

About OpenTravel:

The OpenTravel Alliance provides a community where companies in the electronic distribution supply chain work together to create an accepted structure for electronic messages, enabling suppliers and distributors to speak the same interoperability language, trading partner to trading partner. Tens of thousands of the OpenTravel message structures are in use, carrying tens of millions of messages between trading partners every day.

Members do the work of identifying what messages are needed, prioritize the work and collaborate to create the messages. Members who are looking for more information on related project team work or who wish to access the OTM repository can send inquiries to architecture@opentravel.org.

Note: This document supports implementers using the OTM-DE Model Builder in the creation and sharing of models that automatically generate xml schema. The ability to access and extend the OpenTravel Model is available only to OpenTravel members. For more information please contact us at membership@opentravel.org.

Document Purpose:

This guide provides system and application administrators with the information required to install, configure and maintain a remote OTM repository web service application.

OTM Repository Installation Administration Guide

Document Revision 1.0
Software Version 3.0

July 5, 2013

Contents

INTRODUCTION.....	3
1 REPOSITORY INSTALLATION.....	3
1.1 INSTALLING TOMCAT.....	3
1.2 INSTALLING THE OTM WEB SERVICE APPLICATION	3
1.3 CONFIGURING THE REPOSITORY.....	4
1.3.1 <i>Select Folder Locations for Repository Content and Search Index.....</i>	<i>4</i>
1.3.2 <i>Install the Initial Repository Content Files</i>	<i>5</i>
1.3.3 <i>Assign a Unique ID for the Repository</i>	<i>5</i>
1.3.4 <i>Specifying the Repository Location</i>	<i>5</i>
1.3.5 <i>User Account Management Configuration</i>	<i>6</i>
1.3.6 <i>Repository File Management</i>	<i>10</i>
1.3.7 <i>Creating the Administrator Account</i>	<i>12</i>
1.3.8 <i>Starting the Repository Web Service.....</i>	<i>13</i>
2 REPOSITORY ADMINISTRATION	13
2.1 MODIFYING THE REPOSITORY NAME.....	14
2.2 MANAGING ROOT NAMESPACES	15
2.3 CREATING AND DELETING NAMESPACES	16
2.4 MANAGING USER ACCOUNTS (LOCAL USER MANAGEMENT ONLY).....	17
2.5 MANAGING GROUP ASSIGNMENTS	19
2.6 MANAGING NAMESPACE PERMISSIONS	21
2.7 REFRESHING THE FREE-TEXT SEARCH INDEX	23
2.8 MANAGING OTM ARTIFACTS.....	23
3 APPENDIX: REPOSITORY FILE SYSTEM LAYOUT	25

Change History

Revision	Author(s)	Summary of Changes
1.0	S. Livezey	Initial Draft

Introduction

The OTM Repository is a component of the OTM Development Environment (OTM-DE) that facilitates sharing and collaborative development for OpenTravel models. The OTM Repository comes in two varieties: remote and local. Remote repositories are used to publish, share, and manage OTM models. They are accessible by multiple users and can support free-text searching, as well as complex security and access control policies. Local repositories, on the other hand, are located in the home directory of each user who accesses a remote repository. In addition to maintaining local copies of OTM models that have been downloaded from remote repositories, local repositories can be used to manage personal models that have been authored by an OTM-DE user, but are not yet ready for general publication.

This guide provides system and application administrators with the information required to install, configure, and maintain a remote OTM repository web service application.

1 Repository Installation

For the current release of the OTM-DE software suite, the only web service container approved for the hosting of an OTM Repository web service is Apache Tomcat (version 7.0 or later). It should be possible to deploy the repository software in any J2EE-compliant web application server, but some changes to the configuration settings described in this guide are likely to be required.

1.1 Installing Tomcat

Prior to installing the OTM repository web service application, an Apache Tomcat server must be installed and properly configured for operation. To accomplish this task, please refer to the online setup and installation guide for Tomcat. The guide is available at the following location:

<http://tomcat.apache.org/tomcat-7.0-doc/setup.html>

1.2 Installing the OTM Web Service Application

Once the Apache Tomcat server has been installed, the only step required to install the OTM repository software is to copy the `ota2-repository-service.war` file into the `/webapps` folder of the Tomcat installation. You will also need to copy the `ota2-repository-config.xml` file into your Tomcat `/conf` folder. *Before starting the Tomcat server, however, it is important to properly configure to the OTM repository by following the instructions in section 1.3.*

1.3 Configuring the Repository

Configuring the OTM Repository will typically require some editing of the `ota2-repository-config.xml` file located in the `/conf` directory of the Tomcat web application server. An example listing of this file has been provided in Appendix A of this document; the file is also downloadable from the OpenTravel web site.

Basic configuration tasks can be divided into four major categories:

1. Select the folder locations for the repository content and search index
2. Install the initial repository configuration files
3. Assign a unique ID for the repository
4. Configure the location of the repository's root directory on the server's file system
5. Specify the method to be employed for identification and management of OTM user accounts
6. Configure the method for controlling and maintaining files in the repository's file system directory
7. Create the Administrator Account
8. Start the repository web service
9. Access the OTM Repository web console

1.3.1 Select Folder Locations for Repository Content and Search Index

Besides the location of your Apache Tomcat installation, the two most important folder locations are the path to root directory of the repository content and the folder location of your repository's search index. Before proceeding with configuration of the OTM repository, these folder locations should be identified and created on the server's file system.

