CM System

Setting Up the CM System Production Environment

Version 7.3

Printed on 12 June 2013

Copyright and Licensing Statement

Licenses and Source Code

Other Copyrights

This document was created using ***AuthorIT™, Total Document Creation*** (see AuthorIT Home - http://www.author-it.com).

Schema documentation was created using XMLSpy™.

Percussion Software
600 Unicorn Park Drive
Woburn, MA 01801 U.S.A.
781.438.9900
Internet E-Mail: technical_support@percussion.com
Website: http://www.percussion.com

Contents

Setting Up the Production Environment..7

Implementing a Tiered CM System Environment..8

Process Overview..9

Deploying a CM System Environment..11

Deployment Process..12

Updating the Datasource..13

Updating the Spring Configuration..15

Updating the CM System Servlet Configuration..16

Special Deployment Case..17

Updating the Deployed Environment..18

Implementing Security in the Production Environment..19

Defining Security Providers..20

Implementing a Directory Connection Security Provider..20

Implementing a Backend Table Security Provider..33

Implementing a Web Server Security Provider..36

Adding Users to CM System Roles..37

Updating Default Security Settings..38

Changing the Default Server Access Control List..39

Replacing the Default Users with New Users..39

Setting Up Publishing in the Production Environment..41

Publishing to a Mapped or Mounted Drive..42

Scheduling Publishing..45

Defining a Publishing Schedule..45

Copying a Scheduled Task..48

Defining a Task Notification..49

Quality Assurance Testing the Production Environment..53

Going Live..55

Adding Content to Your Production Content Management System..56

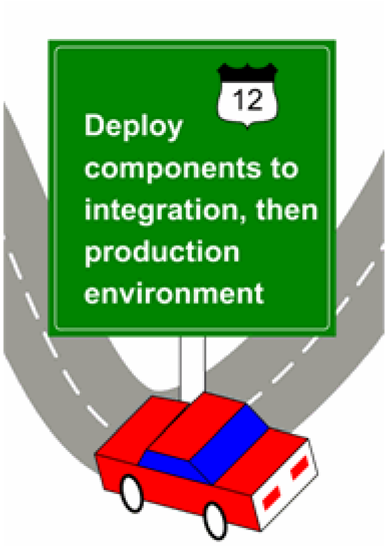
Training Users on CM System..57

Putting CM System into Production..58

Index..59

CHAPTER 1

Setting Up the Production Environment


12
Deploy components to integration, then production environment

Setting up the production environment is the final phase of CM System implementation. In this phase, you set up a production CM System server, and deploy your implementation from the development environment to the production environment. Once you have completed the deployment, you can begin adding your content to the production server, test it, and go live. Setting up production takes place after you have completed implementation of the CM System objects specified during your modeling and design process.

In many cases, customers also set up additional servers for staging or quality assurance testing. These servers should be exact replicas of the production server. Use the same procedures to set up those servers.

Before performing the tasks detailed in this document, you should have read and completed the tasks described in the *CM System Implementation Guide* (including any prerequisite documents and tasks for that manual). If your development plan called for special implementations (such as WebDAV, internationalization and localization, or Web services), you should have read the appropriate documents and completed those implementation tasks as well.

Implementing a Tiered **CM System Environment**

Percussion Software strongly recommends the use of a multi-tiered environment. Your CM System environment should consist of at least two, and preferably three tiers:

- Development tier

  The development tier is the environment in which you implement modifications to your system and develop new implementations.

- Integration tier

  The integration tier is an intermediate tier in which you merge design elements developed on different development servers to ensure that they function together correctly.

- Production tier

  The production tier is the CM System server that your end users will access to create and maintain live content.

Separating these tiers offers a variety of benefits. As you continue to develop your CM System implementation, you do not want to expose new CM System design elements to your end users until you have completed development of these elements and validated them. Newly developed elements can also introduce instability into the system, which you do not want to expose to the production server. Implementing a production environment separate from your development environment avoids these problems.

Implementing an integration tier adds an additional layer of safety to your system. If you have multiple developers working on your CM System implementation, you should always deploy new design elements to this tier first to ensure that they work together as intended. Once you have validated that new elements function correctly in this environment, you can deploy them to the production environment.

While this document emphasizes the implementation of a production environment, you can use the same processes and procedures to set up the integration environment. In fact, recommended practice after going live is to deploy any archives created in the development tier to the integration tier first so you can validate them before deploying them to the production tier.

Process Overview

The process of setting up the production environment involves the following phases:

1. ***Deploying a copy of the development server to the production environment***.
2. ***Implementing security in the production environment.*** (see "Implementing Security in the Production Environment" on page 19)
3. ***Implementing Publishing in the production environment.*** (see "Setting Up Publishing in the Production Environment" on page 41)
4. ***Quality Assurance Testing the production environment*** (on page 53).
5. ***Going live*** (on page 55).

This document includes a chapter describing each of these phases.

CHAPTER 2

Deploying a CM System Environment

Recommended practice for deploying a CM System environment from one tier to another is to copy a tree from a stable location and paste it to the target location; a dump of the stable database should be restored to the target database location.

In most cases, the environment being copied is a development environment. The development environment should be stable and functioning as desired before deploying it.

If you do not want to deploy development content to the new environment, that content should be purged from the development server before starting this process.

Deployment Process

To deploy a CM System environment:

1. If you are deploying to a pristine machine in a Windows environment, install CM System to the machine. If the implementation being deployed is based on FastForward, FastForward applications should be installed, but FastForward data should not be installed. All standard preparation and prerequisites apply to this installation.
2. Create a dump of the development database/schema.

   The specific process for creating the dump depends on the RDBMS. Consult the RDBMS documentation for instructions about this process.
3. Restore the dump to the production location.

   The specific process for restoring the dump depends on the RDBMS. Consult the RDBMS documentation for instructions about this process.

   In the database views shipped with CM System, the database owners and schema/database names are hard-coded. If the new schema/database has a different owner or name, you must manually update the views to match the owner and name in the new location.
4. Copy the development tree.
5. Paste the copied tree to the production machine.
6. Perform the following manual updates:

   a) Update the datasource. See ***Updating the Datasource*** (on page 13).

   b) If either the database property or origin property of the database configuration is different in the target environment than it is in the source environment, update the Spring configuration. See ***Updating the Spring Configuration*** (on page 15).

   c) If you install the tree in the new environment to a different path than the path in the development environment, update the CM System servlet configuration. See ***Updating the CM System Servlet Configuration*** (on page 16).

   For example, if your development server is installed to /user/local/Rhythmyx, but your production server is installed to /home/Production/CMS/Rhythmyx, you must modify the servlet configuration.

   If both servers are installed to C:\Rhythmyx, you do not need to modify the servlet configuration.
7. If the path in the new environment is different from the path in the development environment, an additional workaround is required. When CM System is installed, the path to the JRE is hard-coded into the executables and binaries. To start the server (and other binaries/executables), you must override this path.

   - In a Solaris or Linux environment, create a symbolic link that points the development installation location to the new deployed location.

- In a Windows environment, if you start the CM System server manually, create a new shortcut. Open the shortcut, and on the Shortcut tab, in the **Target** field, add the parameter `-is:javahome ./JRE.` NOTE: You must add a space between the `RhythmyxServer.exe` portion of the target path and the `-is:javahome` parameter.
- In a Windows environment, if you run the CM System server as a service, modify the registry entry for the CM System server service. (WARNING: Use extraordinary caution when modifying registry entries. Only make the specific modifIcations described and do not modify any other registry entries. Modifying other registry entries may cause other programs and applications to fail, or even cause the Windows operating system as a whole to fail.)

i) Start the Registry Editor. (Click the [**Start**] button and choose **Run**. Enter **regedit** and click the [**OK**] button.)

ii) Find the key for the CM System server service. (My Computer\ HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ {Rhythmyx server service name})

iii) Expand the Folder, and click on the Parameters Subfolder.

iv) Right-click on the Start Cmd, and from the popup menu, choose **Modify**. Modify the Start Cmd Data column value by adding `-is:javahome ./JRE` to the command. For example, if the command is `"C:\Rhythmyx\RhythmyxServer.exe"`, change it to `"C:\Rhythmyx\RhythmyxServer.exe" -is:javahome ./JRE`. Note that you must include a space between `"RhythmyxServer.exe"` and `-is:javahome ./JRE`.

Updating the Datasource

To update the datasource:

1. In a simple text editor, open `<RhythmyxRoot>\AppServer\server\rx\deploy\rx-ds.xml`.
2. Find the `<connection-url>` node and replace the SOURCE credentials with the TARGET credentials. See the example code below for illustrations of updates to commonly used RDBMSs.
3. Replace the `<security-domain>`…`</security-domain>` node with `<user-name>` and `<password>` nodes containing the appropriate data. See the example code below for illustrations of updates to commonly used RDBMSs.

Once you have completed all updates and have started the CM System server, log in to the Server Administrator and go to the Datasources tab. Click the [**Apply**] button. CM System will re-encrypt the security credentials.

In the following example code, the bolded code illustrates the update to the `<connection-url>` and the underlined code illustrates the replacement of the `<security-domain>` node with the `<user-name>` and `<password>` nodes.

Example Code: MS SQL Server

Source

```
<datasources>
   <local-tx-datasource>
      <jndi-name>jdbc/RhythmyxData</jndi-name>
      <connection-url>jdbc:jtds:sqlserver://source</connection-url>
      <driver-class>net.sourceforge.jtds.jdbc.Driver</driver-class>
      <security-domain>rx.datasource.jdbc_RhythmyxData</security-domain>
      <min-pool-size>5</min-pool-size>
      <max-pool-size>100</max-pool-size>
      <idle-timeout-minutes>15</idle-timeout-minutes>
      <metadata>
         <type-mapping>MS SQLSERVER2000</type-mapping>
      </metadata>
   </local-tx-datasource>
</datasources>
```

Target

```
<datasources>
   <local-tx-datasource>
      <jndi-name>jdbc/RhythmyxData</jndi-name>
      <connection-url>jdbc:jtds:sqlserver://target</connection-url>
      <driver-class>net.sourceforge.jtds.jdbc.Driver</driver-class>
      <user-name>sa</user-name>
          <password>sample</password>
      <min-pool-size>5</min-pool-size>
      <max-pool-size>100</max-pool-size>
      <idle-timeout-minutes>15</idle-timeout-minutes>
      <metadata>
         <type-mapping>MS SQLSERVER2000</type-mapping>
      </metadata>
   </local-tx-datasource>
</datasources>
```

Example Code: Oracle

Source

```
<datasources>
   <local-tx-datasource>
      <jndi-name>jdbc/RhythmyxData</jndi-name>
      <connection-url>jdbc:oracle:thin:@source:1521:source</connection-url>
      <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
      <security-domain>rx.datasource.jdbc_RhythmyxData</security-domain>
      <min-pool-size>5</min-pool-size>
      <max-pool-size>100</max-pool-size>
      <idle-timeout-minutes>15</idle-timeout-minutes>
      <valid-connection-checker-class-
name>org.jboss.resource.adapter.jdbc.vendor.OracleValidConnectionChecker
</valid-connection-checker-class-name>
      <exception-sorter-class-
name>org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter</excep
tion-sorter-class-name>
      <metadata>
         <type-mapping>Oracle8</type-mapping>
      </metadata>
   </local-tx-datasource>
</datasources>
```

Target

```
<datasources>
   <local-tx-datasource>
      <jndi-name>jdbc/RhythmyxData</jndi-name>
      <connection-url>jdbc:oracle:thin:@target:1521:target</connection-url>
      <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
      <user-name>rxuser</user-name>
          <password>sample</password>
      <min-pool-size>5</min-pool-size>
      <max-pool-size>100</max-pool-size>
      <idle-timeout-minutes>15</idle-timeout-minutes>
      <valid-connection-checker-class-
name>org.jboss.resource.adapter.jdbc.vendor.OracleValidConnectionChecker
</valid-connection-checker-class-name>
      <exception-sorter-class-
name>org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter</excep
tion-sorter-class-name>
      <metadata>
         <type-mapping>Oracle8</type-mapping>
      </metadata>
   </local-tx-datasource>
</datasources>
```

Updating the Spring Configuration

To update the Spring configuration:

