					
Flash Development					
Unit Testing					
Integration Testing					

Key Dates

- Apr 25, 2002
 - Integration Testing Begins

	May-02				
Task	4	11	18	25	31
Site Development					
Database Development					
Flash Development					
Unit Testing					
Integration Testing					
Final Project Complete					

Key Dates

- May 31, 2002
 - Integration Testing Completes
 - Final Project Complete

4.3 Hardware

- Intranet Web Server – Dell PowerApp 120 (Dual Processor)
- Intranet Web Server – Dell PowerApp 120 (Dual Processor)
- Database Server - Dell PowerEdge 6400 (Dual Processor)

4.4 Software

The application will be hosted on Windows 2000 Servers running IIS 5.0. Development tools include Visual Basic 6.0, Visual Interdev, and Flash 5.0. The bulk of the application is comprised of Active Server Pages using vbscript for server processing and JavaScript for client processing. Microsoft SQL Server 7.0 acts as the primary data store and XML files act as a secondary data store. The application uses two middleware components to interact with the two data stores, Microsoft XML Parser 3.0 (MSXML3.dll) and the DB_Connector object based on the Microsoft Developed FMStocks_DB object (2).

- Windows 2000 Server
- MS SQL Server 2000
- Internet Information Server 5
- Visual Basic 6
- Visual Interdev 6
- Flash 5
- MSXML 3.0

5. Proof of Design

5.1 Page Flows

5.1.1 Intranet Site

Nims users will access a set of pages which will allow the users to create and edit presentation content and manage users and presentations groups. Users will navigate the application using tabs which represent a single module of the application's functionality. Each tab is separate and does not require the user to complete the

entire process during a single session. The following diagram illustrates the seven tabs which provide the functionality of the application.

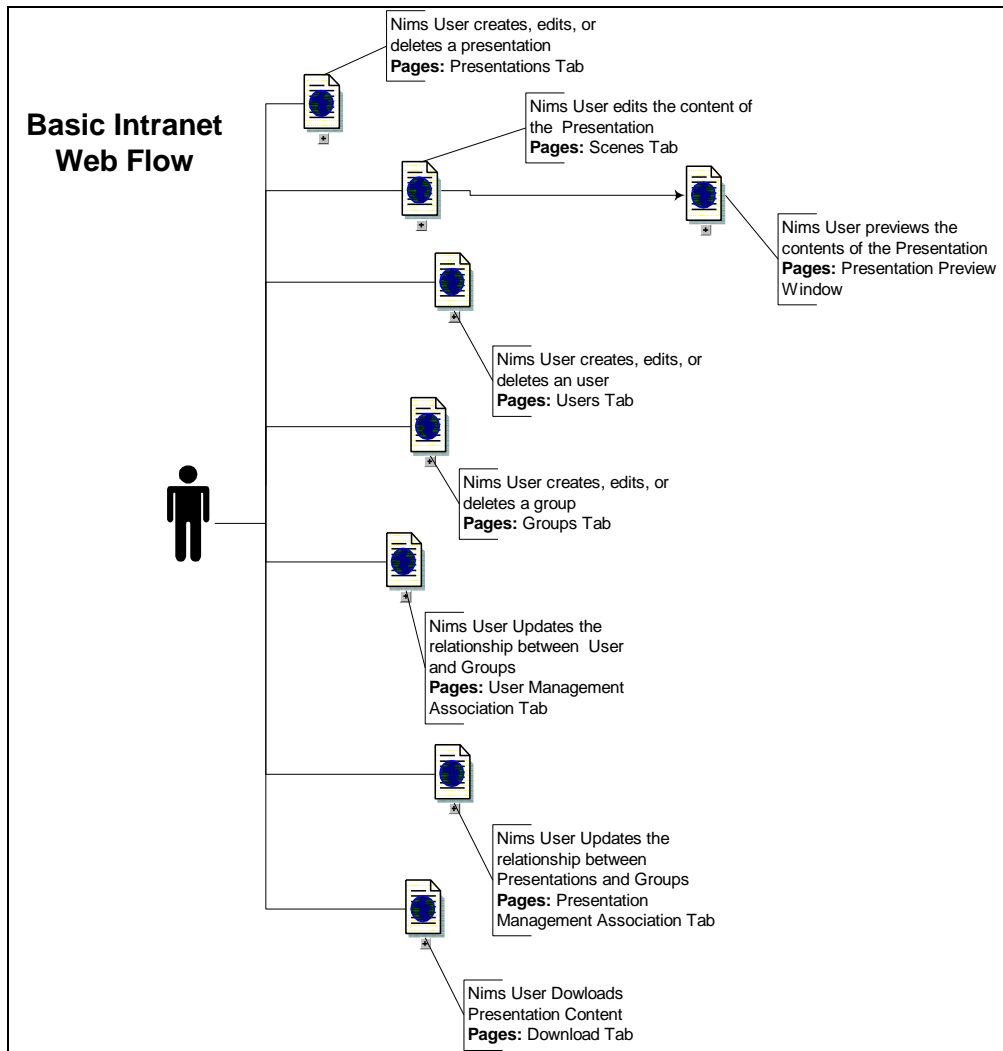
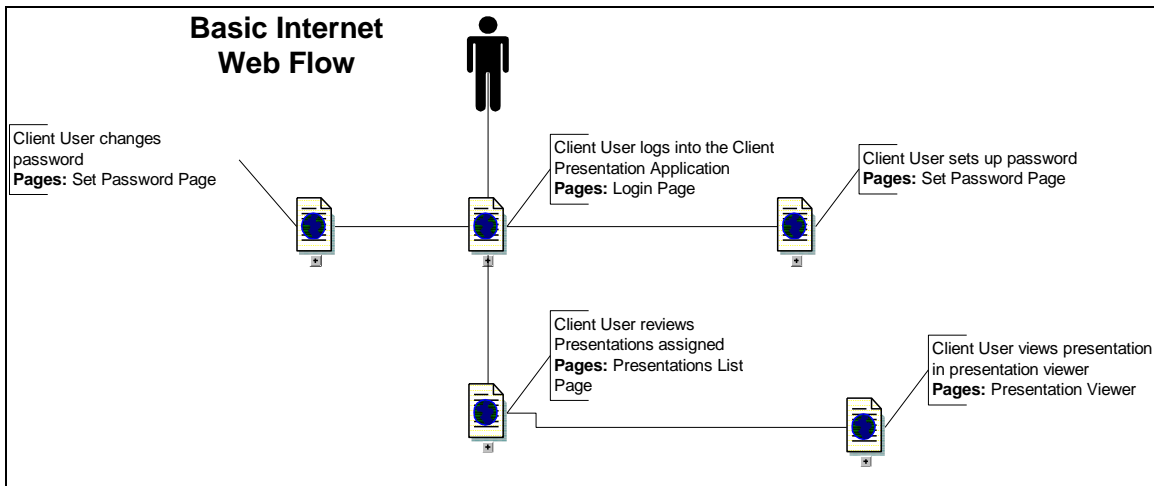


Figure 8. Intranet Page Flow

5.1.2 Internet Site

The Internet portion of NPM application will provide a limited amount of functionality, but ultimately, the end product. This portion includes login, password management, and presentation list pages. The following diagram (Figure 5) illustrates the basic Internet page flow.



5.1.3 Figure 9. Internet Page Flow

5.2 User/Group Management

This section of the application will provide the user the ability to manage users and groups. This section will also allow the Nims user to assign users to the presentation groups. Nims users will also have the ability to reset a user's password in the event that a client user has forgotten his password. These functions are provided through the user interface using the ASP pages detailed in the following sections.

5.2.1 User Management Page

5.2.1.1 Page Function

This page will allow users to create, edit, or delete users in the database. Users to this page can select a user from a list box for edit or deletion, or enter the user information for adding. When the user selects a user from the drop down, the text boxes will be filled with the user information from the database. Update and Delete will be disabled until the user selects a user. After selecting a user, if the user changes the user id, the Update and Delete buttons will be disabled.

5.2.1.2 Data Source

- NimsPres Database

5.2.1.3 Data Elements

- User id
- First Name
- Last Name
- Title
- Company
- Email Address

5.2.1.4 Available Actions

- Create – creates the user in the database
- Update – updates the user's information in the database
- Delete – deletes the user and any group membership
- Clear – clears all form elements

5.2.1.5 Validation

- User ID is checked for duplicates on "add"
- User ID, First Name, and Last Name are required fields
- Confirmation on delete

5.2.1.6 Page Layout

The screenshot displays a web application interface for user management. At the top, there are two tabs: "Presentation Management" and "User Management". Below these, there are three sub-tabs: "Users" (selected), "Groups", and "Assignment". The main content area contains a form for editing a user. The form includes a "Select User" dropdown menu with the value "jake-wells, jake". Below this, there are several input fields: "User ID" (jake), "First Name" (jake), "Last Name" (wells), "Email" (wells_jacob@hotmail.com), "Phone" (513-704-0243), "Title" (Consultant), and "Company" (Fidelity Investments). There is also a "Disabled" checkbox which is currently unchecked. At the bottom of the form, there are five action buttons: "Create", "Update", "Delete", "Clear", and "Reset".