If Subversion file management is to be used for this repository (see section 1.3.6.2), the repository content folder must be under SVN control. To accomplish this, first create a Subversion repository with an empty folder where the OTM file content will be managed. This can be done using the Subversion command-line client or some other GUI application such as TortoiseSVN or Subclipse. Once the SVN repository folder exists, the empty folder should be checked out to a folder that is accessible from the OTM server's file system.¹ Because the repository's free-text search can be re-indexed at any time, it is not necessary to place your search index files under Subversion control.

¹ Hint: To be certain this was done correctly, you should see a `/.svn` directory in the root folder of your server's repository. On many operating systems, this folder will be hidden unless the viewing of hidden files/folders is explicitly enabled.

1.3.2 Install the Initial Repository Content Files

The initial content includes several administrative files that define repository identification, settings, user accounts and group assignments, security permissions. These files are included with the `initial-repository-content.zip` file that is bundled with the OTM repository application files. To install these files, simply unzip them into the root directory of the repository content identified in section 1.3.1. If Subversion file management is to be utilized for the repository it is not required that these files be committed to SVN since that task will be done automatically when the Apache Tomcat service is started for the first time.

1.3.3 Assign a Unique ID for the Repository

The ID of a repository identifies it to all clients who may utilize its content. If multiple repositories are to be active within an enterprise, it is important for each repository's ID to be globally unique.

To assign the unique repository ID, edit the `repository-metadata.xml` file in the root folder of the repository's content directory. Within this file, simply change the value inside the `<otp:ID>` tag to be whatever globally unique identifier is to be assigned.

```

<otp:RepositoryInfo>
  <otp:ID>xyz-repository</otp:ID>
  <otp:DisplayName>OTA2.0 XYZ Company Repository</otp:DisplayName>
  <otp:RootNamespace>http://www.xyzcompany.com</otp:RootNamespace>
  <otp:RemoteRepositories>
  </otp:RemoteRepositories>
</otp:RepositoryInfo>
  
```

1.3.4 Specifying the Repository Location

The files contained in and managed by an OTM repository are contained within a single directory that must be accessible from the file system of the machine where the Tomcat server is running. There are two principle folder locations that must be specified in this configuration: the root folder of the repository itself, and the root folder of the search index that is used for free-text searching.

These folder locations are specified in the `ota2-repository-config.xml` configuration file as follows:

```
<bean id="repositoryLocation" class="java.io.File" scope="singleton">
  <constructor-arg value="/users/local/myuserid/ota2/repository" />
</bean>

<bean id="searchIndexLocation" class="java.io.File" scope="singleton">
  <constructor-arg value="/users/local/myuserid/ota2/search-index" />
</bean>
```

1.3.5 User Account Management Configuration

While the OTM repository can certainly be configured to support anonymous user access, it is often desirable to configure more prescriptive security rules for publishing and updating OTM models. To facilitate these security features, the OTM repository supports two mechanisms for user identification and authorizations: local account management and directory account management. When configuring an OTM repository one and only one of these two methods must be employed.

1.3.5.1 Local Account Management

Local account management provides a way to define and manage user accounts that are only meaningful for a single repository instance. This is often useful for development systems or small organizations who do not already maintain a global directory of users.

To configure the repository for local account management, use the following settings for the 'authenticationProvider' bean in the `ota2-repository-config.xml` file:

```
<bean id="authenticationProvider"
  class="com.sabre.schemacompiler.security.impl.FileAuthenticationProvider">
  <constructor-arg ref="repositoryLocation" />
</bean>
```

1.3.5.2 Directory Account Management

Directory account management provides a way to use an organization's existing domain login accounts to establish user access to an OTM repository. There are three principle modes available for configuring connectivity to a corporate directory. The best one to use depends on the specifics of the directory's configuration.

The available directory configuration modes are:

- **User Authentication Mode** – In this mode, each user's credentials are used to attempt a login to the remote directory server. This approach is sometimes considered more secure because it does not require an LDAP administrator's password to be stored with the configuration settings of the repository. In some cases, however, this mode is not possible because user accounts in a corporate directory are not granted permission to login to the LDAP server itself.

- **User Lookup Mode** – In user lookup mode, an authenticated user (typically an LDAP administrator) is used to establish all connections to the remote directory. User accounts are identified by a distinguished name format that is the same for all users defined in the directory. Once identified, encrypted password credentials are retrieved from the directory and compared with the credentials provided by the remote user of the repository.
- **User Search Mode** – Like user-lookup, this mode of operation establishes remote connections using a single authenticated user account. User accounts are located by searches within the directory using one or more configurable query strings. Once user accounts are located by a search, the user's encrypted password credentials are retrieved from the directory and compared with the credentials provided by the remote user of the repository.