1. In a simple text editor, open `<RhythmyxRoot>\AppServer\server\rx\deploy\rxapp.ear\rxapp.war\WEB-INF\config\spring\server-beans.xml`
2. Find the node `<bean class="com.percussion.services.datasource.PSDatasourceConfig" id="RhythmyxData">`.
3. Change either the `database` property or the `origin` property appropriately

Example Code: MS SQL Server

Source

```
<bean class="com.percussion.services.datasource.PSDatasourceConfig"
id="RhythmyxData">
```

```
      <property name="name" value="RhythmyxData"/>
      <property name="dataSource" value="jdbc/RhythmyxData"/>
      <property name="database" value="rxmaster"/>
      <property name="origin" value="dbo"/>
</bean>
```

Target

```
<bean class="com.percussion.services.datasource.PSDatasourceConfig"
id="RhythmyxData">
      <property name="name" value="RhythmyxData"/>
      <property name="dataSource" value="jdbc/RhythmyxData"/>
      <property name="database" value="rxproduction"/>
      <property name="origin" value="dbo"/>
</bean>
```

Example Code: Oracle

Source

```
<bean class="com.percussion.services.datasource.PSDatasourceConfig"
id="RhythmyxData">
      <property name="name" value="RhythmyxData"/>
      <property name="dataSource" value="jdbc/RhythmyxData"/>
      <property name="database" value=""/>
      <property name="origin" value="development"/>
</bean>
```

Target

```
<bean class="com.percussion.services.datasource.PSDatasourceConfig"
id="RhythmyxData">
      <property name="name" value="RhythmyxData"/>
      <property name="dataSource" value="jdbc/RhythmyxData"/>
      <property name="database" value=""/>
      <property name="origin" value="production"/>
</bean>
```

Updating the CM System Servlet Configuration

To update the CM System servlet configuration:

1. In a simple text editor, open `<RhythmyxRoot>\AppServer\server\rx\deploy\rxapp.ear\rxapp.war\WEB-INF\ web.xml`
2. Find the `<init-param>` where the value of the `<param-name>` child is *rxDir*.
3. Change the value of the `<param-value>` child node to match the directory path of the CM System root location.

Example Code: Windows

Source

```
<init-param>
      <param-name>rxDir</param-name>
      <param-value>C:\Rhythmyx</param-value>
</init-param>
```

Target

```
<init-param>
      <param-name>rxDir</param-name>
      <param-value>D:\CMS\Rhythmyx</param-value>
</init-param>
```

Example Code: Solaris/Linux

Source

```
<init-param>
      <param-name>rxDir</param-name>
      <param-value>/usr/local/rx/Rhythmyx</param-value>
</init-param>
```

Target

```
<init-param>
      <param-name>rxDir</param-name>
      <param-value>/home/Production/CMS/Rhythmyx</param-value>
</init-param>
```

Special Deployment Case

One specific deployment case requires special attention: deploying a development environment to production, then deploying that production environment back over the development environment.

Development > > Production >> Development

If any Multi Server Manager archives have been deployed to the production environment before it is redeployed over the original development environment, subsequent deployment of Multi Server Manager archives from the development environment to the production environment will not behave correctly. Instead of updating the existing design elements on the production server with changes from the development server, the design elements will be added to the production server as new design objects. This behavior occurs because the original ID mappings between the development environment and the production environment have been invalidated by the redeployment of the production server.

For example, assume that you created a new Template in development with the ID 301. If you use Multi Server Manager to deploy the Template to the production environment, it is assigned a new ID (for example, 1021). When you redeploy the production environment by copying it over the development environment, the original Template ID (301) is overwritten with the ID from production (1021). If you subsequently attempt to deploy the Template from the updated development environment to production, no mapping exists for the development Template with the ID 1021, so Multi Server Manager adds the Template to the production environment as if it were a new Template.

To prevent this problem, if you overlay a source environment (such as a development environment) with a target environment (such as a production environment), you need to clear the ID mappings in the target environment, and remap them when you deploy your Multi Server Manager packages.

To clear the ID mappings, in the Repository database, delete all rows from the table DPL_ID_MAPPINGS.

Updating the Deployed Environment

After you deploy your production environment you will probably continue to develop and refine your implementation. All such modifications should be implemented in your development environment. Once they are working as intended. Use Rhythmyx Multi-Server Manager to deploy updates from the development environment to the production environment.

CHAPTER 3

Implementing Security in the Production Environment

To implement security for CM System, you must define a security provider, which supplies a list of users for CM System and authenticates them when they log in. You can then add your users to their Roles.

Finally, CM System provides some default security settings which you should modify to prevent unintended access to your system.

Security providers link to external resources that list users that can use CM System. These external resources provide authentication support when a user attempts to log in to CM System, and in some cases also define the CM System Roles the user belongs to when logged in.

Four types of security providers are available for CM System:

- Directory Connection

  A directory connection security provider uses a directory server, such as LDAP or Microsoft Active Directory, to list and authenticate users. This security provider can also define the CM System Roles for the users. The directory connection security provider is the recommended security provider.

- Backend Table

  The backend table security provider uses a database table to list and authenticate users. No graphic front end is provided for this table, however, and both the user name and the password are stored as clear text. While this security provider is useful for development environments (the default CM System users shipped by Percussion Software are stored in the USERLOGIN table in the Repository database), it is not recommended for production environments.

- Web Server

  The web server security provider derives users from a web server's security provider, and uses that security provider to authenticate the users. The exact security mechanism is controlled by the web server itself. This security provider is recommended for use only on portals or when CM System runs as a servlet on another servlet container that provides security.

Defining Security Providers

Before you can add users to your system, you must define a security provider.

Security providers link to external resources that list users that can use CM System. These external resources provide authentication support when a user attempts to log in to CM System, and in some cases also define the CM System Roles the user belongs to when logged in.

Four types of security providers are available for CM System:

- Directory Connection

  A directory connection security provider uses a directory server, such as LDAP or Microsoft Active Directory, to list and authenticate users. This security provider can also define the CM System Roles for the users. The directory connection security provider is the recommended security provider.

- Backend Table

  The backend table security provider uses a database table to list and authenticate users. No graphic front end is provided for this table, however, and both the user name and the password are stored as clear text. While this security provider is useful for development environments (the default CM System users shipped by Percussion Software are stored in the USERLOGIN table in the Repository database), it is not recommended for production environments.

- Web Server

  The web server security provider derives users from a web server's security provider, and uses that security provider to authenticate the users. The exact security mechanism is controlled by the web server itself. This security provider is recommended for use only on portals or when CM System runs as a servlet on another servlet container that provides security.

Implementing a Directory Connection Security Provider

A directory connection security provider uses a directory server, such as LDAP or Microsoft Active Directory, to store the list of users and to provide authentication when they attempt to log in to CM System. LDAP servers also provide the option of defining customer attribute identifiers, which provides additional flexibility for defining attributes you can use in CM System. This feature is particularly useful for associating Roles with your users in your LDAP server. You can maintain your Roles as part of the user attributes in LDAP rather than in CM System. Using this approach simplifies user maintenance in CM System.

Note that Active Directory does not provide this ability. If you use Active Directory as your directory server, you will have to assign users to Roles in CM System rather than using the directory server to maintain their Roles.

Implementing a directory connection security provider involves two phases:

- Defining a directory services configuration

  A directory services configuration defines the data used to connect to the directory server and to authenticate the user. A directory services configuration consists of the following data:

  - Authentication

Authentication data defines the data used to log in to the directory server.

- Directory Configuration

  A Directory Configuration defines the data required to connect to a specific directory.

- Directory Sets

  A Directory Set defines a group of Directory Configurations that can be accessed together, and the connection information required to connect to them. You must define a Directory Set before you can define a Directory Connection Security Provider. A Directory Set may consist of a single Directory Configuration, of multiple Directory Configurations for directories on the same directory server, or of multiple Directory Configurations for directories on different directory servers.

- Role Providers (optional)

  A Role Provider defines the data that determines how CM System will use directory server information to determine the user's Roles once they have been authenticated.

- Defining the directory connection security provider

This section will describe an example set up of a directory connection security provider. You may find it useful to download and install an LDAP browser to facilitate your directory services configuration. The browser allows you to look up and confirm attribute, connection, and directory information. An LDAP browser makes it easier to complete the directory services configuration, but the browser is not required to complete the configuration successfully.

LDAP Configuration

In our example configuration, we will be using a SunONE directory server that resides on a machine named FastForward. The directory server listens on port 389 (the default LDAP port).

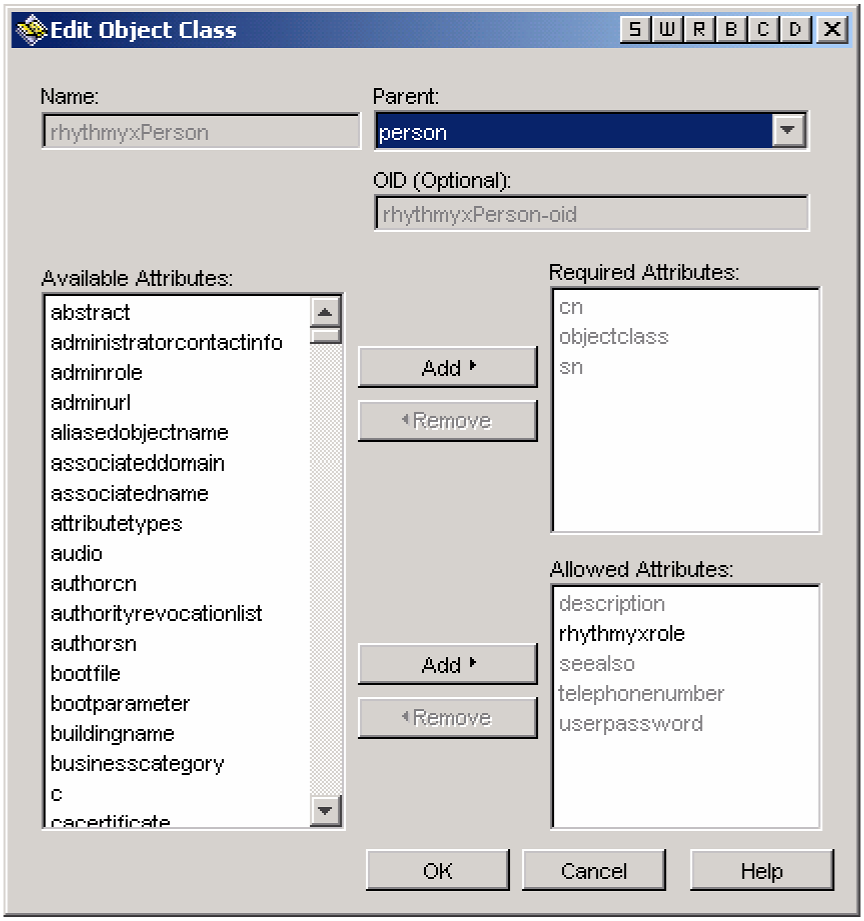
We use LDAP to authenticate three users:

- Lisa Kerr, Ed Wong, and Kent Hoyt are members of the Content Contributors Group. We want to assign them to the Author Role in CM System. These users do not belong to any other groups, and they should not be assigned to any other functional Roles.
- Bernadette Bridge is a member of the Team Captains Group, but she is the only member of that group we want to have access to CM System. She will be a member of the Admin Role.
- All of these users must be Members of the XI_Members Community.

We will also use Bernadette Bridge to bind to the directory server, so she must be able to search for and read attribute values for any user in the directory that CM System will catalog on or authenticate. If we were going to implement a Group Provider, Bernadette would also have to be able to search and read attribute values of any Groups CM System would use to authenticate.

In addition to authentication, we want to derive Role information from the directory server (in other words, we want to use it as a Role Provider). When using a directory server as a Role Provider, the user objects searched during a query must contain attributes that will correlate to an existing CM System Role. In other words, each user must have an object or group of objects, whose value equals the name of a Role in CM System. Thus, the user objects for Lisa, Ed, and Kent must include an object whose values include "Author" and "XI_Members".

Best practice when using LDAP is not to add a custom attribute to an existing object class, such as person. Instead, create a new object class and add the custom attribute to that object's class. In our example, we will create the new object class rhythmyxperson, and add the custom attribute, rhythmyxrole, to that object class.