Select User	jake-wells, jake
User ID	jake
First Name	jake
Last Name	wells
Email	wells_jacob@hotmail.com (ie. johndoe@doe.net)
Phone	513-704-0243 (ie. 678-555-9013)
Title	Consultant
Company	Fidelity Investments
<input type="checkbox"/> Disabled	

Figure 10. User Management

5.2.1.7 Flow Chart

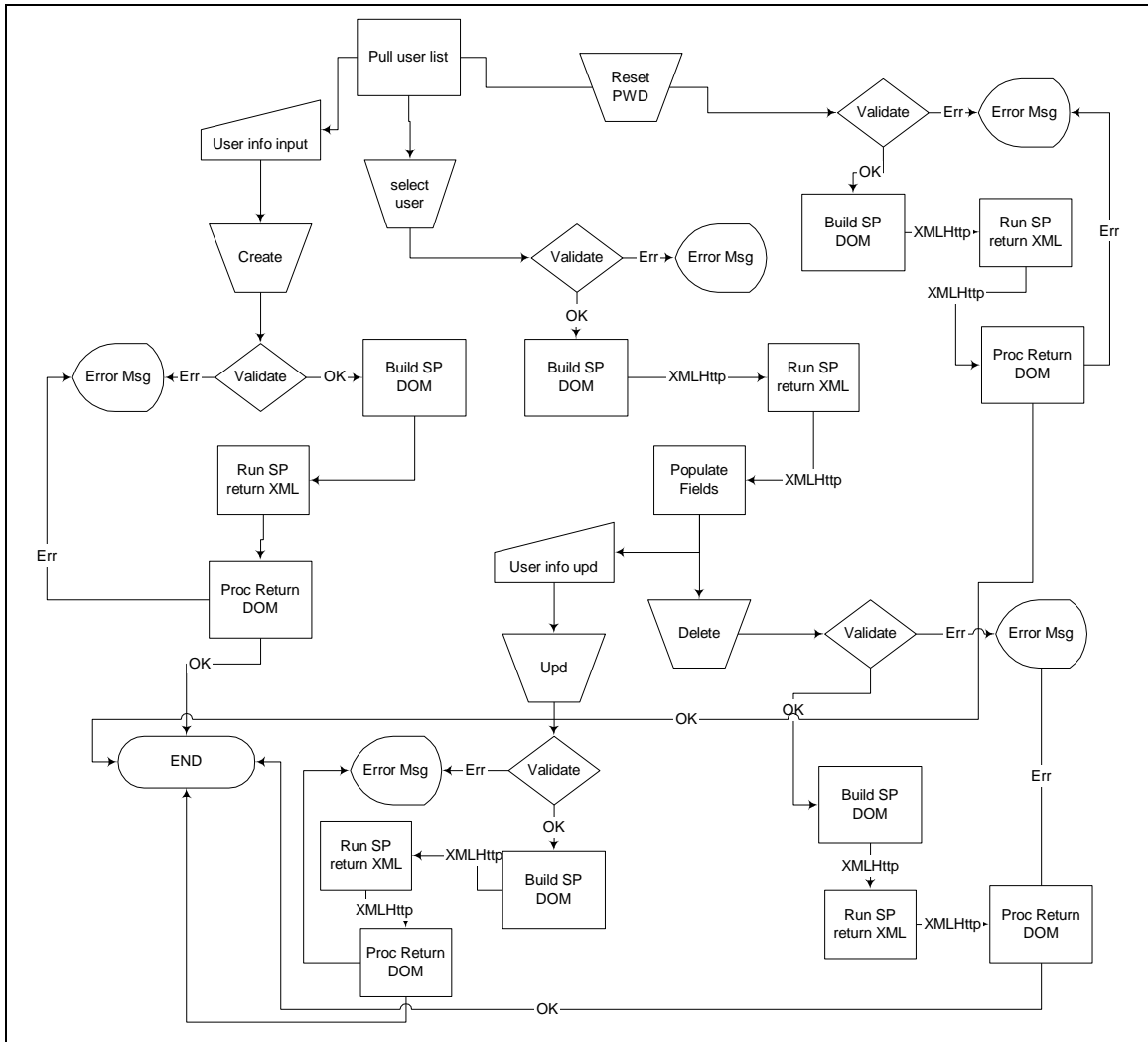


Figure 11. User Management Flow Chart

5.2.2 Group Management

5.2.2.1 Page Function

This page allows the user to create, edit, or delete groups. Users can select the group from the drop down for editing or deleting, or can fill in the information and add the new group. When a group is selected from the drop down, the text boxes will be filled with the information contained in the database. Update and Delete buttons will be disabled until the user selects a user from the drop down. Group assignments are deleted when a group is deleted. Presentation assignments to groups are deleted when a group is deleted. Users are not removed from the system when a group is deleted. In addition, the Nims user will also be able to reset the password of the user through this interface. The reset will set the user's password to null, and will force the client user to enter a new password on the next visit to the site.

5.2.2.2 Data Source

- NimsPres Database

5.2.2.3 Data Elements

- Group List Box
- Group Name
- Group Description

5.2.2.4 Available Actions

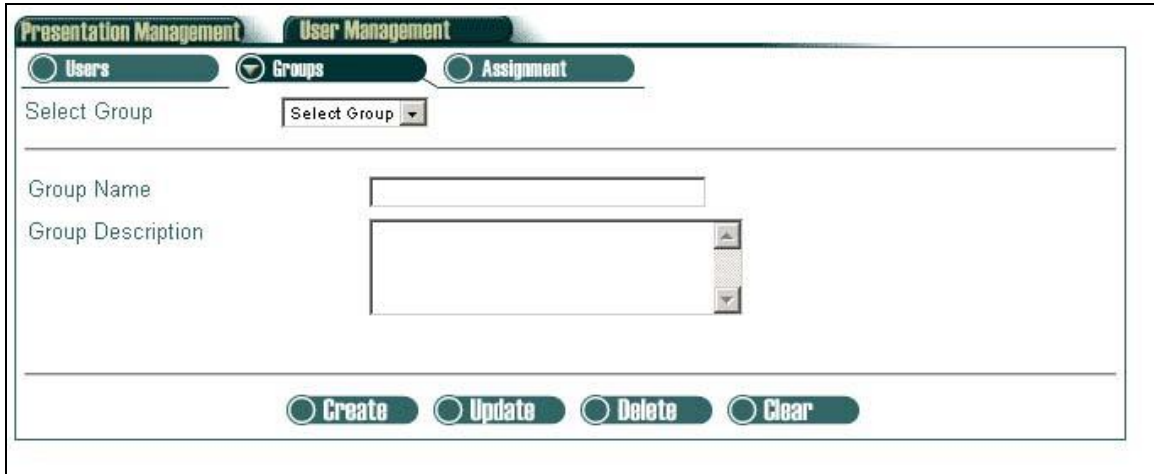
- Create – Creates new group
- Update – Updates group name, group description
- Delete – deletes group from database, eliminates any associations with users and presentations
- Clear – clears all form elements

- Reset – resets the users password to “null”

5.2.2.5 Validation

- Group Text box must have text
- Delete is confirmed

5.2.2.6 Page Layout



The screenshot displays a web application interface for Group Management. At the top, there are two tabs: "Presentation Management" and "User Management". Below the "User Management" tab, there are three sub-tabs: "Users", "Groups", and "Assignment". The "Groups" sub-tab is currently selected. Below the sub-tabs, there is a "Select Group" label and a dropdown menu with "Select Group" as the selected option. Below this, there are two input fields: "Group Name" and "Group Description". The "Group Description" field is a larger text area with a vertical scrollbar. At the bottom of the form, there are four buttons: "Create", "Update", "Delete", and "Clear".