The configuration settings for each of these authentication modes should be applied to the 'authenticationProvider' bean in the file as follows.

```
<bean id="authenticationProvider"
  class="com.sabre.schemacompiler.security.impl.JNDIAuthenticationProvider">
  <property name="connectionUrl" value="ldap://somecompany.com" />
  <property name="securityAuthentication" value="simple" />
  <property name="userPattern" value="GLOBAL\{0}" />
  <property name="referralStrategy" value="follow" />
</bean>
```

The available configuration settings and their applicability to each of the three authentication modes is described in Table 1 below:

Property Name	Description	User Auth. Mode	User Lookup Mode	User Search Mode
contextFactory	Fully qualified Java class name of the factory class used to acquire our JNDI InitialContext. By default, assumes that the standard JNDI LDAP provider will be utilized.	Optional	Optional	Optional
connectionUrl	The connection URL to be passed to the JNDI driver when establishing a connection to the directory.	Required	Required	Required

Property Name	Description	User Auth. Mode	User Lookup Mode	User Search Mode
alternateUrl	If a socket connection cannot be made to the provider at the connectionURL an attempt will be made to use this address.	Optional	Optional	Optional
connectionProtocol	A string specifying the security protocol to use. If not given the providers default is used.	Optional	Optional	Optional
securityAuthentication	A string specifying the type of authentication to use. "none", "simple", "strong" or a provider specific definition can be used. If no value is given the providers default is used.	Optional	Optional	Optional
connectionTimeout	The timeout in milliseconds to use when establishing the connection to the LDAP directory. If not specified, a value of 5000 (5 seconds) is used.	Optional	Optional	Optional
authenticationCacheTimeout	The amount of time (in milliseconds) that the results of a user's login attempt should be cached. Default value is 5 minutes.	Optional	Optional	Optional
connectionPrincipal	The directory username to use when establishing a connection to the directory for LDAP search and lookup operations. If not specified an anonymous connection is made, which is often sufficient unless you specify the connectionPassword property.	N/A	Required	Required

Property Name	Description	User Auth. Mode	User Lookup Mode	User Search Mode
connectionPassword	The directory password to use when establishing a connection to the directory for LDAP search and lookup operations. If not specified an anonymous connection is made.	N/A	Required	Required
userPattern	Pattern for the distinguished name (DN) of the user's directory entry, with {0} marking where the actual username should be inserted.	Required	Required	N/A
userSearchBase	The base element for user searches performed using the 'userSearchPatterns' expressions.	N/A	N/A	Required
searchUserSubtree	Set to true if you want to search the entire subtree of the element specified by the 'userSearchBase' property for the user's entry. The default value of false causes only the top level to be searched.	N/A	N/A	Optional
userSearchPatterns	A colon-separated list of LDAP filter expressions to use when searching for a user's directory entry, with {0} marking where the actual username should be inserted.	N/A	N/A	Required
userSearchTimeout	Specifies the time (in milliseconds) to wait for records to be returned when employing the user-search mode of operation. If not specified, the default of 0 is used which indicates no limit.	N/A	N/A	Optional

Property Name	Description	User Auth. Mode	User Lookup Mode	User Search Mode
userPasswordAttribute	Specifies the name of the attribute where passwords are stored on user entries. If not specified, a default value of "userPassword" is assumed.	N/A	Optional	Optional
referralStrategy	Specifies the strategy for JNDI referrals; allowed values are "ignore", "follow", or "throw" (see javax.naming.Context.REFERENTIAL for more information). Microsoft Active Directory often returns referrals. If you need to follow them set referrals to "follow". Caution: if your DNS is not part of AD, the LDAP client lib might try to resolve your domain name in DNS to find another LDAP server.	Optional	Optional	Optional
digestAlgorithm	The digest algorithm to apply to the plaintext password offered by the user before comparing it with the value retrieved from the directory. Valid values are those accepted for the algorithm name by the java.security.MessageDigest class. If not specified the plaintext password is assumed to be retrieved.	N/A	Required	Required
digestEncoding	The encoding character set to use when applying the digest algorithm.	N/A	Optional	Optional

1.3.6 Repository File Management

The final aspect of repository configuration involves selecting the method of file management that will be employed by the server. For an OTM repository server, the

managed content (typically OTM models) is stored on the locally accessible file system. The file management strategy specifies the steps that will be employed to store and update persistent repository data, as well as how complex interactions such as multi-file transactions and exception handling will be addressed.

There are currently two strategies available for handling files in an OTM repository – local file management and Subversion file management.

1.3.6.1 Local File Management

The local file management strategy for OTM content is relatively straightforward. Whenever files are modified or deleted, a backup is first created for each of the affected files. Once the processing of a change has been completed, all of the backup files are deleted from the local file system. If an error occurs, the changes are rolled-back by deleting the modified files and restoring the backups.

Local file management is the default configuration of the ‘repositoryManager’ bean in the `ota2-repository-config.xml` file.

```
<bean id="repositoryManager"
      class="com.sabre.schemacompiler.repository.RepositoryManager"
      scope="singleton">
  <constructor-arg ref="repositoryLocation" />
</bean>
```

1.3.6.2 Subversion File Management

Subversion file management is similar to local management in that repository content is still accessed via the server’s local file system. The principle difference is that the local files are maintained in the local checkout directory of a Subversion client. Subversion is typically used in software development projects for source code control and software configuration management (SCM). For OTM repositories, Subversion simply serves as a persistent storage manager for the files that are maintained within the repository.

Since the root folder of the repository should already be correctly configured (as per section 1.3.2), the ‘repositoryManager’ bean of the `ota2-repository-config.xml` file can now be configured as follows:

```

<bean id="repositoryManager"
      class="com.sabre.schemacompiler.repository.RepositoryManager"
      scope="singleton">
  <constructor-arg>
    <bean
      class="com.sabre.schemacompiler.repository.SVNRepositoryFileManager">
      <constructor-arg ref="repositoryLocation" />
      <constructor-arg ref="svnConfigFolder" />
      <constructor-arg ref="svnCredentialsFile" />
    </bean>
  </constructor-arg>
</bean>

<bean id="svnConfigFolder" class="java.io.File" scope="singleton">
  <constructor-arg value="#{systemProperties['user.home']}" />
  <constructor-arg value="/.subversion" />
</bean>

<bean id="svnCredentialsFile" class="java.io.File" scope="singleton">
  <constructor-arg value="#{systemProperties['catalina.base']}" />
  <constructor-arg value="/conf/svnCredentials.properties" />
</bean>

```

In the example configuration above, the 'svnConfigFolder' bean provides the location of the Subversion configuration directory. Typically, this will be located in the home directory of the user who owns the process for the Apache Tomcat server.