Name: rhythmyxPerson

Parent: person

OID (Optional): rhythmyxPerson-oid

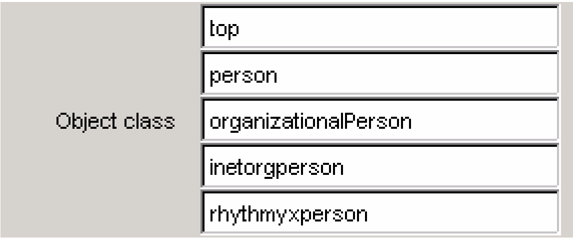
Available Attributes:
abstract
administratorcontactinfo
adminrole
adminurl
aliasedobjectname
associateddomain
associatedname
attributetypes
audio
authorcn
authorityrevocationlist
authorsn
bootfile
bootparameter
buildingname
businesscategory
c
cacertificate

Required Attributes:
cn
objectclass
sn

Allowed Attributes:
description
rhythmyxrole
seealso
telephonenumber
userpassword

Figure 1: rhythmyxPerson Object Class with rhythmyxrole Attribute

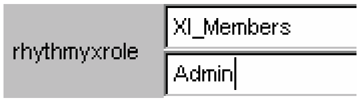
We can then add this class to the list of object classes for the user object:



Object class	top person organizationalPerson inetorgperson rhythmyxperson

Figure 2: rhythmyxperson Object Class Added to the User Object

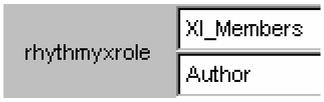
Once we have added the rhythmyxperson object class, we can add the rhythmyxrole attribute to the user object and assign it values. For example, Bernadette Bridge's Role definition would be:



rhythmyxrole	XI_Members Admin

Figure 3: LDAP Role Definition for Bobby Bluefin

While the Role definition for Lisa, Ed, and Kent would be:



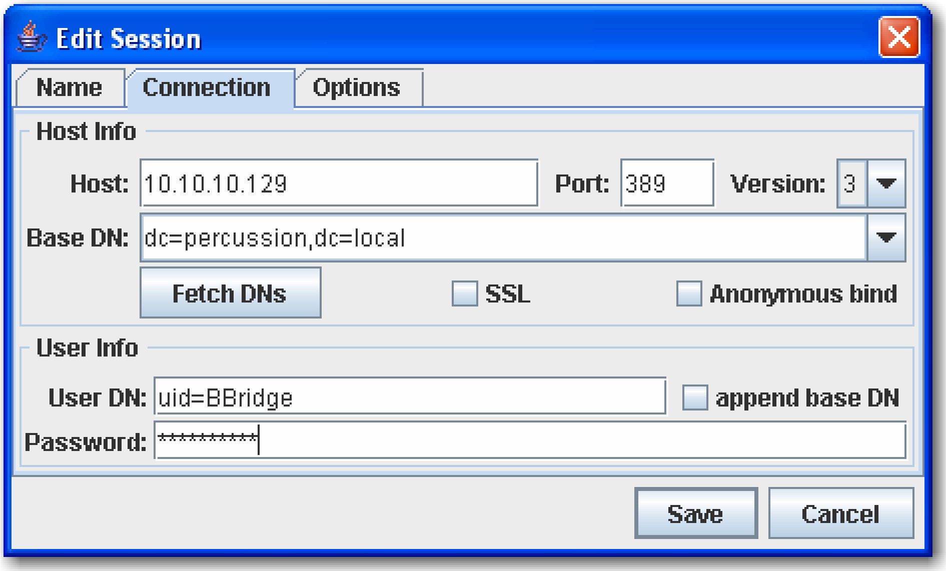
rhythmyxrole	XI_Members Author

Figure 4: LDAP Role Definition for Tiara Tuna and Nancy Needlenose

Defining an Authentication for a Directory Services Configuration

The first step in defining the directory server configuration is to define an Authentication.

Before defining the Authentication, you might want to use an LDAP browser to test that the authentication data will work. Start the browser and create a connection to your directory server:



Host Info

Host: 10.10.10.129 Port: 389 Version: 3

Base DN: dc=percussion,dc=local

□ SSL □ Anonymous bind

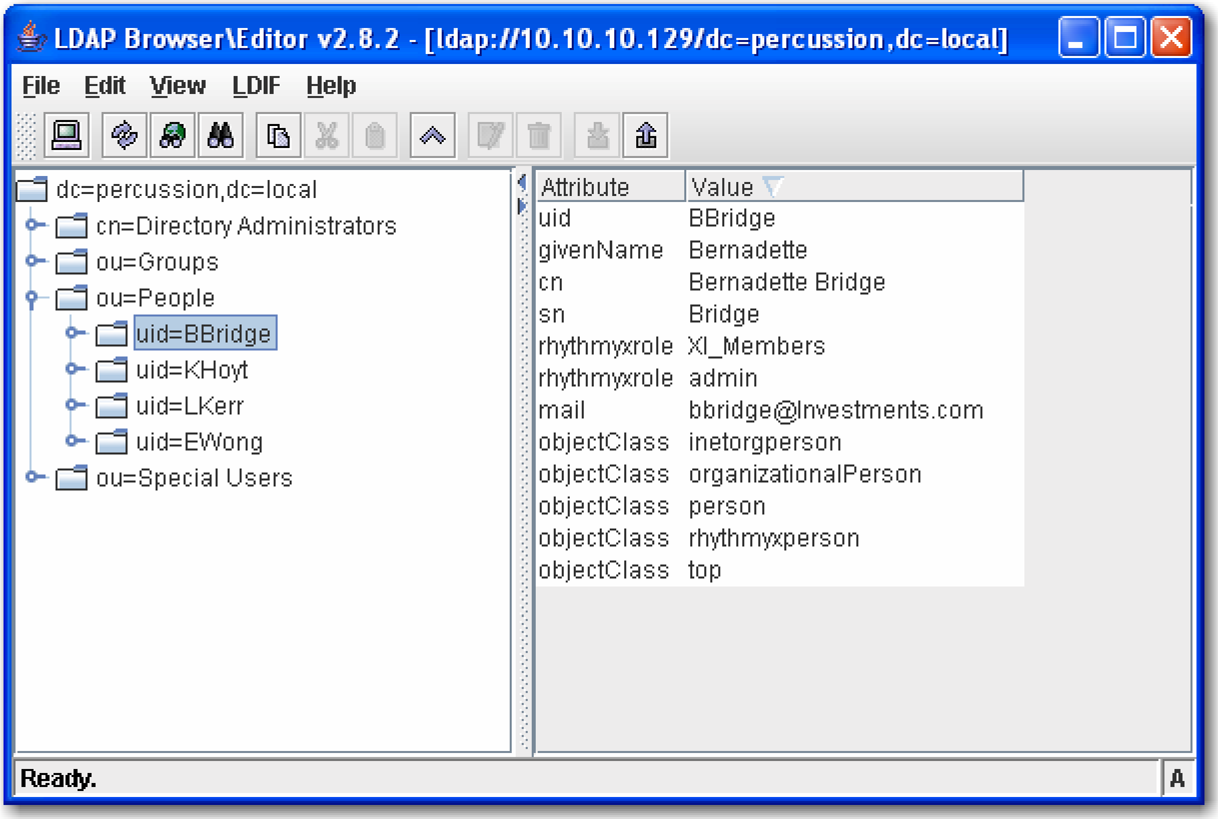
User Info

User DN: uid=BBridge □ append base DN

Password: **********

Figure 5: Connection for Bernadette Bridge

If the credentials are correct, you will be able to bind and catalog the directory:


LDAP Browser\Editor v2.8.2 - [ldap://10.10.10.129/dc=percussion,dc=local]
File
Edit
View
LDIF
Help
Ready.
A

dc=percussion,dc=local
- cn=Directory Administrators
- ou=Groups
- ou=People
  - uid=BBridge
  - uid=KHoyt
  - uid=LKerr
  - uid=EWong
- ou=Special Users

Attribute	Value
uid	BBridge
givenName	Bernadette
cn	Bernadette Bridge
sn	Bridge
rhythmyxrole	XI_Members
rhythmyxrole	admin
mail	bbridge@Investments.com
objectClass	inetorgperson
objectClass	organizationalPerson
objectClass	person
objectClass	rhythmyxperson
objectClass	top

Figure 6: LDAP Browser connected using Bernadette Bridge's Connection

To define an Authentication:

1 Start the Rhythmyx Server Administrator.

2 Click the Directory Services tab. The default subtab is the Authentications tab.

3 Click the [**Add**] button.

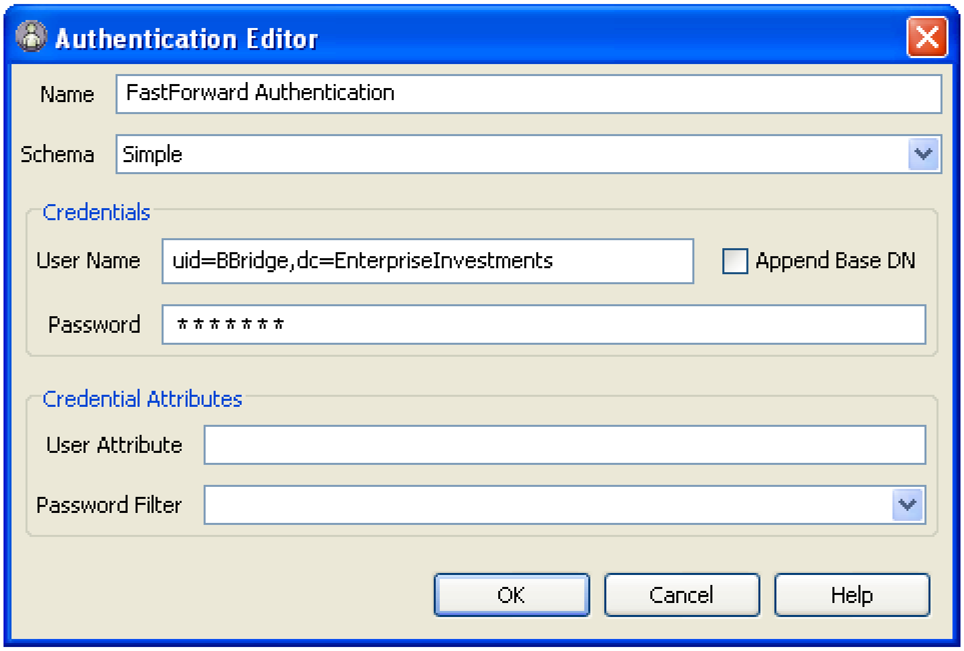
CM System displays the Authentication Editor.

4 In the **Name** field, enter *FastForward Authentication*.

5 In the **Schema** drop list, choose *Simple*.

6 In the **User Name** field, enter the fully-qualified distinguished name for Bernadette Bridge: *uid=BBridge,dc=EnterpriseInvestments*. Note that since we are using a SunONE server, the user attribute is *uid*. If we were defining an Authentication for Active Directory, the Naming Attribute would be CN (note UPPER CASE).

7 In the **Password** field, enter *DeepSea* (which is Bernadette Bridge's password).



Authentication Editor

Name: FastForward Authentication

Schema: Simple

Credentials

User Name: uid=BBridge,dc=EnterpriseInvestments

□ Append Base DN

Password: *******

Credential Attributes

User Attribute:

Password Filter:

Figure 7: Authentication for FastForward Directory Server

NOTE: Do not modify the **Append Base DN** or **User Attribute** fields.

8 Click the [**OK**] button to save the Authentication.

9 On the Rhythmyx Server Administrator, click the [**Apply**] button.

Defining a Directory for a Directory Services Configuration

The second step in defining a directory services configuration is to define a Directory. We will use the FastForward Authentication to connect to this directory.

To define a Directory:

1 In the Rhythmyx Server Administrator, click on the Directory Services tab, then on the Directories subtab.

2 Click the [**Add**] button.

CM System displays the Directory Editor.