Figure 12. Group Management

5.2.2.7 Flow Chart

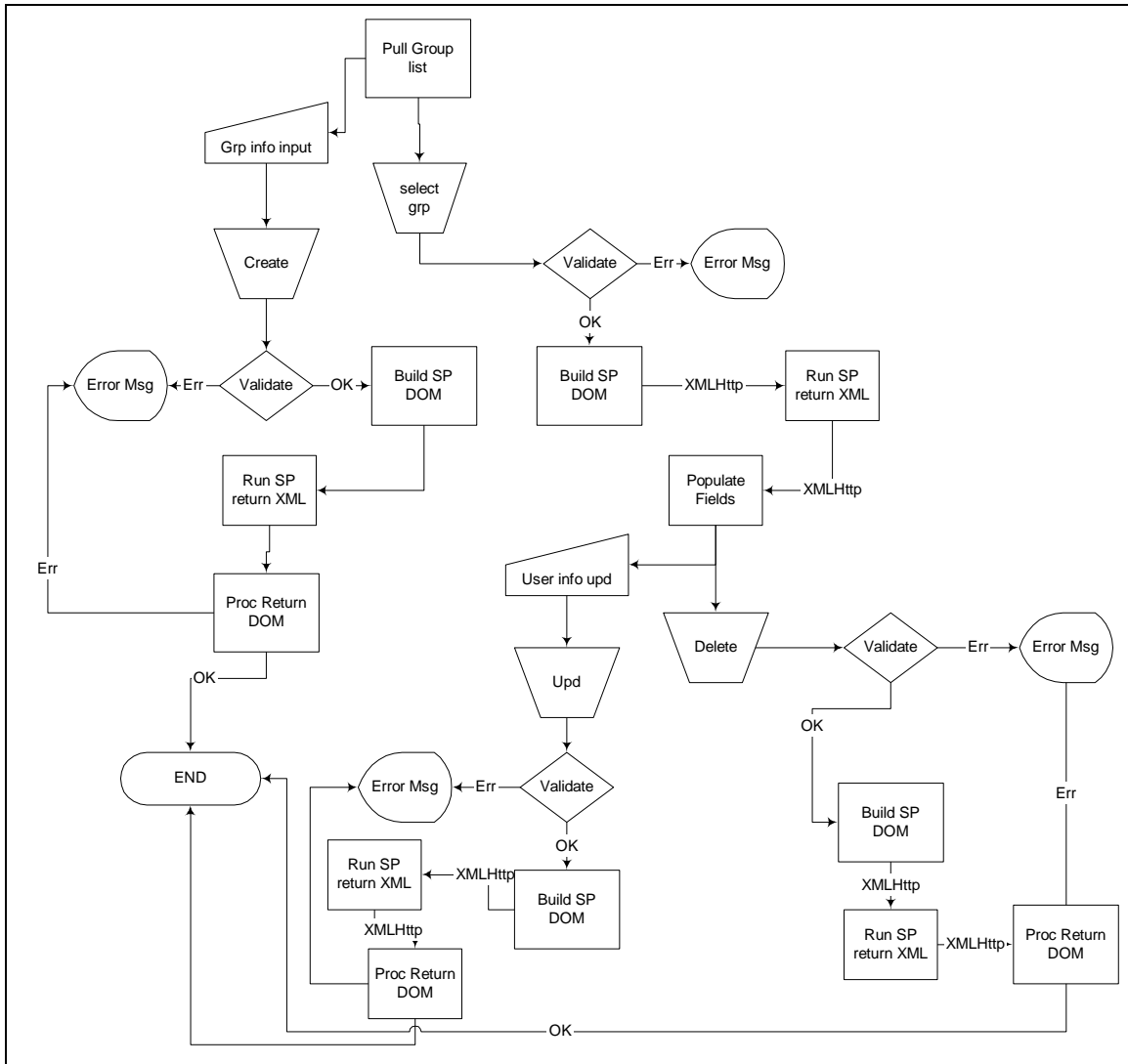


Figure 13. Group Management Flow Chart

5.2.3 User Group Association

5.2.3.1 Page Function

The User Group Association page allows users to be assigned to groups. Group assignment is what ultimately will give access to presentations for the user.

5.2.3.2 Data Source

- NimsPres Database

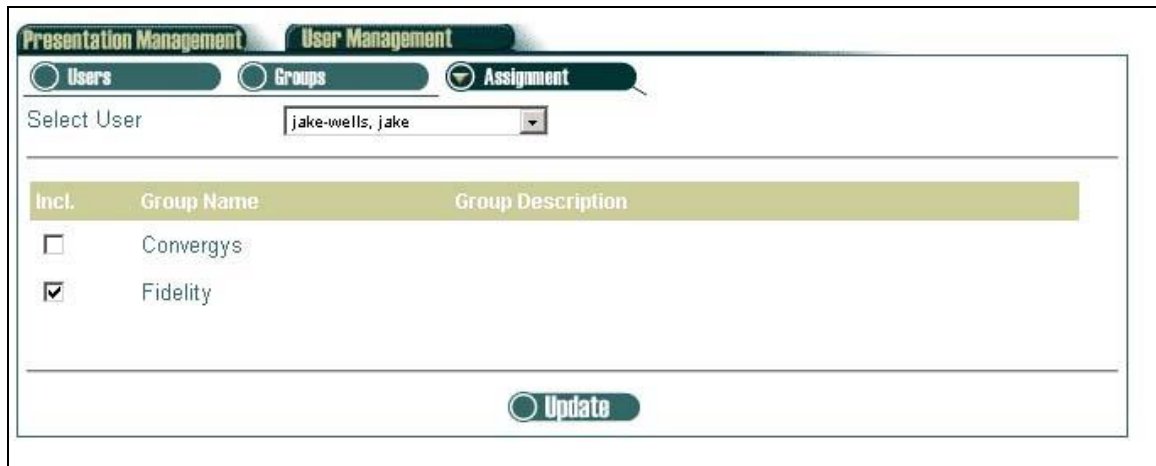
5.2.3.3 Data Elements

- User List Box
- Group Name
- Group Description
- User Group Association checkbox

5.2.3.4 Available Actions

- Update – updates the associations between the user and group

5.2.3.5 Page Layout



The screenshot displays a web application interface for user management. At the top, there are two tabs: "Presentation Management" and "User Management". Under "User Management", there are three sub-tabs: "Users", "Groups", and "Assignment". The "Assignment" tab is currently selected. Below the tabs, there is a "Select User" dropdown menu with the text "jake-wells, jake" and a downward arrow. Below this is a table with the following structure:

Incl.	Group Name	Group Description
<input type="checkbox"/>	Convergys	
<input checked="" type="checkbox"/>	Fidelity	

At the bottom of the interface, there is an "Update" button.

Figure 14. User Group Association

5.2.3.6 Flow Chart

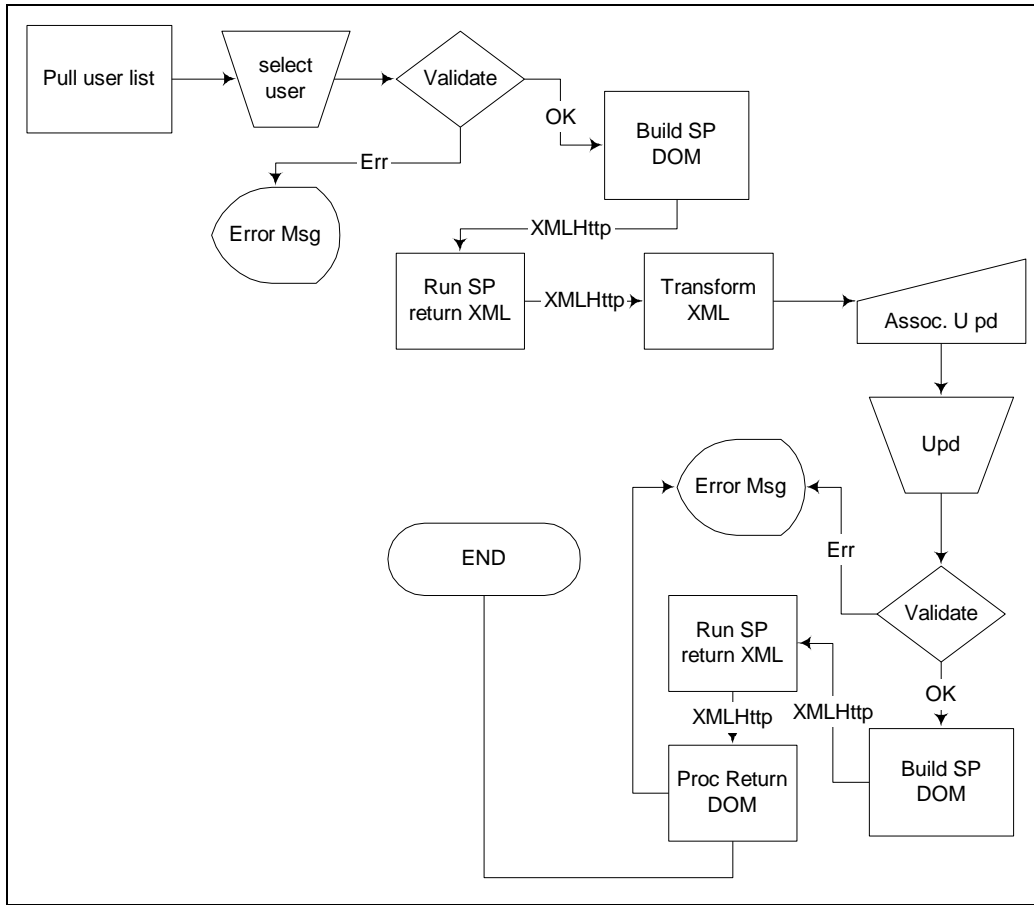


Figure 15. User Group Association Flow Chart

5.3 Presentation Management

The Presentation Management section is the heart of the application. It provides the ability to the Nims User to create, edit, and delete presentations, edit presentation content, and associate presentations with a presentation group. These functions are provided by the user interface through the ASP pages detailed in the following sections.

5.3.1 Presentation Creation

5.3.1.1 Page Function

This is the "launch page" for creating a presentation. A user will choose an existing presentation for edit or deletion, or fill in the textboxes and choose a style for the creation of a new presentation. Update and Delete will not be enabled until an existing presentation is selected. If a user selects a presentation and clicks Create, then a new presentation will be created using the selected presentation as a template.