In many cases, it is also useful to specify the user ID and password credentials that will be used to access files in the Subversion repository. In the above example, the 'svnCredentialsFile' bean specifies the location of this file. The format of the credentials file is as follows:

```

svn.userid=ota2user
svn.password=password

```

NOTE: Because the password in the SVN credentials file is specified in plain-text, it is important to restrict the visibility and access to this file to the user(s) who will own the Apache Tomcat process for the OTM repository server.

1.3.7 Creating the Administrator Account

If local account management is being employed, a default administrator account is already included in the `initial-repository-contents.zip` file that was installed previously. The ID of the administrator user is 'admin' and the initial password

for the account is 'password'. At a minimum, the password should be changed when the repository web service is first started.

If directory account management is being used, the group-assignments.xml file should be edited, and the 'admin' user ID in the <Member> element should be changed to the login ID of the user who will administer the OTM repository.

```
<GroupAssignments>
  <Group name="Administrators">
    <Member>admin</Member>
  </Group>
</GroupAssignments>
```

1.3.8 Starting the Repository Web Service

Once all of the OTM repository configuration tasks are complete, all that remains is to launch the repository web service. This task is easily accomplished by running the /bin/startup.sh script (or /bin/startup.bat for Windows) in the Apache Tomcat installation directory.

To ensure that everything is running correctly, check the Tomcat server logs (initially configured to be located at /logs/catalina.out). If any errors are displayed in the log, re-check the configuration settings and re-start the server.

As a final check to ensure the OTM repository is running correctly, open a web browser to the following address and login using the administrator account.

<http://myserver.com:8080/ota2-repository-service>

NOTE: In the above URL, you will need to change the server address and port number to match the installation address of your Apache Tomcat server.

2 Repository Administration

The run-time administration of an OTM repository is done using the web application console. For OTM users, the web console provides an online portal for browsing and searching the contents of the repository. Administrative users have a number of additional functions available for the management of repository users and artifacts.

Generally speaking the artifacts that are managed by an OTM repository are organized hierarchically according to the namespace to which each library, schema, or project is assigned. This is a fancy way of saying that namespaces are like a folder structure, and each of the artifacts managed in the repository is assigned to a location in that structure. Administrative users have the ability to define access control policies for the various namespaces that are managed by a repository.

The primary responsibilities of an OTM repository administrator include the following tasks:

- Defining the name and root namespaces of the repository
- Managing user accounts and group assignments
- Managing user permissions for the repository's namespaces
- Maintaining the index used for free-text searching
- Managing the lifecycle state of OTM managed artifacts

When a user has logged into the web console using an administrator account, he/she can access the main administration page by clicking on the 'Administration' link in the upper-right corner of the page.