3 In the **Name** field, enter *FastForward Directory*.

4 In the **Catalog** drop list, leave the default option, *Shallow*, selected.

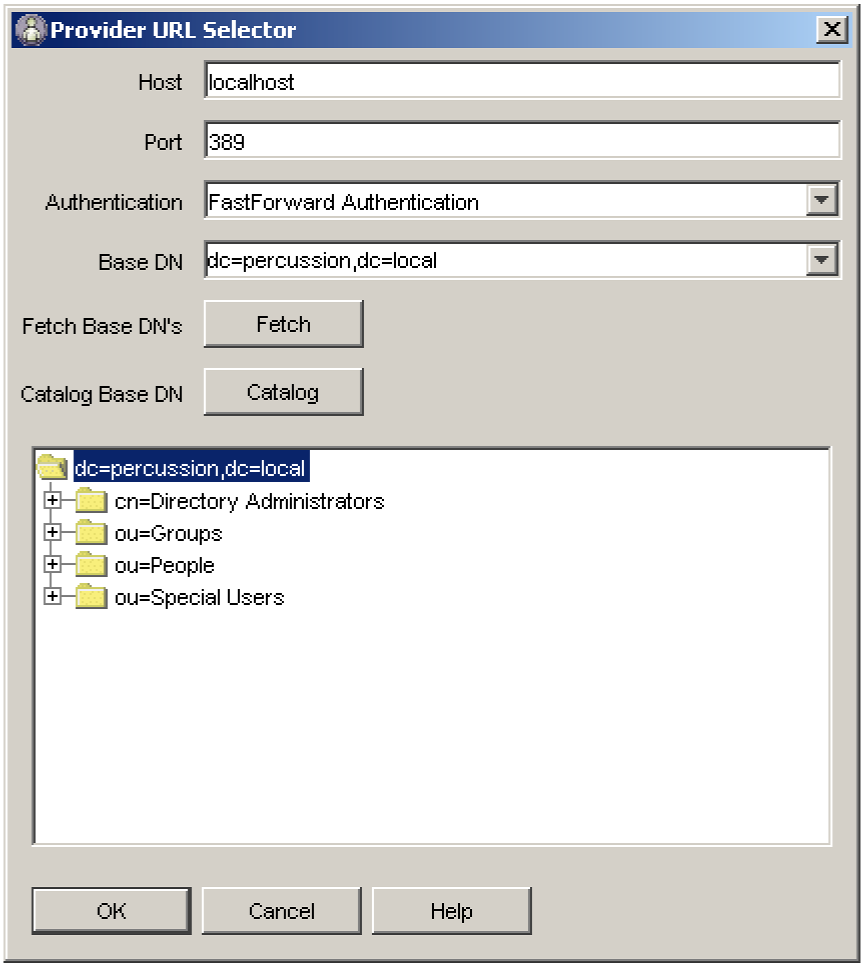
5 In the **Factory** drop list, leave the default option, *com.sun.jndi.ldap.LdapCtxFactory*, selected.

6 In the **Authentication** drop list, choose the *FastForward Authentication*.

7 Complete the provider URL:

d) Click the browse button next to the **Provider URL** field.

e) CM System displays the Provider URL dialog. If your Authentication works correctly, it will already have cataloged and fetched the Base DN.



Provider URL Selector

Host: localhost

Port: 389

Authentication: FastForward Authentication

Base DN: dc=percussion,dc=local

dc=percussion,dc=local
- cn=Directory Administrators
- ou=Groups
- ou=People
- ou=Special Users

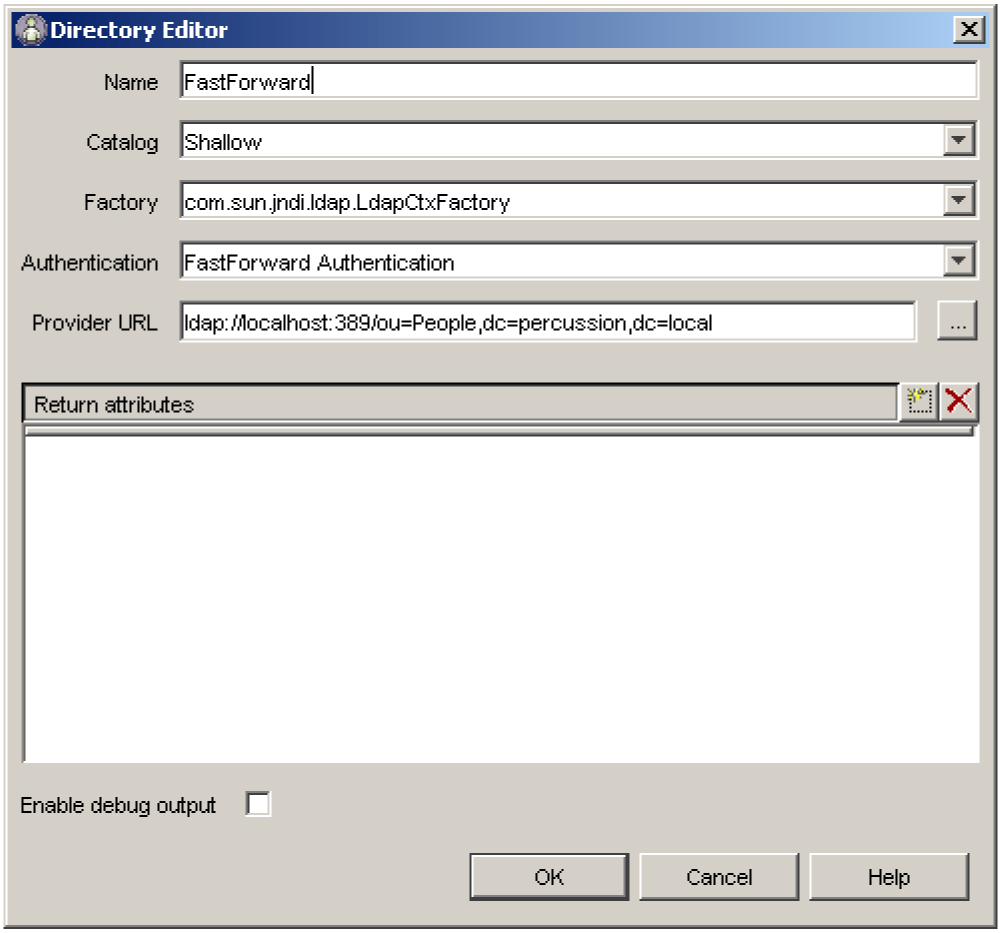
Figure 8: Provider URL Select Dialog showing DNs Cataloged and Fetched

f) If the **Base DN** field and the catalog field (the large unlabelled field at the bottom of the dialog) are blank, click the [**Catalog**] button to catalog DNs, then click the [**Fetch**] button to fetch the base DN.

g) In the directory tree in the large unlabelled field, select the ou=People directory.

h) Click the [**OK**] button.

i) CM System returns to the Directory Editor dialog.



Directory Editor

Name: FastForward

Catalog: Shallow

Factory: com.sun.jndi.ldap.LdapCtxFactory

Authentication: FastForward Authentication

Provider URL: ldap://localhost:389/ou=People,dc=percussion,dc=local

Figure 9: FastForward Directory Definition

8 Click the [**OK**] button to save the Directory definition.

9 On the Rhythmyx Server Administrator, click the [**Apply**] button.

Defining a Directory Set for a Directory Services Configuration

The third step in defining a directory services configuration is to define the Directory Set. You must define a Directory Set before you can create a Directory Connection Security Provider. Our Directory Set will consist of the FastForward Directory.

To define a Directory Set:

1 On the Rhythmyx Server Administrator, click the Directory Services tab and the Directory Sets sub-tab.

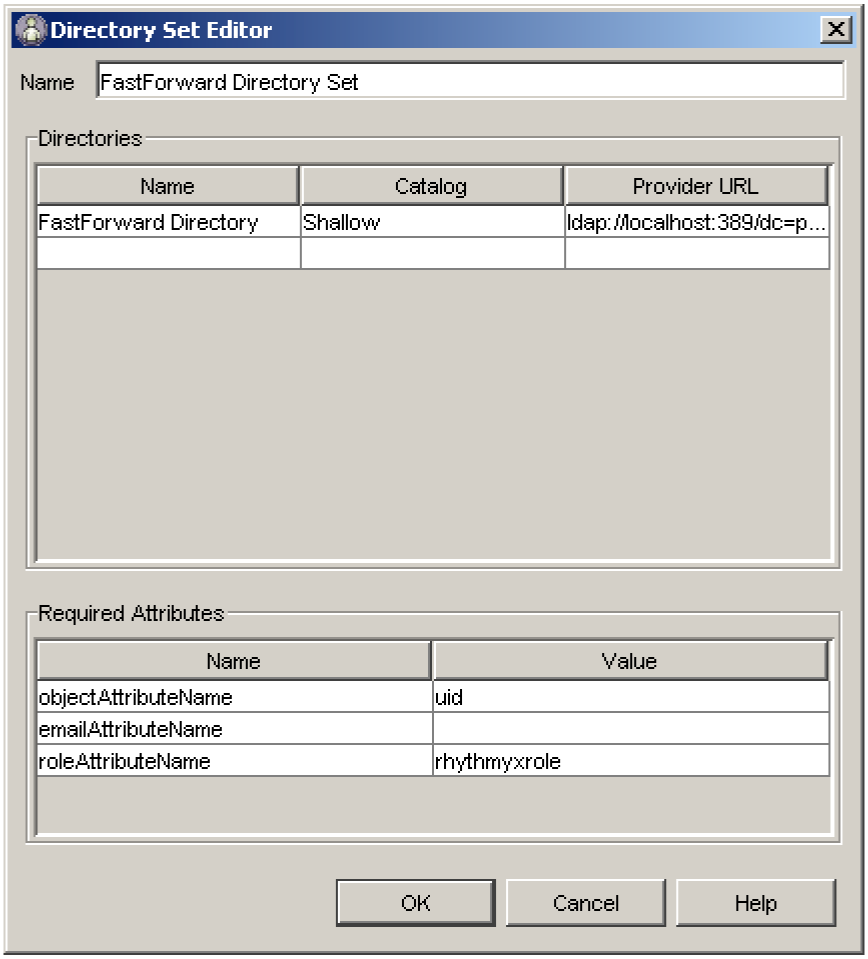
2 On the Directory Set sub-tab, click the [**Add**] button.

CM System displays the Directory Set Editor.

3 In the **Name** field, enter *FastForward Directory Set*.

4 In the Directories box, double-click in the **Name** column and from the drop list, choose *FastForward Directory*. When you choose this option, values for Catalog (*Shallow*) and Provider URL (*ldap://localhost:389/dc=percussion,dc=local*) will be added automatically.

5 In the Required Attributes box, click on the **Values** column of the roleAttributeName row, and enter *rhythmyxrole*.



Directory Set Editor

Name: FastForward Directory Set

Directories

Name	Catalog	Provider URL
FastForward Directory	Shallow	ldap://localhost:389/dc=p...

Required Attributes

Name	Value
objectAttributeName	uid
emailAttributeName	
roleAttributeName	rhythmyxrole

Figure 10: FastForward Directory Set

6 Click the [**OK**] button to save the Directory Set.

7 On the Rhythmyx Server Administrator, click the [**Apply**] button.

Defining a Role Provider for a Directory Services Configuration

Since we want to use our directory server to maintain CM System Roles, we must define a Role Provider.

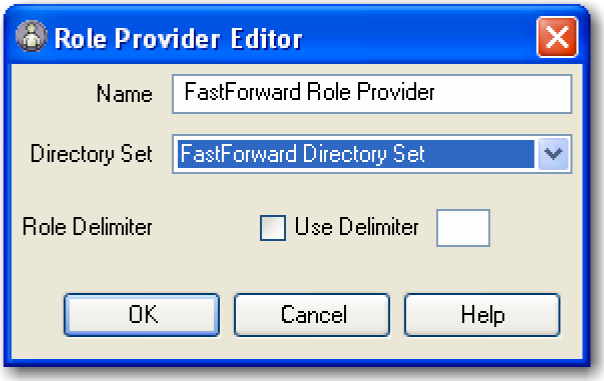
NOTE: You cannot use Microsoft ActiveDirectory as a Role Provider. If you use Microsoft ActiveDirectory, you must maintain Roles manually.

To define a Role Provider:

1 On the Rhythmyx Server Administrator, click the Directory Services tab, then the Role Providers subtab.

2 On the Role Providers tab, click the [**Add**] button.

3. In the **Name** field, enter *FastForward Role Provider*.
4. In the **Directory Set** drop list, choose *FastForward Directory Set*.