5.3.1.2 Data Source

- NimsPres Database

5.3.1.3 Data Elements

- Presentation List Box – presents all available presentations
- Disabled Checkbox
- Style List Box
- Presentation Name Text Box
- Presentation Description

5.3.1.4 Available Actions

- Create – Creates a presentation from the master content
- Update – Updates Presentation Name, description, disabled flag
- Delete – Deletes presentation from the database
- Clear – Clears form fields

5.3.1.5 Validation

- User must select style
- User must enter name

5.3.1.6 Page Layout

The screenshot shows a web application interface for creating a presentation. At the top, there are two tabs: "Presentation Management" (active) and "User Management". Below the tabs are four radio buttons: "Presentations" (selected), "Scenes", "Assignment", and "Download". The main form area contains the following elements:

- A "Select Presentation" label followed by a dropdown menu with "Select Presentation" as the selected option.
- A horizontal line separator.
- Two checkboxes: "Disabled" (checked) and "Public" (unchecked).
- A "Style" label followed by a dropdown menu with "Select Style" as the selected option.
- A "Presentation Name" label followed by a text input field.
- A "Presentation Description" label followed by a text area with a vertical scrollbar.
- A horizontal line separator.
- Four radio buttons at the bottom: "Create" (selected), "Update", "Delete", and "Clear".

Figure 16. Presentation Creation

5.3.1.7 Flow Chart

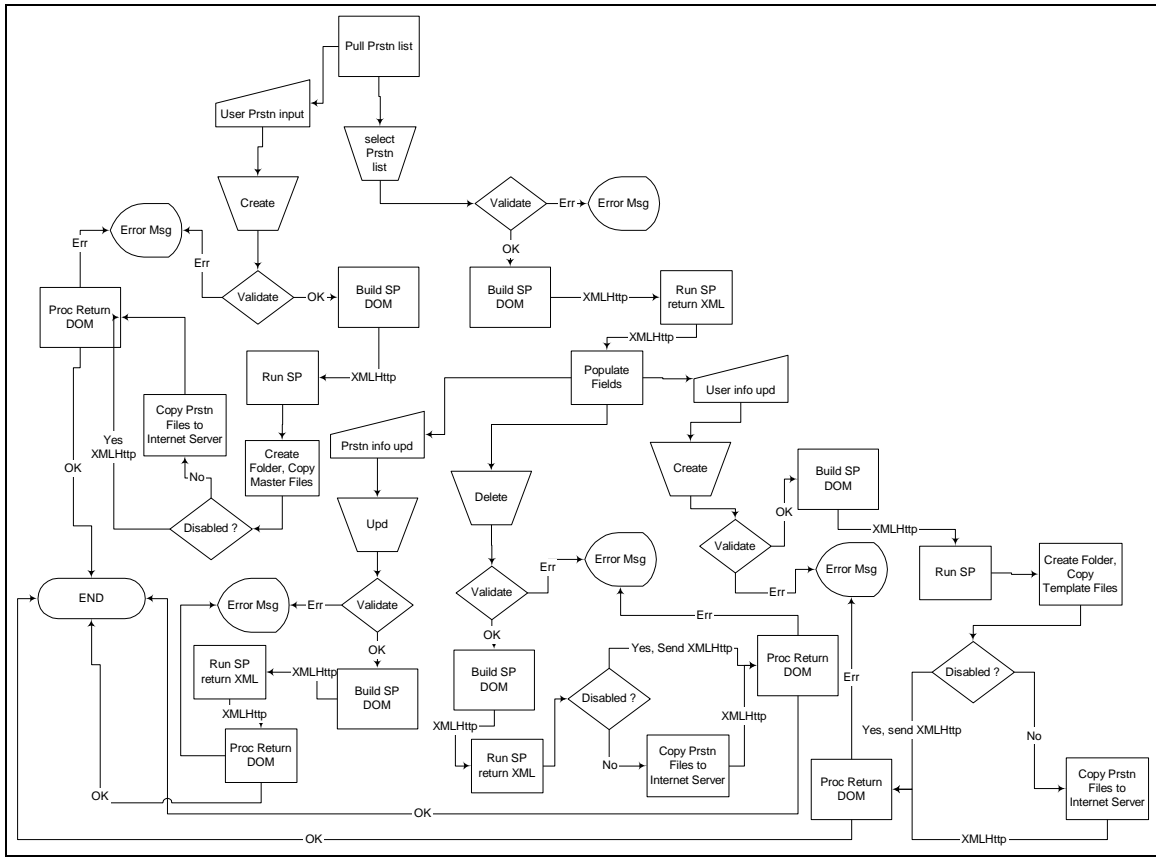


Figure 17. Presentation Creation Flow Chart

5.3.2 Presentation Association

5.3.2.1 Page Function

This page is used to assign presentations to groups. The user will select the presentation from the drop down menu, then check all groups that he would like to give rights. After clicking the Update button, all assignments will be removed, then for each of the checked security group, a new assignment will be added. The user can also check the "Make the presentation public" checkbox, which will give rights to all users in the system. Making the presentation public will remove all assignments and mark the Public flag in the presentation table true.

5.3.2.2 Data Source

- NimsPres Database

5.3.2.3 Data Elements

- Presentation List Box
- Group Name
- Group Description
- Presentation/Group Association CheckBox

5.3.2.4 Available Actions

- Update – updates the presentation group associations

5.3.2.5 Page Layout



The screenshot displays a web application interface for managing presentations. At the top, there are two tabs: "Presentation Management" (active) and "User Management". Below the tabs are four buttons: "Presentations", "Scenes", "Assignment", and "Download". The "Presentations" button is selected. Below the buttons, there is a "Select Presentation" dropdown menu with "testing" selected. Below the dropdown is a table with three columns: "Incl.", "Group Name", and "Group Description". The table contains two rows: "Convergys" and "Fidelity". The "Fidelity" row has a checked checkbox in the "Incl." column. Below the table is an "Update" button.

Incl.	Group Name	Group Description
<input type="checkbox"/>	Convergys	
<input checked="" type="checkbox"/>	Fidelity	

Figure 18. Presentation Association

5.3.2.6 Flow Chart

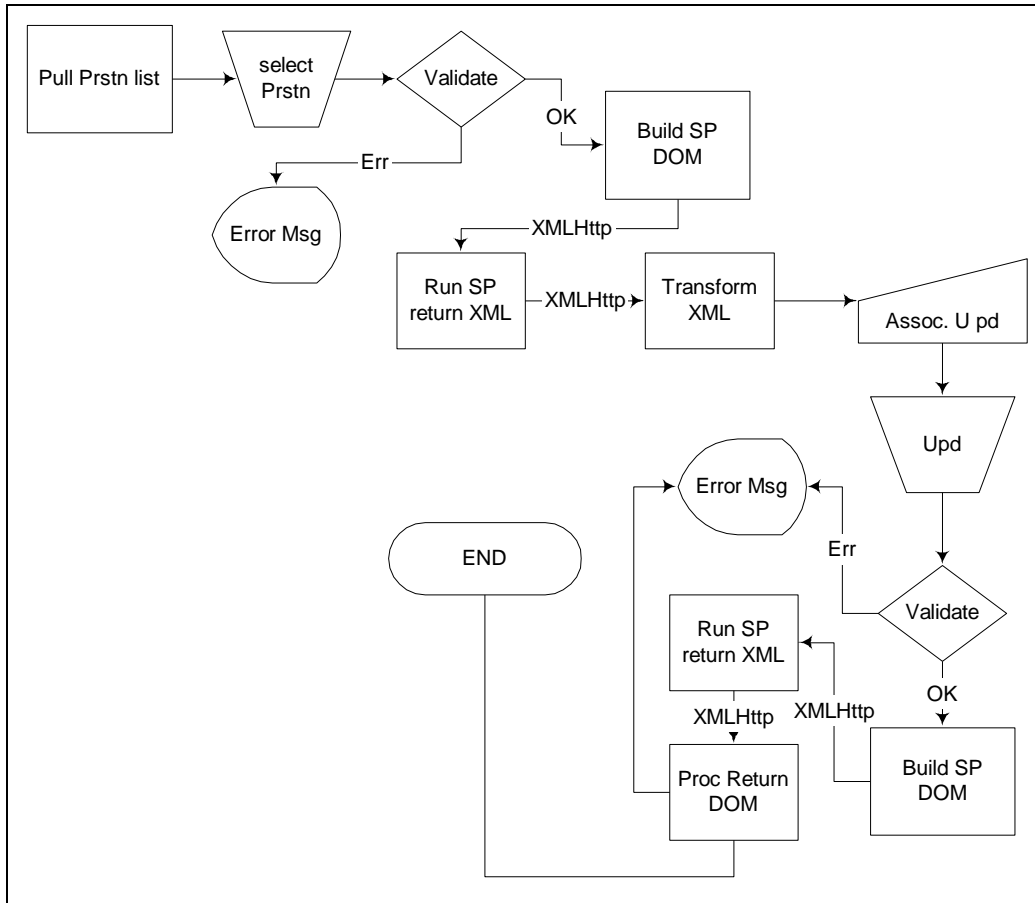


Figure 19. Group Presentation Association Flow Chart

5.3.3 Content Edit

5.3.3.1 Page Function

This page will allow the user to select the scenes for inclusion in the presentation, edit master content (content used through the presentation), and scene content. The user will be presented with a list of scenes that are associated with a given style and will check the scenes that he wants to include. The page also includes a section for user input of Master Content. Master content will be used on more than one scene (i.e. Presentation Title, Date, Company Name). Scenes can also include textboxes that will allow the user to edit or insert content.