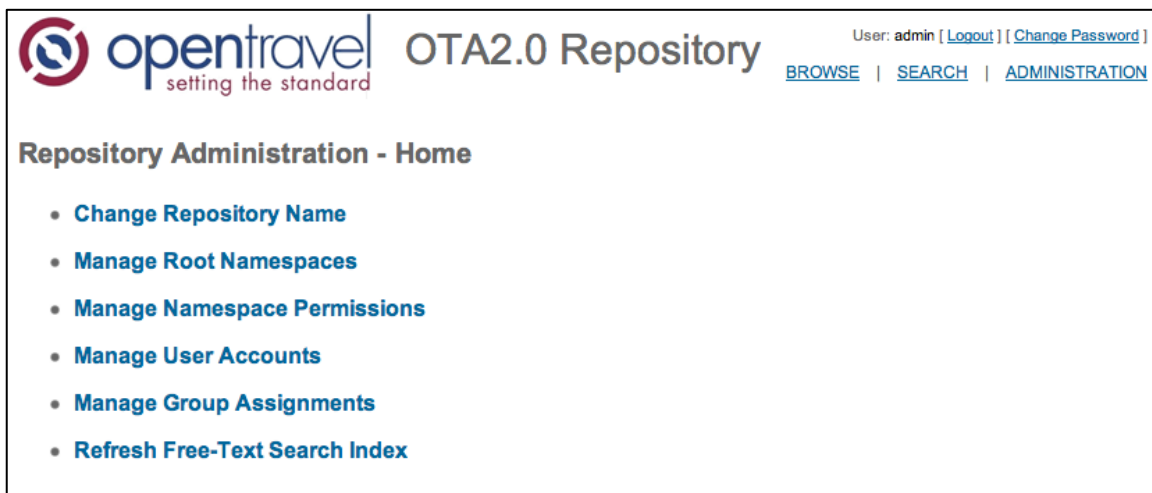


Figure 1: OTM Repository Administration Home

2.1 Modifying the Repository Name

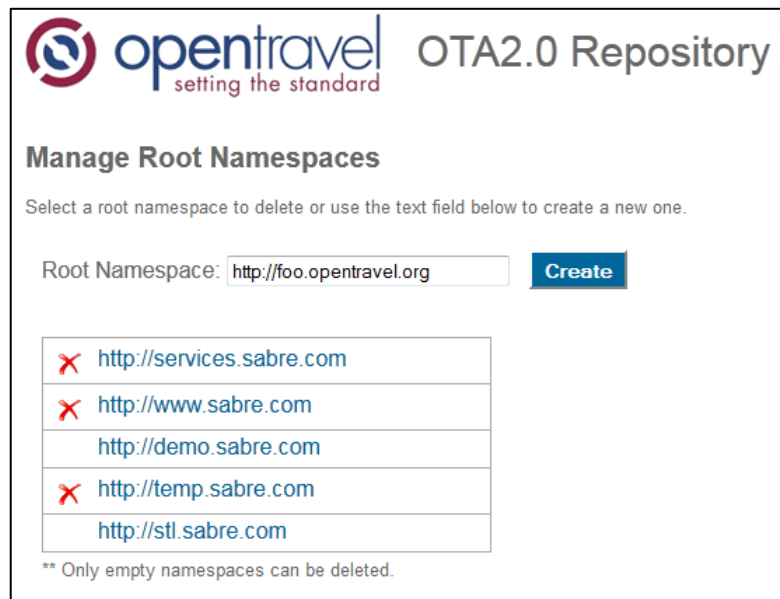
Modifying the display name for the repository is a relatively simple task. After clicking on the 'Change Repository Name' link of the main administration page, just modify the existing display name and click the update button.



Figure 2: Modifying the Display Name of a Repository

2.2 Managing Root Namespaces

If the namespace hierarchy is the folder structure of a repository, then the root namespaces can be thought of as the drives or top-most level in the structure. Root namespaces can be easily created and removed by clicking the 'Manage Root Namespaces' link on the main administration page.



opentravel setting the standard **OTA2.0 Repository**

Manage Root Namespaces

Select a root namespace to delete or use the text field below to create a new one.

Root Namespace:

<input type="checkbox"/>	<input type="checkbox"/>	http://services.sabre.com
<input type="checkbox"/>	<input type="checkbox"/>	http://www.sabre.com
<input type="checkbox"/>	<input type="checkbox"/>	http://demo.sabre.com
<input type="checkbox"/>	<input type="checkbox"/>	http://temp.sabre.com
<input type="checkbox"/>	<input type="checkbox"/>	http://stl.sabre.com

** Only empty namespaces can be deleted.

Figure 3: Managing Root Namespaces of a Repository

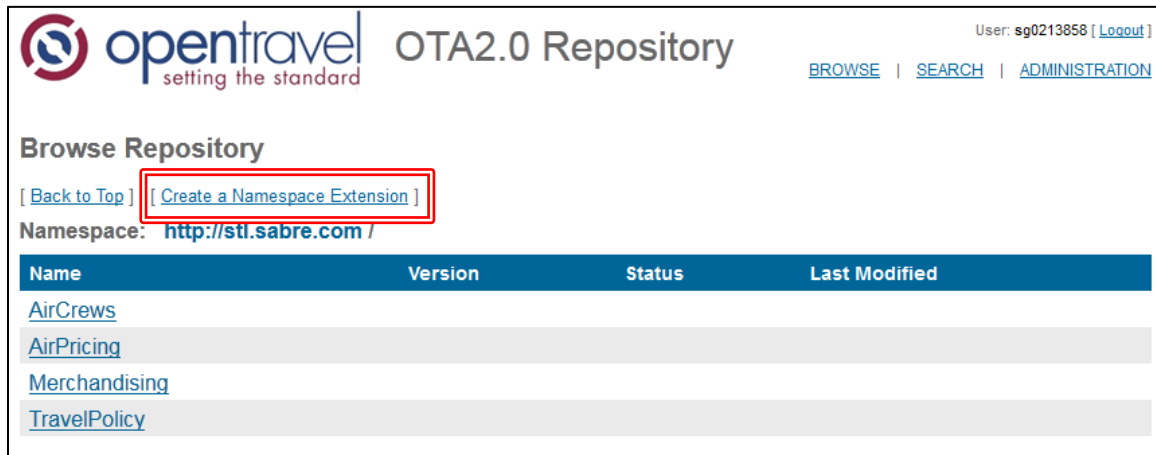
To create a new root namespace, simply type the URI into the text box and click the 'Create' button. To delete an existing root namespace, simply click the 'X' next to the URI and confirm the deletion by clicking 'Ok'. Note that only namespaces that do not contain sub- namespaces or artifacts are eligible for deletion.

The following business rules apply when creating new root namespaces:

- Root namespaces must conform to a proper URL format
- Root namespaces cannot be nested within one another (e.g. the root namespaces 'http://services.opentravel.com' and 'http://opentravel.com' cannot co-exist).