Name: FastForward Role Provider

Directory Set: FastForward Directory Set

Role Delimiter: □ Use Delimiter

Figure 11: FastForward Role Provider

5. Click the [**OK**] button to save the Role Provider.
6. On the Rhythmyx Server Administrator, click the [**Apply**] button.

Creating a Directory Connection Security Provider

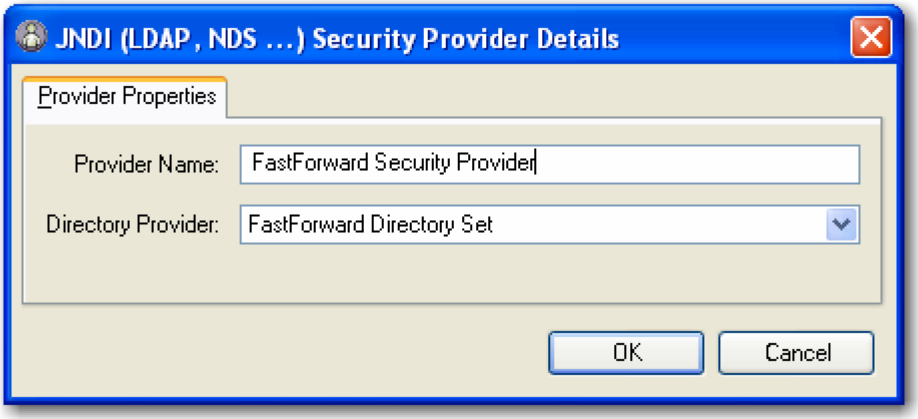
Once you have completed the directory services definition, you can define a Directory Connection Security Provider that allows you to query the directory server to authenticate your users, and in our case to retrieve Role information for our users as well.

To create a Directory Connection Security Provider:

1. On the Rhythmyx Server Administrator, click the Security tab. The default subtab is the Security Providers tab. Leave this tab selected.
2. On the Security Providers tab, click the [**New**] button.
3. CM System displays the Select new security provider type dialog. From the drop list, choose *Directory Connection Security Provider* and click the [**OK**] button.

   CM System displays the JNDI Security Provider Details dialog.
4. In the Provider Name field, enter *FastForward Security Provider*.

5 In the Directory Provider drop list, choose *FastForward Directory Set*.



Provider Name: FastForward Security Provider

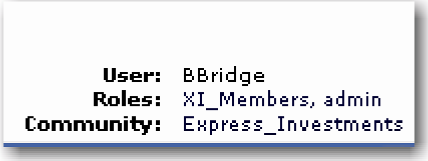
Directory Provider: FastForward Directory Set

Figure 12: FastForward Directory Connection Security Provider

6 Click the [**OK**] button to save the FastForward Security Provider.

Testing the Directory Connection Security Provider

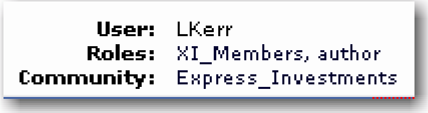
To test the directory services configuration and Directory Connection Security Provider, we start a browser and attempt to log in as Bernadette Bridge. When the browser displays the login dialog, we enter the username BBridge and enter Bernadette's password (DeepSea). If we have configured everything correctly, Bernadette will be authenticated and logged in to the XI_Members Community and the Admin Role:



User: BBridge
Roles: XI_Members, admin
Community: Express_Investments

Figure 13: Bernadette Bridge Logged in to Content Explorer

Lisa Kerr will be logged in to XI_Members Community and the Author Role:



User: LKerr
Roles: XI_Members, author
Community: Express_Investments

Figure 14: Lisa Kerr Logged in to Content Explorer

Implementing a Backend Table Security Provider

A Backend Table Security Provider uses a backend database table that stores usernames and passwords. (CM System uses the USERLOGIN table for this purpose). When authenticating a user, CM System queries the table to find matches for the username and password entered when logging in to CM System. If it finds matching values, the user has been authenticated and is logged in to CM System.

CM System does not provide an interface to the backend table. Use standard RDBMS mechanisms to maintain the values in the tables.

All data in the backend table, including the password, is stored unencrypted. Therefore Percussion Software recommends against using this Security Provider in the production environment.

For the purposes of this exercise, we will assume that we have created a new table, FASTFORWARDUSERS in the rxmaster database. This table consists of two columns:

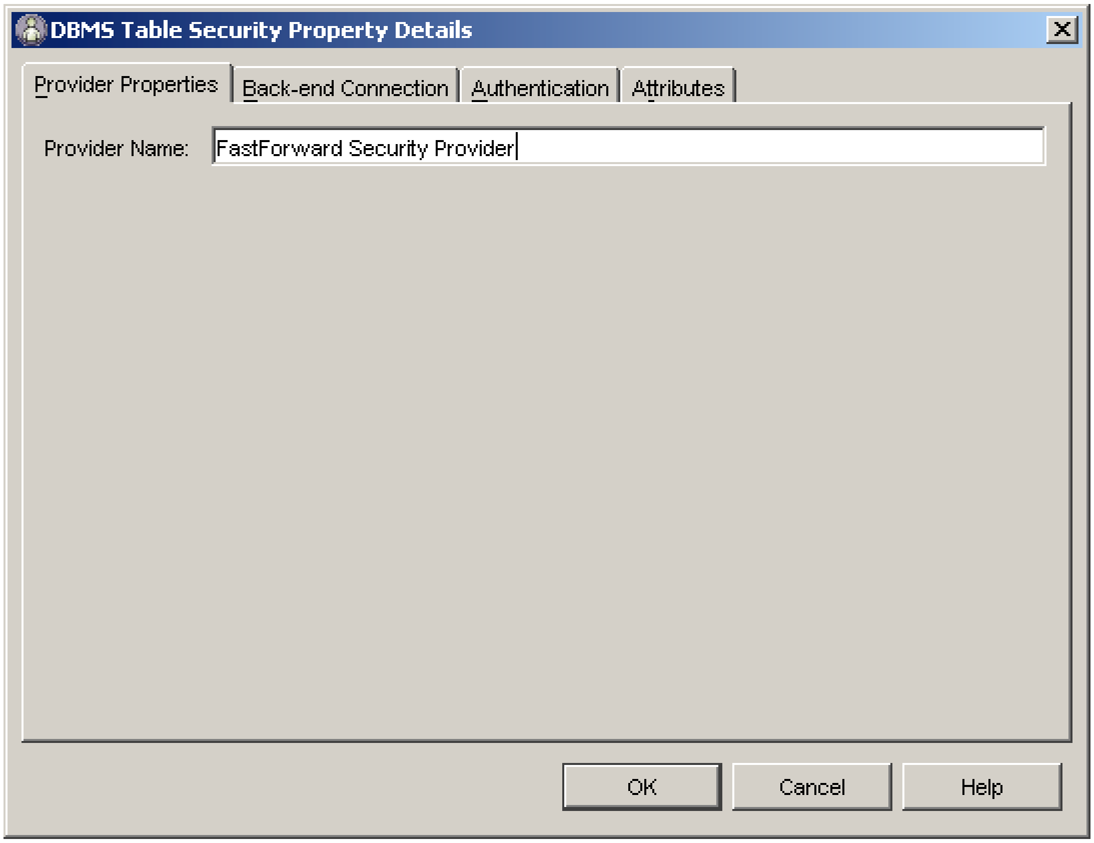
- USERNAME
- LOGIN

To create a backend table security provider:

1. On the Rhythmyx Sever Administrator, select the Security tab. The default subtab is Security Providers. Select the Security Providers tab if it is not already selected.
2. On the Security Providers tab, click the [**New**] button.
3. On the Select New Security Provider Type dialog, select *Backend Table Security Provider* from the drop list. (Note: Backend Table Security Provider is the default option.) Click the [**OK**] button.

   CM System displays the DBMS Table Security Property Details dialog. The default tab for this dialog is the Provider Properties tab.

4 In the **Provider Name** field, enter *FastForward Security Provider*.



Provider Name: FastForward Security Provider

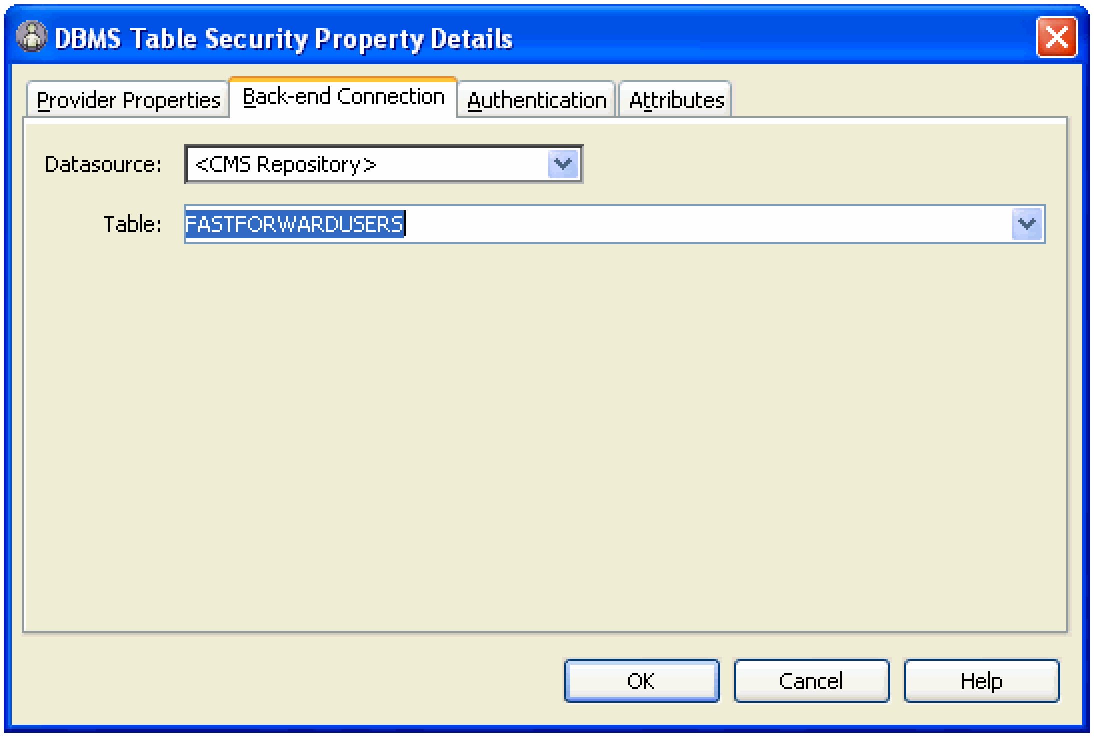
Figure 15: Backend Security Provider Properties Tab

5 Define the backend connection:

a) Click the Backend Connection tab.

b) Choose the **Datasource** you want to use to connect to the database where the authentication data is stored. Options include all Datasources defined in the system. *Note: For information about adding datasources, see the Server Administrator Help*.

c) Choose the **Table** FASTFORWARDUSERS.