5.3.3.2 Data Source

- XML Dom

5.3.3.3 Data Elements

- Presentation List Box
- Additional Content found on this page is built based on the contents of the presentation's XML Dom (see Figure 20). The Dom will be transformed using an XSL stylesheet, which is displayed, in the user interface.

```
<?xml version="1.0"?>
<xml>
<mastercontent>
  <text id="strPresTitle" value="Senior Design III"
    friend_nm="Presentation Title" datatype="textbox"/>
</mastercontent>
<scenes id="test">
  <scene id="003" desc="Overview of the services that Nims can provide."
    thumbnail="/master/nimsintrnt/003.gif" enabled="1">
    <content>
      <text id="strSceneTitle" value="capabilities"
        friend_nm="Scene Title" datatype="textbox"
        scene="003"/>
    </content>
  </scene>
</scenes>
</xml>
```

Figure 20. XML Format

5.3.3.4 Available Actions

- Update – updates the XML Dom
- Preview – updates the XML Dom, opens a separate window with the contents of the presentation

5.3.3.5 Page Layout

Title	Value
Presentation Title	<input type="text" value="Presentation Title"/>
Presentation Date	<input type="text" value="Presentation Date"/>
Presenter	<input type="text" value="Presenter"/>
Company Name	<input type="text" value="Company Name"/>

Figure 21. Content Edit: Master Content

Scene Description: Overview of the services that Nims can provide.

Title	Value
Scene Title	<input type="text" value="capabilities"/>
Bullet 1	<input type="text" value="Professional Staffing"/>
Bullet 2	<input type="text" value="Managed Services"/>
Bullet 3	<input type="text" value="Project Management"/>
Bullet 4	<input type="text" value="Educational Services"/>
Bullet 5	<input type="text"/>
Bullet 6	<input type="text"/>
Bullet 7	<input type="text"/>
Bullet 8	<input type="text"/>
Bullet 9	<input type="text"/>

capabilities

- Professional Staffing
- Managed Services
- Project Management
- Educational Services

Company Name
Presentation Date

Figure 22. Content Edit: Scene Content

5.3.3.6 Flow Chart

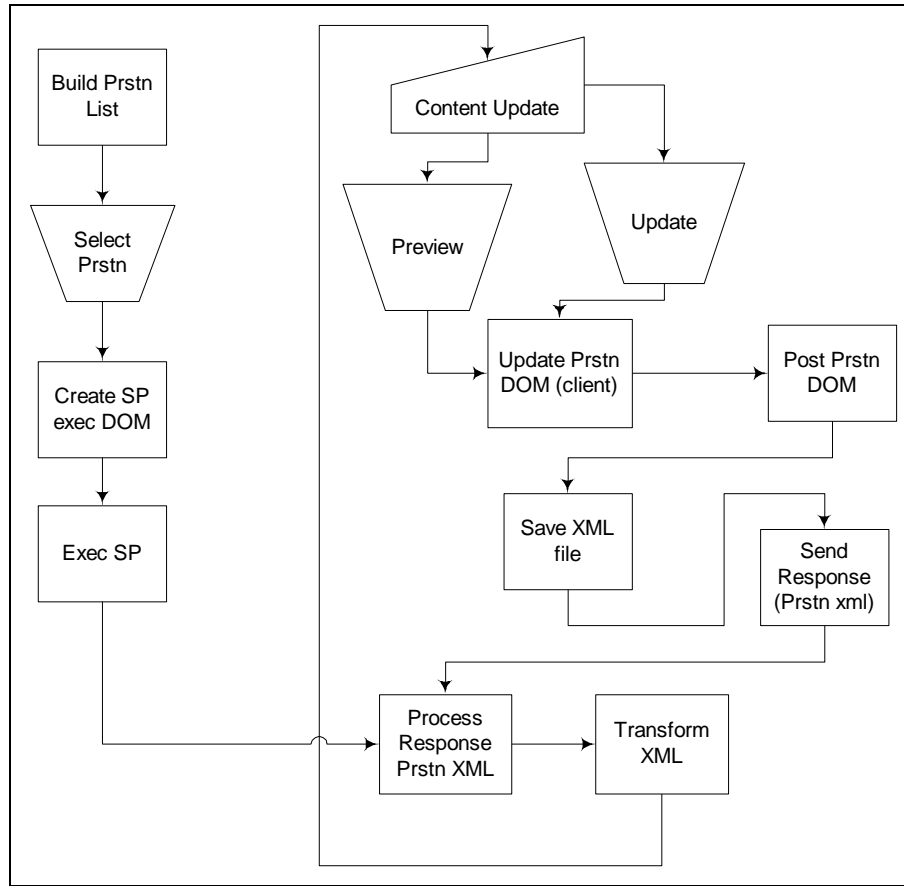


Figure 23. Content Edit Flow Chart

5.3.4 Presentation Download

5.3.4.1 Page Function

The Presentation Download page allows the Nims User to pull down all the necessary files associated with a generated presentation to a laptop or some other machine. This feature allows the presentations not be limited to publishing on the Internet. In the event a customer does not have Internet access, a Nims user can download all the applicable files to a laptop and present the presentation in a traditional manner.

5.3.4.2 Data Source

- NimsPres Database

5.3.4.3 Data Elements

- Flash File (*.swf)
- Content File (presXML.xml)

5.3.4.4 Page Layout



Figure 24. Presentation Download

5.3.4.5 Flow Chart

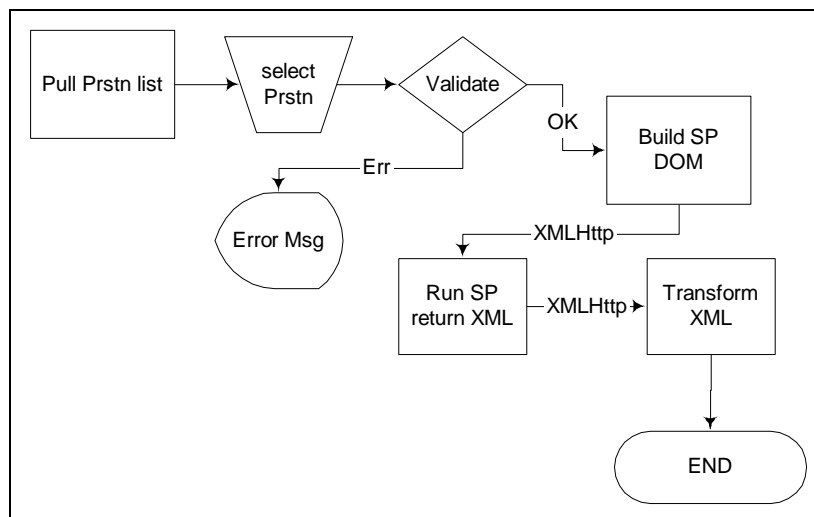


Figure 25. Presentation Download Flow Chart

5.4 Client Presentation Site

The Client Presentation site is the client-facing portion of NPM. This portion of the application will be hosted on the Nims Internet Site. The contents of the site are straightforward and will support various browser types. Presentations will be available to the client at all times and by being located on the Internet Site, additional information is readily available.

5.4.1 Client Login Page

5.4.1.1 Page Function

This page will validate the user's user name and password. The gathered user name will then be used to pull all presentations associated with the `usr_id`

5.4.1.2 Data Source

- NimsPres Database

5.4.1.3 Data Elements

- User ID
- Password

5.4.1.4 Available Actions

- Login - Validates the user against the NimsPres Database

5.4.1.5 Validation

- User ID must be entered

5.4.1.6 Page Layout

Nims Client Presentations

User ID

Password

To view the presentations assigned to you, please enter your user id and password.

Figure 26. Client Login Page

5.4.1.7 Flow Chart

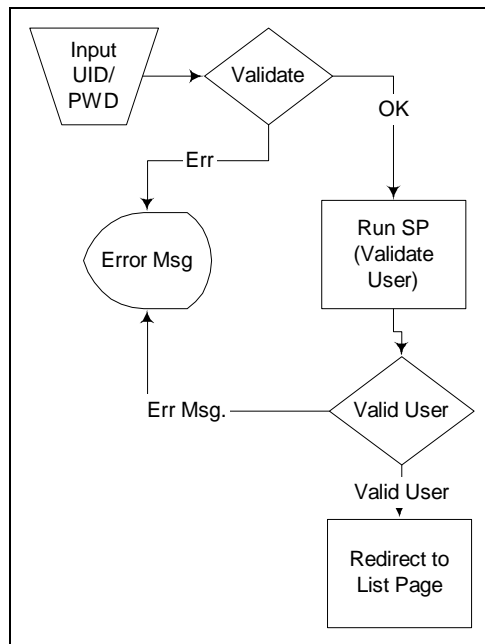


Figure 27. Login Flow Chart

5.4.2 Client Presentation List Page

5.4.2.1 Page Function

This page will give access to the presentations that have been created by the Nims User and have been assigned to the user based on group membership or the

presentations public flag. When a presentation name link is clicked, a new window will be invoked. The contents of the new window will be the applicable presentation.