2.3 Creating and Deleting Namespaces

Unlike the creation and deletion of root namespaces, the management of namespace extensions (i.e. child namespaces or sub-folders) can be done by any user who has write access to a namespace. The create/delete namespace functions can be accessed by browsing to the appropriate namespace view. If the user has write permissions for the namespace a link titled 'Create a Namespace Extension' will be displayed for the user. By clicking this link, the user will be allowed to enter the path of the namespace extension to be added.



opentravel setting the standard OTA2.0 Repository User: sg0213858 [Logout] BROWSE | SEARCH | ADMINISTRATION

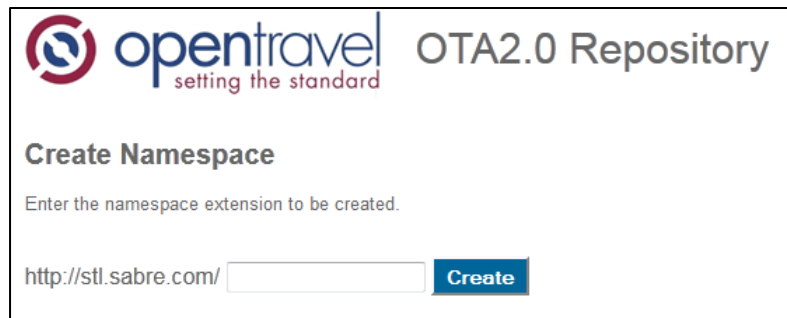
Browse Repository

[Back to Top] [Create a Namespace Extension]

Namespace: [http://stl.sabre.com /](http://stl.sabre.com/)

Name	Version	Status	Last Modified
AirCrews			
AirPricing			
Merchandising			
TravelPolicy			

Figure 4: Create Namespace Link on Browse Page



opentravel setting the standard OTA2.0 Repository

Create Namespace

Enter the namespace extension to be created.

<http://stl.sabre.com/>

Figure 5: Create Namespace Extension

To delete an existing namespace from the repository, the user should navigate to the browse page for the namespace to be deleted. If he/she has write access to this namespace, a link labeled 'Delete This Namespace' will be displayed.

Note that namespaces may only be deleted if they do not contain any child namespaces or managed OTM artifacts. Notice that when deleting a namespace, only the lowest level of the path is deleted.

opentravel setting the standard OTA2.0 Repository User: sg0213858 [Logout] [BROWSE](#) | [SEARCH](#) | [ADMINISTRATION](#)

Browse Repository

[[Back to Top](#)] [[Create a Namespace Extension](#)] [[Delete This Namespace](#)]

Namespace: <http://stl.sabre.com / TravelPolicy/>

Name	Version	Status	Last Modified
No items to display for this namespace.			

Figure 6: Delete Namespace Link on Browse Page

opentravel setting the standard OTA2.0 Repository

Delete Namespace

Delete namespace "http://stl.sabre.com/TravelPolicy". Are you sure?

[Delete](#)

Figure 7: Confirm Deletion of Namespace

2.4 Managing User Accounts (Local User Management Only)

If local user management has been configured for the repository, Administrators may add, delete, and change the passwords for user accounts by clicking the 'Manage User Accounts' link on the main administration page.

opentravel setting the standard OTA2.0 Repository

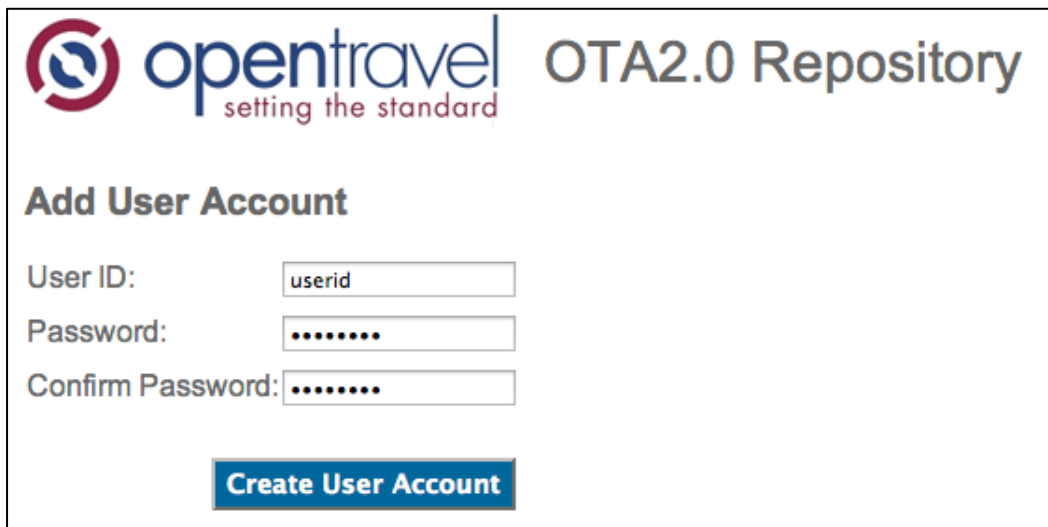
Manage User Accounts

admin			user2			user1				
-------	--	--	-------	--	--	-------	--	--	--	--

[Add a New User](#)