Datasource: <CMS Repository>

Table: FASTFORWARDUSERS

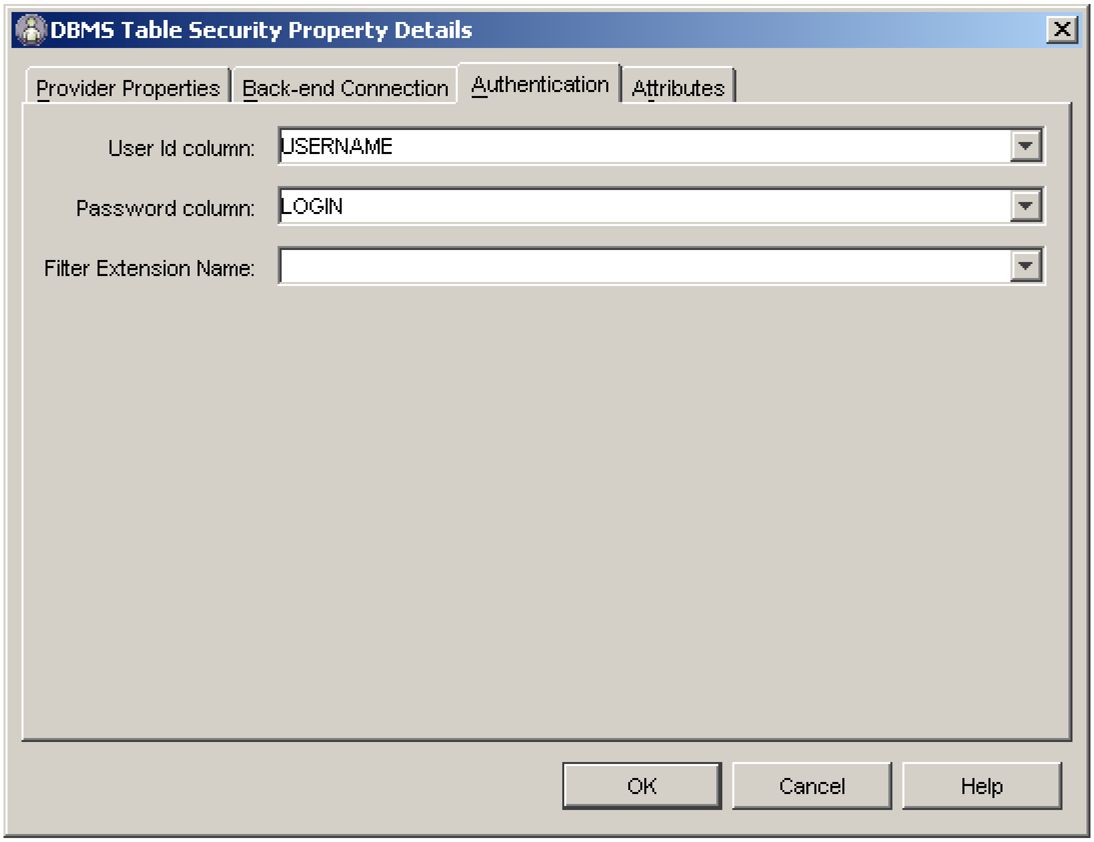
Figure 16: Backend Security Back-end Connection Tab

6 Define the Authentication:

a) Click the Authentication tab.

b) In the **User Id Column** drop list, choose *USERNAME*.

c) In the Password drop list, choose *LOGIN*.



User Id column: USERNAME

Password column: LOGIN

Filter Extension Name:

Figure 17: Backend Security Provider Authentication Tab

7 Click the [**OK**] button to save the new Security Provider.

The new security provider is added to the list of security providers on the Security Provider tab in the Rhythmyx Server Administrator. Users will not be able to log in to Content Explorer successfully, however, until you have ***added them to Roles*** (see "Implementing a Web Server Security Provider" on page 36) as well.

Implementing a Web Server Security Provider

A Web Server Security Provider relies on an existing Web server or Web application server to provide authentication. To use the Web Server Security Provider, you must interface to CM System as a servlet under the Web server or Web application server. This type of authentication is useful when implementing a single sign-on model using an identity and access management system such as Netegrity's SiteMinder, or when accessing CM System through a portal such as BEA WebLogic.

When using a Web Server Security Provider, if the associated Web server can authenticate the user, CM System grants access.

CM System ships with a default Web Server Security Provider that is configured to interface with the CM System servlet. Once you deploy the Rhythmyx.war file to your Web server or Web application server, users should be able to authenticate with no further intervention from you.

Adding Users to CM System Roles

While some security providers, such as the Directory Connection Security Provider and the Web Server Security Provider, allow you to authenticate the user directly into CM System Roles, others, such as the Windows NT and Backend Table Security Providers, require you to add users to Roles manually. Even if you use a security provider that automatically authenticates the user into their Roles, you may prefer to control Role Membership manually.

In any case, you should only add users to their Roles after you have defined a Security Provider. When adding users, you can catalog a Security Provider to retrieve its list of available users, which simplifies the process of adding users significantly.

For details about adding users to Roles, see Adding Users to a Role.

Updating Default Security Settings

When you first install the CM System server, access is controlled by a default Backend Security Provider, which defines the following users:

- admin1
- admin2
- rxserver (See IMPORTANT NOTE below)
- artist1
- artist2
- author1
- author2
- designer1
- designer2
- editor1
- editor2
- qa1
- qa2

Each of these users is assigned the default password *demo*.

Since these users and the default password are available on every new installation of the CM System server, you will probably want to make one of the following modifications to prevent unintended access to your system:

- Assign new passwords to the default users;
- Remove the default users and replace them with new users.

IMPORTANT NOTE: The user `rxserver` is used internally by the CM System server. DO NOT REMOVE OR MODIFY THIS USER. If you remove or modify this user, your CM System server will not work correctly.

Changing the Default Server Access Control List

The default users and their passwords are stored in a backend database table named USERLOGIN. This table consists of two columns:

- USERID
- PASSWORD

The value in the PASSWORD column for each user is *demo*. One quick way to update the value in this column is to write a SQL script to change the value. You can either update all rows with the same new value or you can assign a different value to each column.

If you use and RDBMS that includes an enterprise manager or similar client, (such as Microsoft SQL Server or Oracle), you can modify the value in each column directly. Use Enterprise Manager to open the USERLOGIN table and manually change the password in each column.

If you use Oracle as your RDBMS, you can use the Oracle SQL*Plus application:

1. At the prompt change the password for the admin1 role by entering the following script:

```
update userlogin set password='somethingnotdemo' where
userid='admin1';
```

2. Repeat for each default user, changing the value of `userid` and `password` accordingly.

The new password is in effect immediately.

Replacing the Default Users with New Users

You can also prevent unintended access to your system by deleting the default users. It is best to implement this option ***after you add your production users*** (see "Implementing Security in the Production Environment" on page 19).

If you choose this option, it is strongly recommended that you do not delete the admin1 and admin2 users.

WARNING: DO NOT REMOVE THE USER `rxserver` UNDER ANY CIRCUMSTANCES! This member is used internally in the CM System server. If you remove this user, CM System will not work properly (Aging Transitions and Relationship processing will fail).

To implement this option, remove the users (other than admin1, admin2, and rxserver) from the USERLOGIN table. You can delete the users quickly by writing a SQL script that deletes the unwanted users.

If you use and RDBMS with an enterprise manager client, you can delete the users from the table manually.

1. In Enterprise Manager, select the rows of the users you want to delete, right click on the table and from the popup menu, choose *Delete*.
2. Enterprise Manager displays a confirmation dialog. Confirm that you have not selected admin1, admin2, or rxserver before clicking the [**Yes**] button.

If you use Oracle as your RDBMS, you can use the Oracle SQL*Plus application:

1. At the prompt change the delete the author1 user by entering the following script:

```
delete from userlogin where userid='author1';
```

2 Repeat for each default user other than admin1, admin2, and rxserver, changing the value of `userid` accordingly.

CHAPTER 4

Setting Up Publishing in the Production Environment

Production Web servers that serve content to the public are typically located on a machine accessible from outside of your corporate firewall, typically in the demilitarized zone of your system infrastructure. When publishing your content, you need to move your published pages to the production Web server through the firewall. Two options are available.

- Use file system publishing and map or mount the Web server as a drive on your CM System server. You can also use a UNC path to access the Web server provided the CM System server has permissions to access the Web server's host machine.

  This option requires you to "own" the "external" environment (the environment outside of the firewall) as well as the "internal" environment (the environment inside the firewall). While this requirement may be feasible for intranets, it is not practical for serving content to the Internet, and in either case you must have strong controls over access to both machines.

- Use FTP (or secure FTP [SFTP]) to publish to your Web server.

  If you choose this option, you may also want to use techniques such as SSH tunneling to ensure secure communications between the publisher and the Web server. This implementation is described in the *CM System Implementation Guide*.

Once you have defined the delivery of your content, set up scheduled publishing to automate publishing of your Editions.

Publishing to a Mapped or Mounted Drive

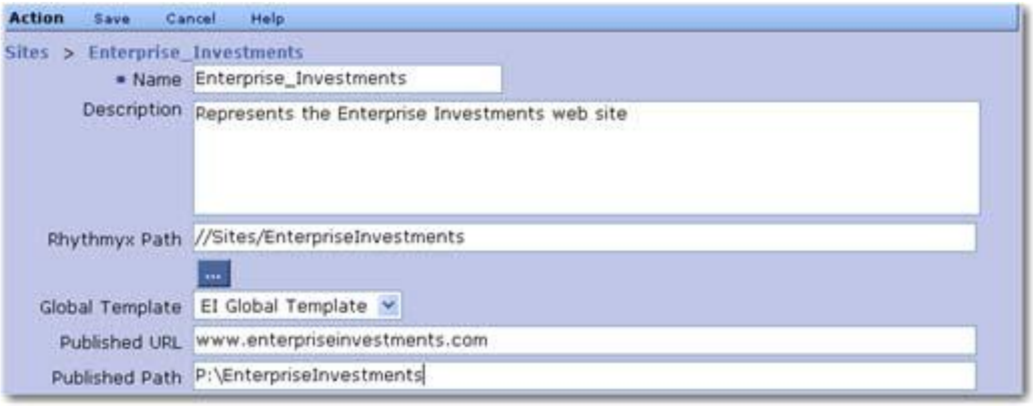
When publishing to a mapped or mounted drive or using a UNC path, you must update the Site configuration for your site to point to the correct location.

For example, assume that:

- we will publish the Enterprise Investments Site to the URL www.enterpriseinvestments.com; and
- the publish location is one of the following:
  - on a Windows machine mapped to P:\EnterpriseInvestments;
  - on a Windows machine at the location \\Quail\htdocs\EnterpriseInvestments (the CM System server has permissions to access this machine);
  - On a Unix machine, mounted to the CM System server as /EIHome.

We will need to update the Site registration for the Express Investments Site as follows:

- Change the value in the **Published URL** field to *www.enterpriseinvestments.com*.
- In the **Published Path**:
  - For a Windows system publishing to a mapped drive, change the value to *P:\EnterpriseInvestments*.



* Name: Enterprise_Investments

Description: Represents the Enterprise Investments web site

Rhythmyx Path: //Sites/EnterpriseInvestments

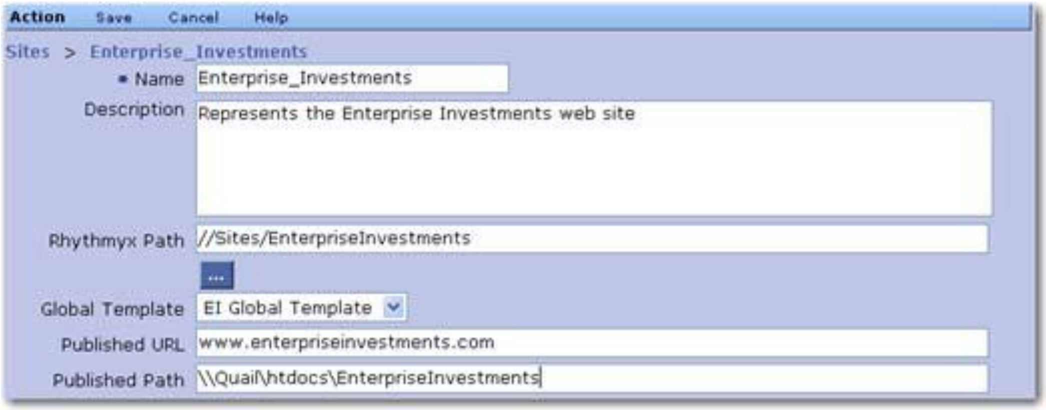
Global Template: EI Global Template

Published URL: www.enterpriseinvestments.com

Published Path: P:\EnterpriseInvestments

Figure 18: Express Investments Site Properties Modified for a Windows Production Server

  - For a Windows system publishing to a UNC path, change the value to *\\Quail\htdocs\EnterpriseInvestments*



* Name: Enterprise_Investments

Description: Represents the Enterprise Investments web site

Rhythmyx Path: //Sites/EnterpriseInvestments

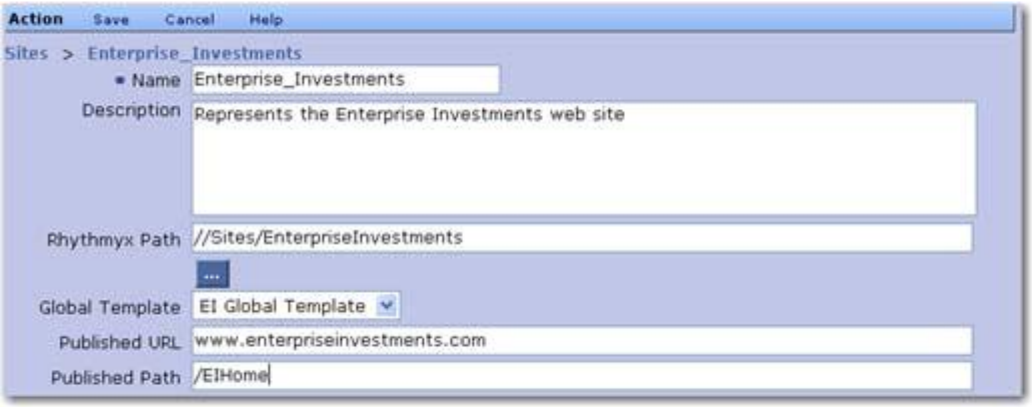
Global Template: EI Global Template

Published URL: www.enterpriseinvestments.com

Published Path: \\Quail\htdocs\EnterpriseInvestments

Figure 19: Express Investments Site Properties Modified for publishing to a UNC path

  - For a Unix system, change the value to */EIHome*



* Name	Enterprise_Investments
Description	Represents the Enterprise Investments web site
Rhythmyx Path	//Sites/EnterpriseInvestments
Global Template	EI Global Template
Published URL	www.enterpriseinvestments.com
Published Path	/EIHome

Figure 20: Express Investments Site Properties for a Unix Production Server

Scheduling Publishing

In a production environment, publishing is usually set up to run automatically on a schedule. You can define scheduled tasks to run Editions (as well as for other purposes) on the Admin tab of Content Explorer. You can also define a task notification template to generate e-mail messages when processing of the Edition (or any other scheduled task) is complete.