5.4.2.2 Data Source

- NimsPres Database

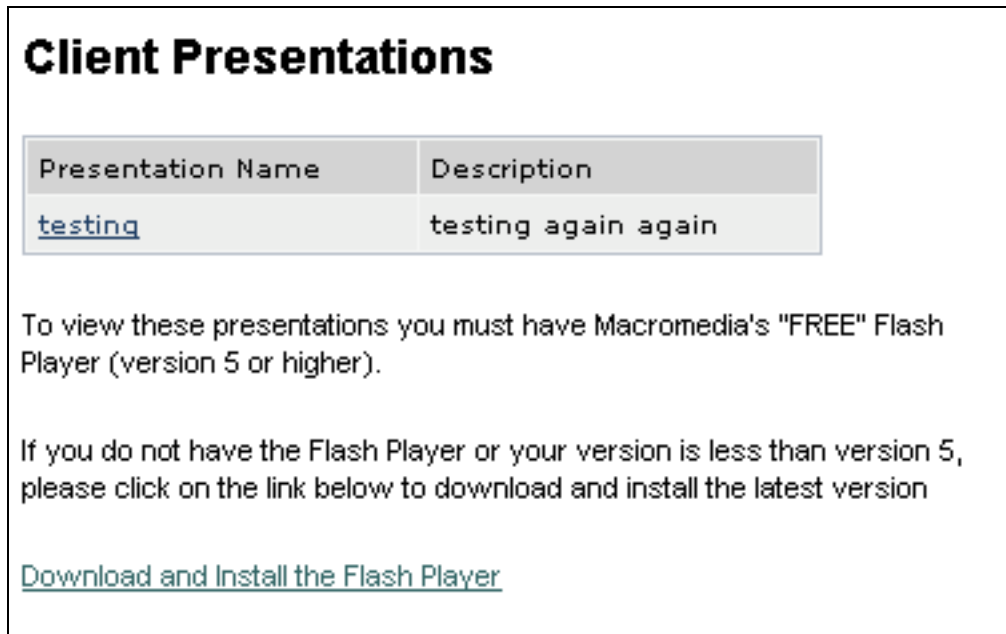
5.4.2.3 Data Elements

- Presentation Links to SWF files

5.4.2.4 Available Actions

- Presentation Links – Opens separate window with the contents of the presentation

5.4.2.5 Page Layout



Client Presentations

Presentation Name	Description
testing	testing again again

To view these presentations you must have Macromedia's "FREE" Flash Player (version 5 or higher).

If you do not have the Flash Player or your version is less than version 5, please click on the link below to download and install the latest version

[Download and Install the Flash Player](#)

Figure 28. Presentation List Page

5.4.2.6 Flow Chart

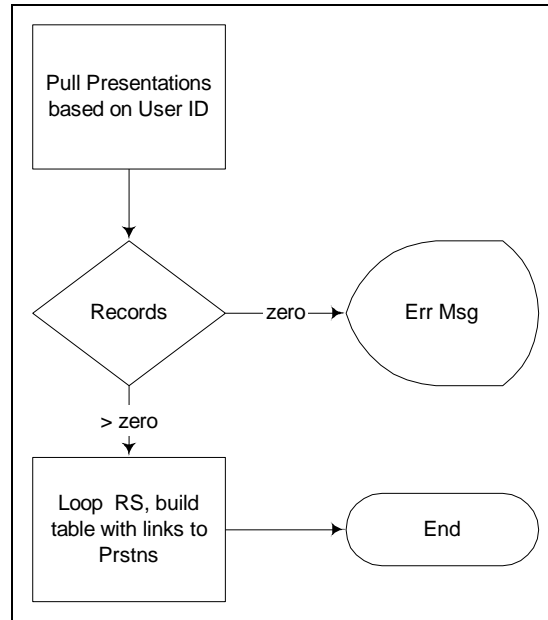


Figure 29. Presentation List Page Flow Chart

5.4.3 Client Change Password Page

5.4.3.1 Page Function

This page will allow the client user to change his password. The page will be invoked if the user's password is set to null, which will be the situation when a user is first set up, or when the Nims user resets the client user's password from the Reset Password page. When a user is first set up, the password will be empty and the system will force the user to set a password. This is a variation of the Change Password page that does not require the user to enter an old password.

5.4.3.2 Data Source

- NimsPres Database

5.4.3.3 Data Elements

- UserID
- Old Password

- New Password
- Confirm Password

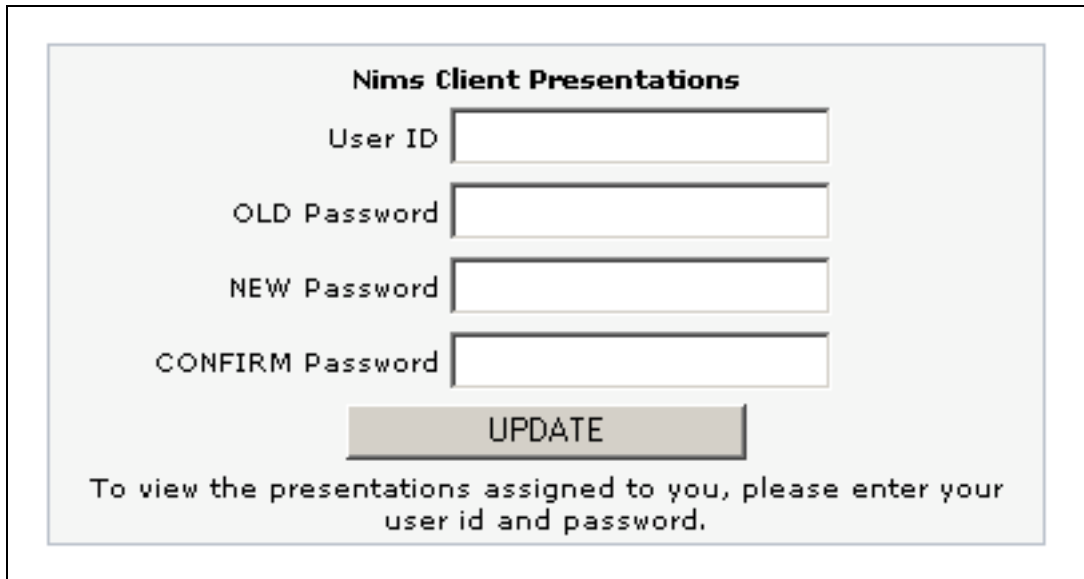
5.4.3.4 Available Actions

- Update – validates old password and changes the user’s password to the new password

5.4.3.5 Validation

- All fields have content
- Old password matches the password in the database

5.4.3.6 Page Layouts



The screenshot shows a web form titled "Nims Client Presentations". It contains four input fields: "User ID", "OLD Password", "NEW Password", and "CONFIRM Password". Below these fields is a button labeled "UPDATE". At the bottom of the form, there is a message: "To view the presentations assigned to you, please enter your user id and password."

Figure 30. Client Change Password Page

You need to set up a password

Nims Client Presentations

User ID

NEW Password

CONFIRM Password

Figure 31. Set Password Page

5.4.3.7 Flow Chart

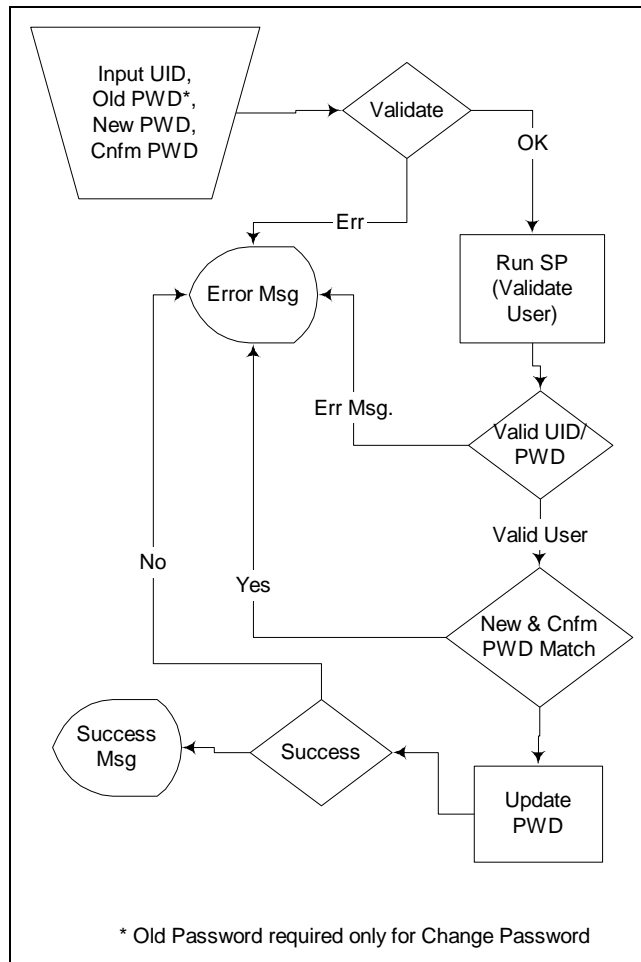


Figure 32. Change and Set Password Flow Chart

6. Conclusions and Recommendation

Although I am pleased with the present state of this project, I believe that it could be a much better product if more time were given to design and development. This project is much bigger than one person is really capable of completing in the time that I have had to work on the project. Additionally, it has become apparent to me that a developer gets “tunnel vision”, and a project like this requires more than one person who is intimately involved in the project to get the best design.

The Flash development alone needs significantly more time. The design as a whole is sound, but individual sections could use some rework. Specifically, a significant amount of time should be spent on the design of a generic scene, which could be included multiple times within the presentation. As the application exists today, a scene must exist for every node in the XML Dom, which increases the size of the SWF significantly. Also, time needs to be dedicated to designing a new module which would allow the user to specify the order of the presentation, as today, the user is constrained to following the existing order of the SWF file.