Figure 8: Manage User Accounts Administration Page

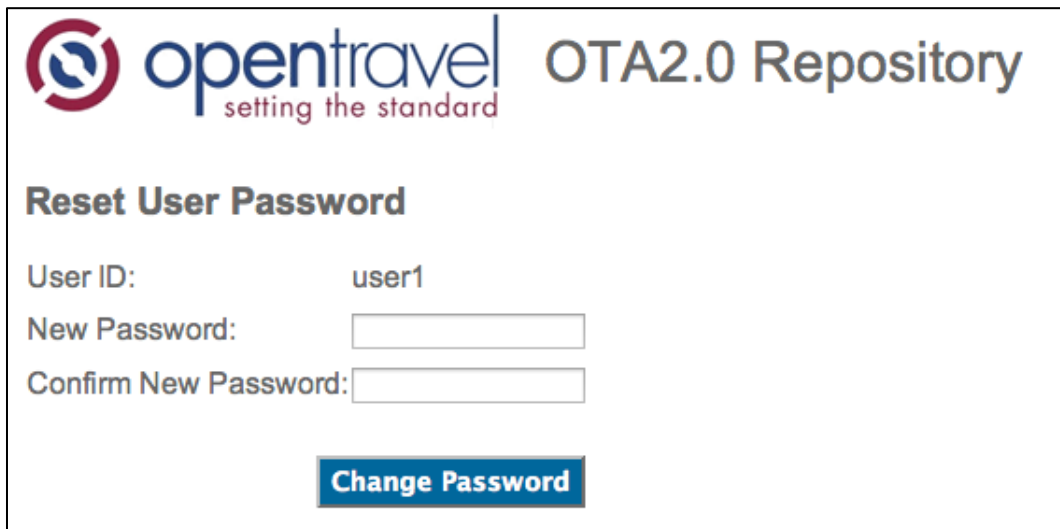
To add a new user account to the repository, click the 'Add a New User' link and enter the user's ID and password.



The screenshot shows the 'Add User Account' form. At the top left is the OpenTravel logo with the tagline 'setting the standard'. To the right is the text 'OTA2.0 Repository'. Below the logo is the heading 'Add User Account'. The form contains three input fields: 'User ID:' with the value 'userid', 'Password:' with masked characters '.....', and 'Confirm Password:' with masked characters '.....'. A blue button labeled 'Create User Account' is positioned at the bottom center of the form.

Figure 9: Add User Account

To change or reset the password for a user's account, the Administrator can click the pencil (edit) icon next to the user's ID on the 'Manage User Accounts' page.



The screenshot shows the 'Reset User Password' form. At the top left is the OpenTravel logo with the tagline 'setting the standard'. To the right is the text 'OTA2.0 Repository'. Below the logo is the heading 'Reset User Password'. The form contains three input fields: 'User ID:' with the value 'user1', 'New Password:' with an empty field, and 'Confirm New Password:' with an empty field. A blue button labeled 'Change Password' is positioned at the bottom center of the form.

Figure 10: Reset User Password

Deleting a user account from the repository is accomplished by clicking the 'X' icon next to the user's ID on the 'Manage User Accounts' administration page. The user's repository account will be deleted once the administrator confirms the action.

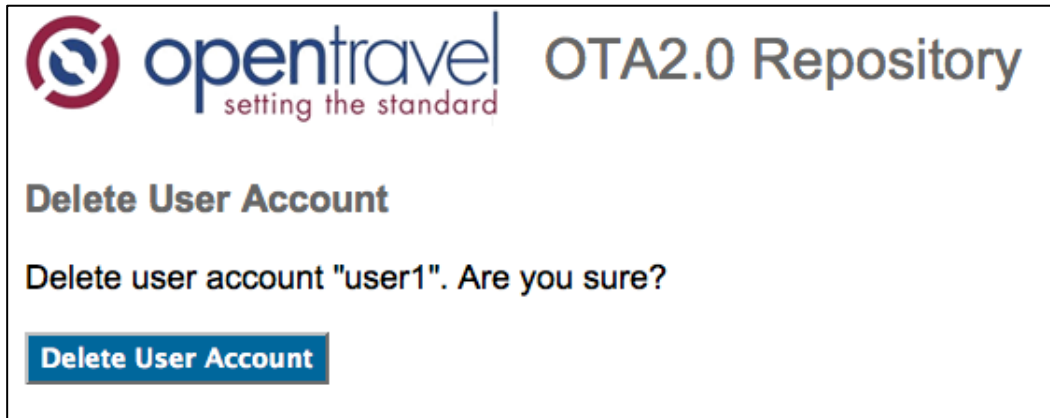


Figure 11: Delete User Account

2.5 Managing Group Assignments

In an OTM repository, user assignments to groups are managed using the Group Assignments administration page. This is true even if the repository has been configured for directory account management (see section 1.3.5.2).²

To view all of the current group assignments, start by clicking the 'Manage Group Assignments' link on the main administration page. To view the members of each group, simply click on the radio button next to each group name. New groups may be added by clicking the 'Add a New Group' link and entering the name of the new OTM user group. To delete an existing group, click the red 'X' icon to the right of the groups name and confirm the deletion.

Note that all members of the 'Administrators' group are automatically granted administrative access to the repository, including all of the administrative tasks that are described in this chapter.

² This is different from many systems that obtain user account information from a corporate directory. Group assignments for OTM repositories are always managed locally, and do not depend on any group or role assignments that might be defined within the corporate directory.



Figure 12: View/Manage Group Assignments

To modify the members of a group, click the pencil (edit) icon to the right of the group’s name. To add users to the group, you can either select a user ID from the list of existing accounts, or enter a new ID in the free-text space provided. Notice that the latter option is only available when directory user management has been configured.