In the following exercises, we will set up a schedule to run the EI_Incremental Edition three times a day, as discussed during modeling and design of the Enterprise Investments Site, and to generate a task notification when publishing of the Edition fails.

Defining a Publishing Schedule

During the modeling and design of the Enterprise Investments Site, the decision was made to run the EI_Incremental Edition three times a day. The final decision about the specific times was put off until implementation time. For the purposes of this exercise, we will assume that the decision was to run the Edition at 10 in the morning, 2 in the afternoon, and 6 in the evening each weekday. To implement this schedule, we must define a scheduled task that runs the Edition at the specified times.

The key element in a scheduled task is the **Cron Specification**, which defines the schedule for the task. CM System uses the Quartz Enterprise Job Scheduler (***http://www.opensymphony.com/quartz/*** (http://www.opensymphony.com/quartz/)). For details about writing a cron expression for Quartz, see ***http://www.opensymphony.com/quartz/wikidocs/CronTriggers%20Tutorial.html*** (http://www.opensymphony.com/quartz/wikidocs/CronTriggers%20Tutorial.html).

The specification for the schedule we want to implement is:

```
   0 0 10,14,18 ? * 2-6
```

Note that we could also use day abbreviations in the day of the week field, which would make the specification:

```
   0 0 10,14,18 ? * MON-FRI
```

Task processing is executed by scheduled task extensions. CM System includes four scheduled task extensions by default:

- sys_PurgePublishingLog

  This task purges the publishing log. The extension includes parameters that define how far back to preserve logs, and whether to archive logs before purging them.

- sys_PurgeSchedulde TaskLog

  This task purges the scheduled task log. The extension includes a parameter that defines how far back to preserve logs.

- sys_runCommand

  This task runs a native system command. The extension includes a parameter where you can define the command you want to run.

- sys_runEdition

  This task runs an Edition. The extension includes a parameter where you can define the Edition you want to publish.

(You can write a custom scheduled task extension if none of these extensions meet your needs. For details, see the *CM System Technical Reference Manual*.)

In this case, we will use the sys_runEdition extension, and specify the EI_Incremental Edition. We will run the Edition locally on the CM System system master. (We have not implemented a publishing hub in this scenario.)

We also need to determine the notification properties for the scheduled task. For the initial implementation, we want EI administrators to receive a notification each time the Edition is run. We will use the Run_Edition_Template, included with CM System by default.

We will name the scheduled task *EI_Incremental_Publish*.

To create the EI_Incremental_Publish Task:

1. Log in to Content Explorer and go to the Admin tab. Click on the Scheduled Tasks link.

   CM System displays the Scheduled Task List.

2. In the menu bar, choose *Action* > *Create Task*.

   CM System displays a Scheduled Task Editor with default values.

3. The value in the **Name** field defaults to *TimedEvent_0*. Change the **Name** to *EI_Incremental_Publish*.

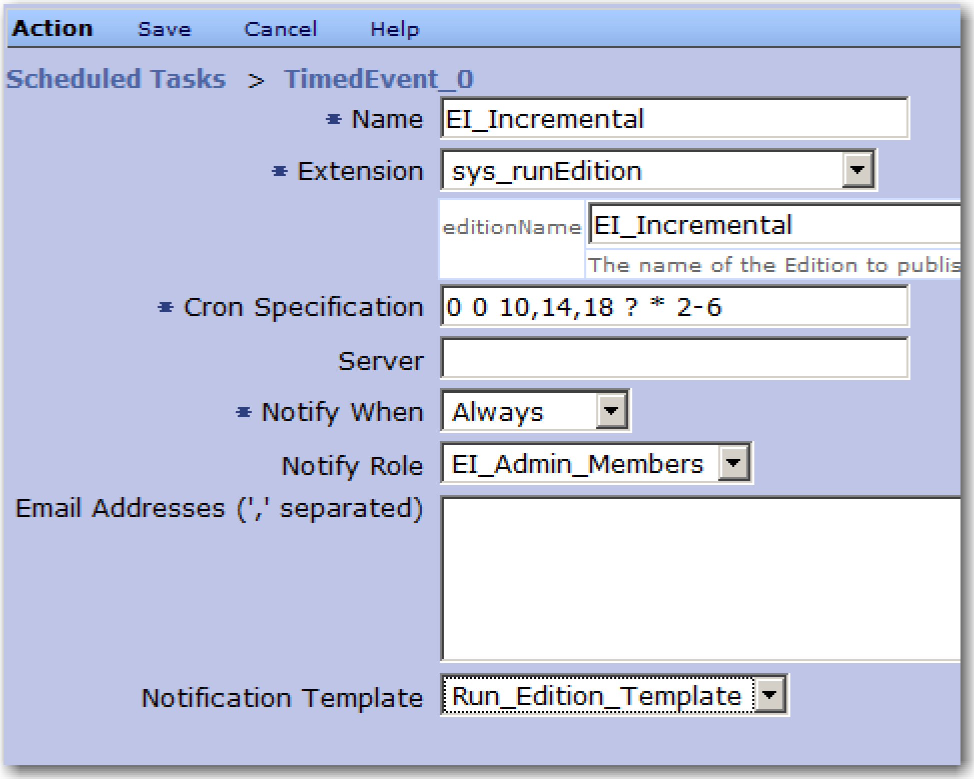
4. In the **Extension** drop list, choose *sys_runEdition*. When you choose this extension, CM System refreshes the page and displays the **Edition Name** field. Enter *EI_Incremental* in this field.

5. In the **Cron Specification** field, enter *0 0 10,14,18 ? * 2-6*. This code defines the the cron specification to implement the schedule we outlined above.

6. We want this task to run locally on the system master, so leave the **Server** field blank.

7. In the **Notify When** drop list. choose *Always*. In the **Notify Roles** field, choose EI_Admin_Members. Note that you could optionally add other **E-mail Addresses** to receive these notifications. Use commas to separate addresses.

8 In the **Notification Template** field, choose *Run_Edition_Notification*.



Scheduled Tasks > TimedEvent_0

* Name	EI_Incremental
* Extension	sys_runEdition
editionName	EI_Incremental The name of the Edition to publis
* Cron Specification	0 0 10,14,18 ? * 2-6
Server	
* Notify When	Always
Notify Role	EI_Admin_Members
Email Addresses (',' separated)	
Notification Template	Run_Edition_Template

Figure 21: EI_Incremental_Publish scheduled task

9 In the Menu bar, click *Save* to save the scheduled task.

Once you have created the scheduled task, you can test it. To test the scheduled task:

1 Open the EI_Incremental_Publish Scheduled Task it in the Scheduled Task Editor.

2 On the Menu bar, choose *Run Now*.

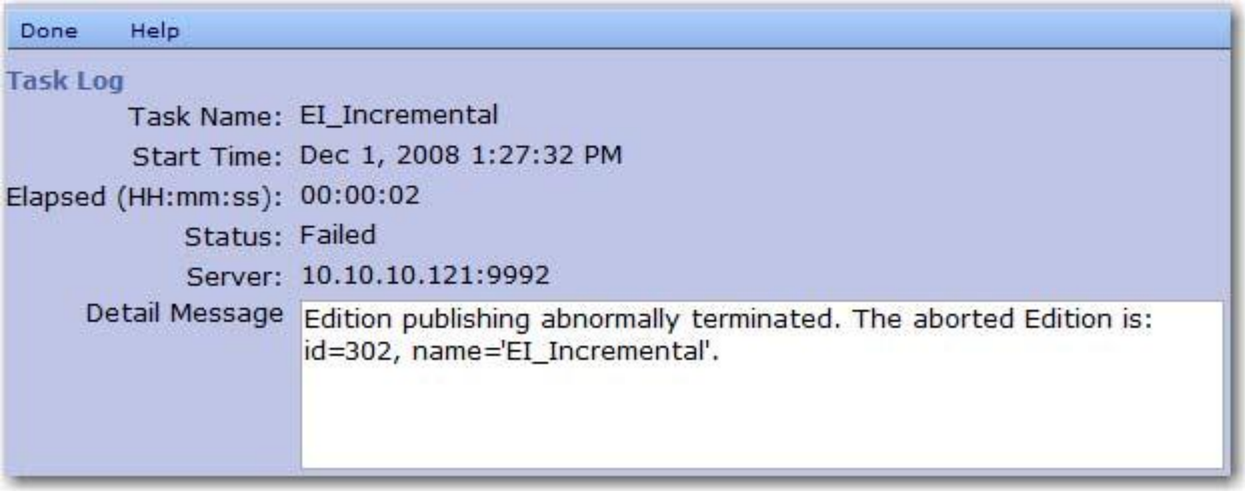
CM System runs the task.

3 Close the Scheduled Task Editor and click on the Task Logs link.

CM System displays the Task Log.

4 Click the EI_Incremental_Publish link in the row with the **Start Time** that matches the time that you ran the task.

CM System displays the Task Details page for the task.



Task Log

Task Name: EI_Incremental

Start Time: Dec 1, 2008 1:27:32 PM

Elapsed (HH:mm:ss): 00:00:02

Status: Failed

Server: 10.10.10.121:9992

Detail Message: Edition publishing abnormally terminated. The aborted Edition is: id=302, name='EI_Incremental'.

Figure 22: Task log details

In this example, you could also check the Publishing Log for the Edition to confirm the results of the Edition, and the e-mail of a member of the EI_Admin_Members Role to confirm that the e-mails were generated correctly.

Note: If we had defined the task to run on a different server (such as a publishing hub), running it from the system master would fail. To test a scheduled task configured to run on a remote server, start a browser, connect to the server, log in to Content Explorer, go to the Admin tab and run the task.

Copying a Scheduled Task

The Enterprise Investments implementation plan calls for a weekly scheduled task to run the full publish Edition. Most of the data for this scheduled task is identical to the EI_Implement_Publish scheduled task we just created, so we can create the new scheduled task quickly by copying the EI_Incremental_Publish scheduled task.

The new scheduled task, which we will call EI_Full_Publish, differs from the EI_Incremental_Publish scheduled task in two ways:

- it runs the EI_Full Edition rather than the EI_Incremental Edition; and
- it has a different Cron Specification to run a different schedule.

  A full publish typically takes substantially longer than an incremental publish, so it is typically run at times when demand on the system is lower, such as at night or over the weekend. In our example, we will run the EI_Full Edition at 11 PM on Saturday. The cron Specification for this schedule is:

```
0 0 23 ? * 7
```

To create the EI_Full_Publish scheduled task:

1. Log in to Content Explorer and go to the Admin tab. Click on the Scheduled Tasks link.

   CM System displays the Scheduled Task List in the View and Edit pane.