Minor changes have occurred over the course of the three quarters that I have worked on this project. In the initial proposal, I had proposed two additional Middleware objects, one to deal with user management and another to deal with the creation of XML. The user management dll was eliminated with the requirement that client users be maintained in the database. So all user management could be performed through ASP and stored procedures. The XML dll was eliminated as I became more acquainted with XML, and the discovery of the XMLHTTP class. It was apparent to me that I client side JavaScript and server side vbscript was more

pliable to me, and it would not be beneficial to try to encapsulate the functionality in a dll. Other changes that have occurred include the combination of the “Create Presentation” and “Select Style” pages, and eliminating the ability to assign presentations directly to users. As the prototype came together, these items did not seem to add value to the overall project.

The development of this project has been a unique experience in two regards. First, I can lay claim to this project as I conceived the idea, designed the product, and performed all the development. In no other project have I had complete involvement in all aspects of the project. This fact, though, is not an ideal situation, as others should be involved in all stages to assure that the project is viewed critically, and other ideas can be introduced into the design and development. Second, this project has forced me to explore new technologies. During the first quarter, I had limited experience with XML and Flash, and this lack of experience is the main reason I included these items in the design. In the year between Senior Design I and II, I gained experience with XML and XSL that greatly improved the design of this project, but I still had little experience with Flash and the ActiveScripting language. So I have been able to spend significantly more time learning aspects of Flash that I would not have experienced if I had not been exposed to XML development during the period between the first two quarters. The development of this project has been a positive learning experience, although at times, very frustrating.

Appendix A. Selection from Page Building (default.asp)

Private Function ShowScene()

```
Dim tmpDetails
Dim intPresid

tmpDetails = tmpDetails & BuildTableHeader("Select Scenes", 650) & Chr(10)
tmpDetails = tmpDetails & "<table width=650 border=0 cellpadding=2
    cellspacing=0 class=table-gold>" & Chr(10)
tmpDetails = tmpDetails & "<tr valign=top>" & Chr(10)
tmpDetails = tmpDetails & "<td width=150>Select Presentation" & Chr(10)
tmpDetails = tmpDetails & "</td>" & Chr(10)
tmpDetails = tmpDetails & "<td width=500>" & Chr(10)

'Build Presentation ListBox
tmpDetails = tmpDetails & BuildPrstnDropDown(-1) & Chr(10)
tmpDetails = tmpDetails & "</td>" & Chr(10)
tmpDetails = tmpDetails & "</tr>" & Chr(10)
tmpDetails = tmpDetails & InsertTableSpacer(2)
tmpDetails = tmpDetails & "<tr valign=top>" & Chr(10)
tmpDetails = tmpDetails & "<td colspan=2 id=PageData name=PageData>"
tmpDetails = tmpDetails & "<STRONG>Please select Presentation from the list
above.</STRONG>" & Chr(10)
tmpDetails = tmpDetails & "</td>" & Chr(10)
tmpDetails = tmpDetails & "</tr>" & Chr(10)
tmpDetails = tmpDetails & "</table>" & Chr(10)
tmpDetails = tmpDetails & "<SCRIPT>" & Chr(10)

If Request.QueryString("presid") = "" Then
    intPresid = -1
Else
    intPresid = Cint(Request.QueryString("presid"))
End If

tmpDetails = tmpDetails & "var presID = " & intPresid & Chr(10)
tmpDetails = tmpDetails & "if(presID != -1)" & Chr(10)
tmpDetails = tmpDetails & "{" & Chr(10)
tmpDetails = tmpDetails & "    loaddata()" & Chr(10)
tmpDetails = tmpDetails & "}" & Chr(10)
```

```

tmpDetails = tmpDetails & "</SCRIPT>" & Chr(10)

%>
<SCRIPT>

function loaddata(location)
{
xmlDom = new ActiveXObject('MSXML2.DomDocument')
xslDom = new ActiveXObject('MSXML2.DomDocument')
xslDom.async = false
xmlDom.load(location + "/presXML.xml")
xslDom.load("/_xsl/scenes.xsl")
document.all('PageData').innerHTML = xmlDom.transformNode(xslDom);
}

function Next()
{
curPageND = xslDom.selectSingleNode('/xsl:stylesheet/xsl:param')
curPageND.attributes.getNamedItem('select').text =
    parseInt(curPageND.attributes(1).text) + 1
document.all('PageData').innerHTML = xmlDom.transformNode(xslDom);
}

function CheckBoxToggle()
{
scenesNode = xmlDom.selectSingleNode('//scenes')
for(var i=0;i < scenesNode.childNodes.length-1;i++)
{
thisScene = scenesNode.childNodes(i)
thisId = thisScene.attributes.getNamedItem('id').text
if(thisId==window.event.srcElement.id)
{
if(window.event.srcElement.checked==false)
{
thisScene.attributes.getNamedItem('enabled').text = 0
}
else
{
thisScene.attributes.getNamedItem('enabled').text = 1
}
}
}
}

```

```

        }
    }

}

function UpdateXML()
{
    MasterTextBoxUpdate()

    ScenesTextBoxUpdate()

    Save()
}

function MasterTextBoxUpdate()
{
    scenesNode = xmlDoc.selectSingleNode('//mastercontent')

    for(var i=0;i < scenesNode.childNodes.length;i++)
    {
        thisScene = scenesNode.childNodes(i)

        thisId = thisScene.attributes.getNamedItem('id').text

        thisScene.attributes.getNamedItem('value').text =
document.all(thisId).value
    }
}

function ScenesTextBoxUpdate()
{
    scenesNode = xmlDoc.selectSingleNode('//scenes')

    for(var i=0;i < scenesNode.childNodes.length;i++)
    {
        thisScene = scenesNode.childNodes(i)

        thisSceneID = thisScene.attributes.getNamedItem("id").text

        contentNode = thisScene.selectSingleNode("content")

        for(var k=0; k < contentNode.childNodes.length; k++)
        {
            thisContent = contentNode.childNodes(k)

            thisContentId = thisContent.attributes.getNamedItem('id').text

            thisContentSceneId = thisContent.attributes.getNamedItem('scene').text

            thisContent.attributes.getNamedItem("value").text =

```

```

        document.all(thisContentId+"-"+thisContentSceneIdJ).value
    }
}
}
function Save()
{
    xmlDomNew = new ActiveXObject('MSXML2.DomDocument')
    xmlHttp = new ActiveXObject('MSXML2.XMLHTTP')

    xmlHttp.Open('POST',
        'Actions.asp?activity=saveXML&presid='+document.all('lstPrstn').value,
        false);
    xmlHttp.send(xmlDom.xml)
    xmlDomNew.load(xmlHttp.responseXML)
    var errorND = xmlDomNew.selectSingleNode("//Error")
    if( errorND == null)
    {
        xmlDom.load(xmlHttp.responseXML)
    }
    else
    {
        alert(errorND.text)
    }
}
function Preview()
{
    openSingleWindow(preslocation+"/"+presfilename)
}
</SCRIPT>
<%
ShowScene = tmpDetails
End Function

```

Appendix B. Selections from Client Processing (default.asp)

```
...
case 'pres':
    if(document.all('txtPres_nm').value == '')
    {
        errMsg = 'A presentation name is required\n'
    }
    if
    {
        document.all('lstStyle').value == -1)
    {
        errMsg = errMsg + 'A style selection is required\n'
    }
    if(errMsg != '')
    {
        alert(errMsg)
    }
    else
    {
        sp = "exec p_Prstn_ins "
        sp = sp + "'" + document.all('txtPres_nm').value + "',"
        sp = sp + "'" + document.all('txtPres_desc').value + "',"
        sp = sp + document.all('lstStyle').value + ","

        if(document.all('chkPublic').checked == false)
        {
            sp = sp + "0," // document.all('chkDisabled').value
        }
        else
        {
            sp = sp + "1," // document.all('chkDisabled').value
        }

        if(document.all('chkDisabled').checked == false)
        {
```



```

        sp = sp + "0" // document.all('chkDisabled').value
    }
else
    {
        sp = sp + "1" // document.all('chkDisabled').value
    }

objND = objXMLDom.createElement("exec")
objND.text = sp
objND.setAttribute("createfrom", document.all('txtFile_Loc').value)
objND.setAttribute("createfromfilenm", document.all('txtFile_nm').value)
objXMLDom.firstChild.appendChild(objND)
objXMLDom.load(postXML("Actions.asp?activity=createpres", objXMLDom))

if(objXMLDom.selectSingleNode("//error") != null)
    {
        alert('An Error occured while attempting to create your
presentation\n'+ objXMLDom.selectSingleNode("//error").text)
    }
else
    {
        alert('The presentation was created successfully.')
    }
}
break;
}
break;

```

...

Appendix C.

Selections from Server Processing (Actions.asp)

Private Sub CreatePres()

```
On Error Resume Next

Dim objXML, objCFNode, objReturnXML, objND
Dim objSQL, sp,rs
Dim objFSO, strTemplateLoc, strTemplateFileNm
Dim NodeLength, execNode, strNewLoc, strMasterLoc, strMasterLoc2

Response.ContentType = "text/xml"

Response.Buffer = False

Set objXML = Server.CreateObject("MSXML2.DomDocument")
Set objSQL = server.CreateObject("DB_Connector.SQL_Utility")
Set objReturnXML = Server.CreateObject("MSXML2.DomDocument")
Set objFSO = Server.CreateObject("Scripting.FileSystemObject")

objXML.load(Request)

Set execNode = objXML.firstChild.selectSingleNode("exec")

sp = execNode.text

Set rs = objSQL.RunSQLStringReturnRS(sp)

If err.number <> 0 Then

    Response.Write("<xml><error>" & err.description & "</error></xml>")

Else

    objReturnXML.appendChild(objReturnXML.createElement("xml"))

    Set objND = objReturnXML.createElement("id")

    objND.text = rs("id")

    objReturnXML.firstChild.appendChild(objND)

    Set objND = objReturnXML.createElement("location")

    objND.text = rs("location")

    objReturnXML.firstChild.appendChild(objND)

    Set objND = objReturnXML.createElement("file_nm")

    objND.text = rs("file_nm")

    objReturnXML.firstChild.appendChild(objND)

    strNewLoc = Request.ServerVariables("APPL_PHYSICAL_PATH") &
Right(replace(rs("location"),"/","\\"), Len(rs("location"))-1)

    objFSO.CreateFolder strNewLoc

    strMasterLoc = Request.ServerVariables("APPL_PHYSICAL_PATH") &
```

```

Right(Replace(rs("masterloc"),"/","\\"), Len(rs("masterloc"))-1)

    strMasterLoc2 = Request.ServerVariables("APPL_PHYSICAL_PATH") & "master"
    strTemplateLoc = execNode.attributes.getNamedItem("createfrom").Text
    strTemplateFileNm = execNode.attributes.getNamedItem("createfromfilenm").Text
    If strTemplateLoc = "" Then
        'copy the SWF file
        objFSO.CopyFile strMasterLoc & "\" & rs("master_file_nm"), strNewLoc & "\" &
rs("file_nm")
        'copy the xml file
        objFSO.CopyFile strMasterLoc & "\presXML.xml", strNewLoc & "\presXML.xml"
    Else
        strTemplateLoc = Replace(strTemplateLoc, "/", "\\")
        strTemplateLoc = Right(strTemplateLoc, Len(strTemplateLoc) - 1)
        strTemplateLoc = Request.ServerVariables("APPL_PHYSICAL_PATH") & strTemplateLoc
        'copy the SWF file
        objFSO.CopyFile strTemplateLoc & "\" & strTemplateFileNm, strNewLoc & "\" &
rs("file_nm")
        'copy the xml file
        objFSO.CopyFile strTemplateLoc & "\presXML.xml", strNewLoc & "\presXML.xml"
    End If
    If err.number <> 0 Then
        Response.Write("<xml><error>" & err.description & "</error></xml>")
    Else
        Response.Write(objReturnXML.xml)
    End If
End If

Set objXML = Nothing
Set objSQL = Nothing
Set objReturnXML = Nothing
Set objFSO = Nothing
Set objCFNode = Nothing
Set objND = Nothing
Set execNode = Nothing
Set strTemplateLoc = Nothing

Response.End

End Sub

```

Appendix D. Stored Procedure Definitions

Group Presentation Association Delete

p_Grp_Prstn_del
@grp_id int = null,
@prstn_id int = null

Group Presentation Association Insert

p_Grp_Prstn_ins
@grp_id int,
@prstn_id int

Group Delete

p_Grp_del
@grp_id int

Group Inquiry

p_Grp_inq
@grp_id int

Group Insert

p_Grp_ins
@grp_nm varchar(25),
@grp_desc varchar(100)

Group Update

p_Grp_upd
@grp_nm varchar(25),
@grp_desc varchar(100),
@grp_id int

Group Fetch

p_Grps_fetch

Presentation Group Association Inquiry

p_Prstn_Grp_inq

Presentation Delete

p_Prstn_del
@prstn_id int

Presentation Inquiry

p_Prstn_inq

@prstn_id int

Presentation Insert

p_Prstn_ins
@prstn_nm varchar(25),
@prstn_desc varchar(100),
@style_id int,
@public_i bit,
@disabled_i bit

Presentation Update

p_Prstn_upd
@prstn_nm varchar(25),
@prstn_desc varchar(100),
@public_i bit,
@disabled_i bit,
@prstn_id int

Presentation Fetch

p_Prstns_fetch

Styles Fetch

p_Styles_fetch

User Group Association Delete

p_Usr_Grp_del
@grp_id int = null,
@usr_id varchar(10) = null

User Group Association Inquiry

p_Usr_Grp_inq
@usr_id varchar(10)

User Group Insert

p_Usr_Grp_ins
@grp_id int,
@usr_id varchar(12)

User Presentation Association Inquiry

p_Usr_Prstn_inq
@usr_id varchar(10)

User Authorization

p_Usr_auth
@input_uid varchar(10),
@input_pwd varchar(20)

User Delete

p_Usr_del
@usr_id varchar(10)

User Inquiry

p_Usr_inq
@usr_id nvarchar(10)

User Insert

p_Usr_ins
@usr_id varchar(10),
@fst_nm varchar(25),
@lst_nm varchar(25),
@email varchar(25),
@cmpy_nm varchar(50),
@title varchar(50),
@phone varchar(12),
@disabled bit

User Update

p_Usr_upd
@usr_id varchar(10),
@fst_nm varchar(25),
@lst_nm varchar(25),
@email varchar(25),
@cmpy_nm varchar(50),
@title varchar(50),
@phone varchar(12),
@disabled bit

Users Fetch

p_Usrs_fetch

User Password Update

p_usr_pwd_upd
@usr_id varchar(10),
@pwd varchar(50)

Appendix E.

Select Code from DB_Connector

```
Public Function RunSQLStringReturnXML(ByVal strSQL As String) As String

    Dim rs As ADODB.Recordset

    Dim strConnection As String

    Dim objStream As ADODB.Stream

    On Error GoTo ErrorHandler

    Set objStream = CreateObject("ADODB.Stream")

    ' Execute the SQL Server stored procedure sp_executesql with passed in SQL command and return the
Recordset.

    Set rs = RunSPReturnRS(g_cstr_STORED_PROC_EXECUTE_SQL, CONNECTION_STRING, _
g_clng_DEFAULT_RECORDS, Array(g_cstr_PARAM_SQL_STATEMENT, adVarChar, _
Len(strSQL), strSQL))

    'Save the rs as xml

    Call rs.Save(objStream, adPersistXML)

    'Open the xml file and return it as a string

    RunSQLStringReturnXML = objStream.ReadText

    Call objStream.Close

    Call rs.Close

    Set rs = Nothing

    Set objStream = Nothing

    CtxSetComplete

    Exit Function

ErrorHandler:

    Set rs = Nothing

    Set objStream = Nothing
```

```
' Call SetAbort on the Context object and raise an error  
Call App.LogEvent("DB_Connector.SQL_Utility - RunSQLStringReturnXML", vbLogEventTypeError)  
CtxRaiseError "DB_Connector.SQL_Utility", "RunSQLStringReturnXML"  
End Function
```

```
Public Function RunSPReturnRS(ByVal strSP As String, ByVal strConString As String, ByVal  
lngMaxRecords As Long, ParamArray varParams() As Variant) As ADODB.Recordset
```

```
On Error GoTo ErrorHandler
```

```
' Create the ADO objects
```

```
Dim objCmd As ADODB.Command
```

```
Dim objctx AsObjectContext
```

```
Set objctx = GetObjectContext
```

```
Set RunSPReturnRS = CtxCreateObject("ADODB.Recordset")
```

```
Set objCmd = CtxCreateObject("ADODB.Command")
```

```
If Not TypeName(objctx) = "Nothing" Then
```

```
    'log some error
```

```
End If
```

```
' Init the ADO objects & the stored proc parameters
```

```
With objCmd
```

```
    .ActiveConnection = strConString
```

```
    .CommandText = strSP
```

```
    .CommandType = adCmdStoredProc
```

```
    .CommandTimeout = 90
```

```

        .Properties.Item("Command Time Out").Value = 90
    End With

    CollectParams objCmd, varParams

    ' Execute the query for readonly
    With RunSPReturnRS
        .CursorLocation = adUseClient
        .MaxRecords = lngMaxRecords
        .Open objCmd, , adOpenForwardOnly, adLockReadOnly
    End With

    ' Disconnect the recordset
    Set objCmd = Nothing
    Set RunSPReturnRS.ActiveConnection = Nothing

    Exit Function

ErrorHandler:
    Call App.LogEvent("DB_Connector.SQL_Utility - RunSQLStringRS", vbLogEventTypeError)
    Set RunSPReturnRS = Nothing
    Set objCmd = Nothing
    RaiseError g_cstr_MOD_NAME_DATABASE, "RunSPReturnRS(" & strSP & ", ...)"
End Function

```


References

1. Hall, Branden. "XMLNitro 2.0".
<http://chattyfig.figleaf.com/~bhall/code/xmlnitro.as>. October 1, 2001.
2. Microsoft Corporation. "FMStocks_DB".
<http://msdn.microsoft.com/code/sample.asp?url=/MSDN-FILES/026/000/009/MsdnCompositeDoc.xml&frame=true>. June 30, 1999.