2. Select the radio button in the row of the EI_Incremental_Publish scheduled task.

3 In the Menu bar, choose *Action* > *Copy Scheduled Task*.

CM System copies the scheduled task and displays it in the Scheduled Task Editor.

4 The **Name** defaults to *Copy_of_EI_Incremental_Publish*. Change the **Name** to *EI_Full_Publish*.

5 Change the **editionName** field to *EI_Full*.

6 Change the **Cron Specification** to *0 0 23 ? * 7*.

7 In the Menu bar, click *Save*.

Now you can test the scheduled task as described in ***Defining a Publishing Schedule*** (on page 45).

Defining a Task Notification

A task notification is a template CM System uses to generate an e-mail to send when triggered by a scheduled task. The template lets you define the text for both the subject line and the body of the e-mail message.

The subject line is defined by a JEXL expression, the body by a Velocity Template. CM System resolves the code in each of these fields and uses the results to generate the notification e-mail. Because both fields include JEXL, you can include binding variables in the format $variablename; for example *$sys.completed*. Each Scheduled Task extension defines a set of binding variables that can be used when coding the subject and body templates.

NOTE: While JEXL and Velocity are related, they use different coding standards. Specifically, while a Velocity directive must begin with a # character, a JEXL function statement cannot. A common mistake when writing code in the Subject field of a task notification is to use the # character, which results in a processing error and a blank subject line in the output e-mail.

In this exercise, we will illustrate the creation of the Run_Edition_Template, which uses the sys_runEdition Scheduled Task extension.

For the subject of the e-mails, we want a simple notice: did the Edition run successfully. If so, we want the subject to resemble:

```
Publishing of the Edition <Edition Name> was completed successfully
```

If the Edition failed, we want a different subject line:

```
Publishing of the Edition <Edition Name> failed to complete
```

In pseudocode, this would be:

```
If publishing task is successful,
Publishing of the Edition <Edition Name> was completed successfully
Otherwise
Publishing of the Edition <Edition Name> failed to complete
```

The sys_runEdition Scheduled Task extension includes the following binding variables:

Variable Name	Type	Description
$editionName	String	Edition name, as defined by the editionName parameter of the extension.

$tools.		Velocity tools utilities available in CM System. Available utilities are defined in the file <Rhythmyxroot>/AppServer/server/rx/deploy/rxapp.ear/rxapp.war/ WEB-INF/config/velocity/tools.xml
$sys.taskName	String	The name of the task.
$sys.completed	Booelan	True if job processing was completed; otherwise false.
$sys.problemDesc	String	If processing was not completed, the description of the problem that cased the failure.
$sys.editionName	String	Name of the published Edition.
$sys.siteName	String	Name of the published Site.
$sys.editionLogUrl	String	URL that can be used to view the Edition log.
$sys.failureCount	String	Number of Content Items for which publishing failed.
$sys.successCount	String	Number of Content Items for which publishing succeeded.
$sys.executionDateTime	String	Starting date and time of processing of the extension.
$sys.executionElapsedTime	Long	The duration of the execution in milliseconds.

We can use the $sys.completed binding variable when testing the success of the publishing task, and use the $sys.editionName variable to add the Edition name to the string. We can use the JEXL if/else function for the conditional processing. The code for the Subject field is thus:

```
if ($sys.completed)
{'Publishing of the Edition ' + $sys.editionName + ' was completed
successfully';}
else
{'Publishing of the Edition '+ $sys.editionName + ' failed to
complete';}
```

In the body of the e-mail, the first information we want to deliver is the name of the Site that was published. We can use the $sys.siteName binding variable to provide this information:

```
Site: $sys.siteName
```

Next, we want to provide more details about the results of the run. If publishing of the Edition was completed, we want to show the number of Content Items that were published successfully, the number that failed, and the URL of the log where the recipient can access more detailed information. If the Edition failed, we want to provide a message describing the reason for the failure. We can use a Velocity #if/#else directive to provide the conditional processing; we can use the following binding variables to provide the data: $sys.successCount, $sys.failureCount, $sys.editionLogUrl, and $sys.problemDesc. The code for this section would be:

```
#if ($sys.completed)
Success Count: $sys.successCount
Failure Count: $sys.failureCount

Log URL: $sys.editionLogUrl
#else
The problem was: $sys.problemDesc
#end
```

Finally, we want to provide task processing details; specifically, when did processing start and how long did it take. We can use the $sys.executionDatetime and $sys.executionElapsedTime to provide this data. The value of the $sys.executionDatetime variable is a string, which we must convert to a standard date format using $tools.date.format:

```
Starting Time: $tools.date.format("yyyy-MM-dd HH:mm:ss",
$sys.executionDatetime)
```

The value of the $sys.executionElapsedTime variable is the number of milliseconds the processing required. To make it easy to read, we convert it to a standard hours:minutes:seconds format using the $tools.math.div:

```
Elapsed Time: $tools.number.format("00",
$tools.math.div($sys.executionElapsedTime,3600000)):$tools.number.format
("00",
$tools.math.div($sys.executionElapsedTime,60000)):$tools.number.format("
00", $tools.math.div($sys.executionElapsedTime,1000))
```

The complete template code is:

```
Site: $sys.siteName

#if ($sys.completed)
Success Count: $sys.successCount
Failure Count: $sys.failureCount

Log URL: $sys.editionLogUrl
#else
The problem was: $sys.problemDesc
#end

Starting Time: $tools.date.format("yyyy-MM-dd HH:mm:ss",
$sys.executionDatetime)
Elapsed Time: $tools.number.format("00",
$tools.math.div($sys.executionElapsedTime,3600000)):$tools.number.format
("00",
$tools.math.div($sys.executionElapsedTime,60000)):$tools.number.format("
00", $tools.math.div($sys.executionElapsedTime,1000))
```

To create the Run_Edition_Template:

1. Log in to Content Explorer and go to the Admin tab. Click on the Task Notifications link.

   CM System displays the Task Notifications List.

2. In the Menu bar, choose *Action* > *Create Task Notification*.

   CM System displays the Task Notification Editor with default values.

3. The **Name** defaults to *Notification_0*. Change the name to *Run_Edition_Template*.

4. The **Subject** defaults to *'Set a new subject'*. Enter the subject code described above into this field.

5. Enter the Velocity Template code defined above in the **Template** field.

6. In the Menu bar, click *Save*.

To test the notification, associate it with a scheduled task and run the task manually.

CHAPTER 5

Quality Assurance Testing the Production Environment

Once you have completed the installation and deployment of your production server and implemented publishing, you should perform a quality assurance test on the production implementation to confirm that it is performing as you expect. Percussion Software, Inc., recommends the following procedures:

- Test Content Editors and Content Assembly
  - Create a new Content Item with each Content Editor
  - Preview each Content Item using each available Template and Variant. If you use multiple global templates, preview each Content Item in each Template and Variant with each available global template.
  - Add related content to each Slot available on each Content Item.
  - Preview each Content Item again with each available Template and Variant. If you use multiple global templates, preview each Content Item in each Template and Variant with each available global template.
  - Pass one Content Item through all States in each Workflow. Confirm that each Content Editor is using the correct Workflow.
  - Repeat all tests for each Site in your implementation.
- Test Navigation Implementation
  - Add a new Folder to each Site. Move a Content Item in and assign it as a landing page. Preview Content Items from each additional Folder in the Site to ensure that the navigation is being built correctly.
  - Repeat the test for each Site in your implementation.
- Test publishing
  - If your production Web server is already in use, implement an alternative site to test publishing.
  - Publish content to the site to confirm that everything is publishing correctly and that Navigation is being assembled correctly in the published output.

CHAPTER 6

Going Live

Once you have completed QA of your system and confirmed that it is operating as expected, you are ready to take the final steps to going live:

- Adding content to the system;
- Training users on CM System.

Adding Content to Your Production Content Management System

Navigation Content Items may have been deployed into the production environment with your Folder structure, but adding the rest of your content is a manual process. Users will need to review the pages in your current Web site and add the content of those pages to the system as Content Items. They will need training to help them learn the difference between content that CM System manages and other portions of the page that will be derived from local or global templates in the published pages.

This is a prime opportunity to introduce CM System to an initial set of users. Consider asking some of your more technically savvy users to participate in this process. They will grasp the concepts more quickly and can help "evangelize" the product to other users. In addition, they are more likely to be tolerant of any technical issues that arise during the process of adding the content.

Text content must be added to the system manually (often using copy and paste functionality), but consider implementing WebDAV to simplify mass upload of graphics and other binary Content Items. For details about using WebDAV, see the document *Implementing WebDAV in CM System*. Note, however, that once this content is added to CM System, users will still have to manually add it to text Content Items
to assemble the desired output pages. Users will also have to create any other Active Assembly Relationships between text Content Items after adding them to the system.

Training Users on CM System

Before putting your system into production, you should train you users on CM System. At the very least, distribute the *CM System Concepts Guide* to them and emphasize the sections most relevant to the tasks they will be performing. Ideally, however, you should develop and plan a training class that outlines in rough terms how CM System works, the elements of your implementation, and common tasks in Content Explorer.

Putting CM System into Production

Once you have added your content to the system and trained your users, you are ready to put CM System into production to maintain your Web site. Pick a date and launch your first production publishing run.

Congratulations! Welcome to Content Management with CM System!

Index

A

Active Directory • 22, 29
Adding Content to Your Production Content Management System • 56
Adding Users to CM System Roles • 37
Authentication • 25

B

Backend Table Security Provider • 33

C

Changing the Default Server Access Control List • 39
Copying a Scheduled Task • 48
Creating a Directory Connection Security Provider • 31

D

Datasource • 13
Default Security Settings • 38, 39
Defining a Directory for a Directory Services Configuration • 27
Defining a Directory Set for a Directory Services Configuration • 29
Defining a Publishing Schedule • 45, 49
Defining a Role Provider for a Directory Services Configuration • 30
Defining a Task Notification • 49
Defining an Authentication for a Directory Services Configuration • 25
Defining Security Providers • 20
Deploying a CM System Environment • 9, 11
Deploying an implementation • 11, 12, 13, 15, 16, 17, 18
Deployment Process • 12
Development Tier • 8
Directory (for Directory Services Configuration) • 27
Directory Connection Security Provider • 22, 25, 27, 29, 30, 31, 32
Directory Sets • 29

G

Going Live • 9, 55, 56, 57, 58

I

Implementing a Backend Table Security Provider • 33
Implementing a Directory Connection Security Provider • 21
Implementing a Tiered CM System Environment • 8
Implementing a Web Server Security Provider • 36
Implementing Security in the Production Environment • 9, 19, 39
Integration Tier/Server • 8

L

LDAP • 22, 25, 27, 29, 30
LDAP Configuration • 22

N

Notification • 49

P

Process Overview • 9
Production Environment • 7
    Procedure for Setting Up • 9
Production Tier • 8
Publishing • 41
    mapped drive • 42
    mounted drive • 42
Publishing to a Mapped or Mounted Drive • 42
Putting CM System into Production • 58

Q

Quality Assurance Testing • 53
Quality Assurance Testing the Production Environment • 9, 53

R

Replacing the Default Users with New Users • 39
Role Providers • 30
Roles
    adding users to • 37

S

Scheduled Publishing • See Scheduled Task
Scheduled Task • 45, 48, 49

Scheduling Publishing • 45
Security • 19, 20, 21, 22, 25, 27, 29, 30, 31, 32, 33, 36, 37, 38, 39
Security Provider • 19, 20, 21, 33, 36
  Backend Table Security Provider • 33
  Directory Connection Security Provider • 22, 25, 27, 29, 30, 31, 32
  Web Server Security Provider • 36
Servlet • 16
Setting Up Publishing in the Production Environment • 9, 41
Setting Up the Production Environment • 7
Special Deployment Case • 17
Spring • 15

T

Task • See Scheduled Task
Task Notification • See Notification
Testing the Directory Connection Security Provider • 32
Tiered Environment • 8
Training Users on CM System • 57

U

Updating Default Security Settings • 38
Updating the Datasource • 12, 13
Updating the Deployed Environment • 18
Updating the CM System Servlet Configuration • 12, 16
Updating the Spring Configuration • 12, 15
USERLOGIN • 39
Users
  adding to a Role • 37

W

Web Server Security Provider • 36