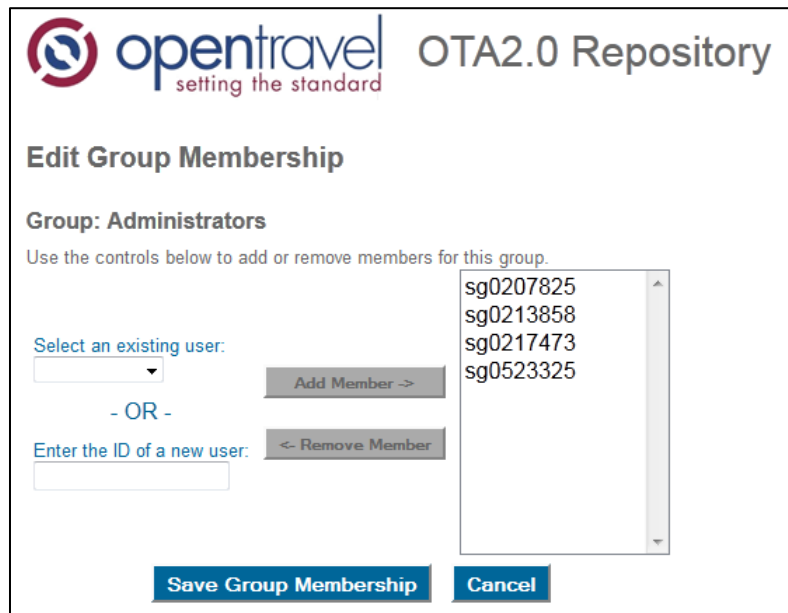


Figure 13: Edit Group Membership

2.6 Managing Namespace Permissions

Perhaps the most important task for OTM administrators is managing user group permissions for the repository's namespaces. Before discussing the mechanics of configuring those permissions, it is important to understand the types of permissions that can be granted (or denied) and the hierarchical nature of those permissions.

Table 1 lists each possible permission and the implications of granting or denying that permission in a particular namespace.

Permission	Grant Implications	Deny Implications
Read-Final	Allows read access to managed artifacts in Final status. Items in Draft status are not visible.	Denies read (and write) access to all managed artifacts in a namespace.
Read-Draft	Allows read access to managed artifacts in Draft status. Implies Read-Final access.	Denies read access to artifacts in Draft status (implies denial of write access). Users still have Read-Final access if they were otherwise granted that permission in a higher-level namespace.
Write	Allows write access to managed artifacts in Draft status. Implies Read-Draft and Read-Final access.	Denies write access to managed artifacts. Users still have Read-Draft (and/or Read-Final) access if they were otherwise granted that permission in a higher-level namespace.

Table 1: Namespace Permissions Overview

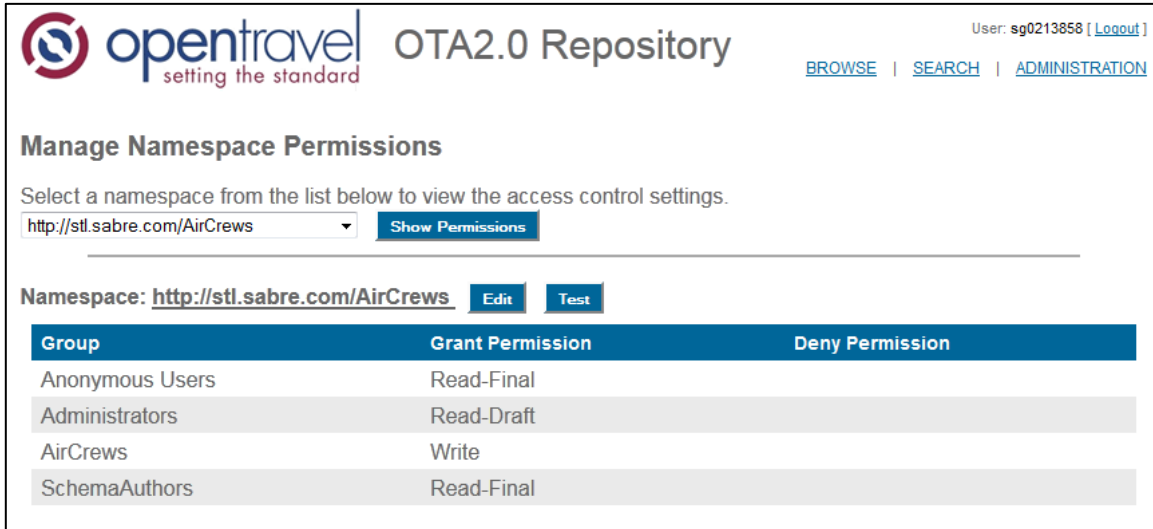
Permissions can be granted (or denied) to user groups at the individual namespace level, allowing administrators to define very fine-grained and sophisticated access control schemes if needed. Likewise, if a very simple scheme is all that is required, administrators can define a very simple set of global permissions that will apply to the entire repository.

It is also important to note that permissions are hierarchical in nature, in that they are inherited by lower-level paths in the namespace hierarchy. Under this scheme, permission grants in lower-level namespaces are additive in nature, while permission denials are considered subtractive.

In the example below, the ModelAuthors group has been granted write permission to the entire repository. This permission has been overridden in the /pricing namespace by denying the Read-Draft permission to the members of that group (note that ModelAuthors would still have Read-Final permissions in the /pricing namespace).

- + Global Repository Permissions
 - + http://www.mycompany.com
 - + /pricing
- ← Grant Write to ModelAuthors group
 - ← Deny Read-Draft from ModelAuthors group

To display the permissions that have been granted and/or denied in a particular namespace, click the 'Manage Namespace Permissions' link on the main administration page. From there, select a managed namespace from the drop-down list and click the 'Show Permissions' button.



User: sg0213858 [Logout]

[BROWSE](#) | [SEARCH](#) | [ADMINISTRATION](#)

Manage Namespace Permissions

Select a namespace from the list below to view the access control settings.

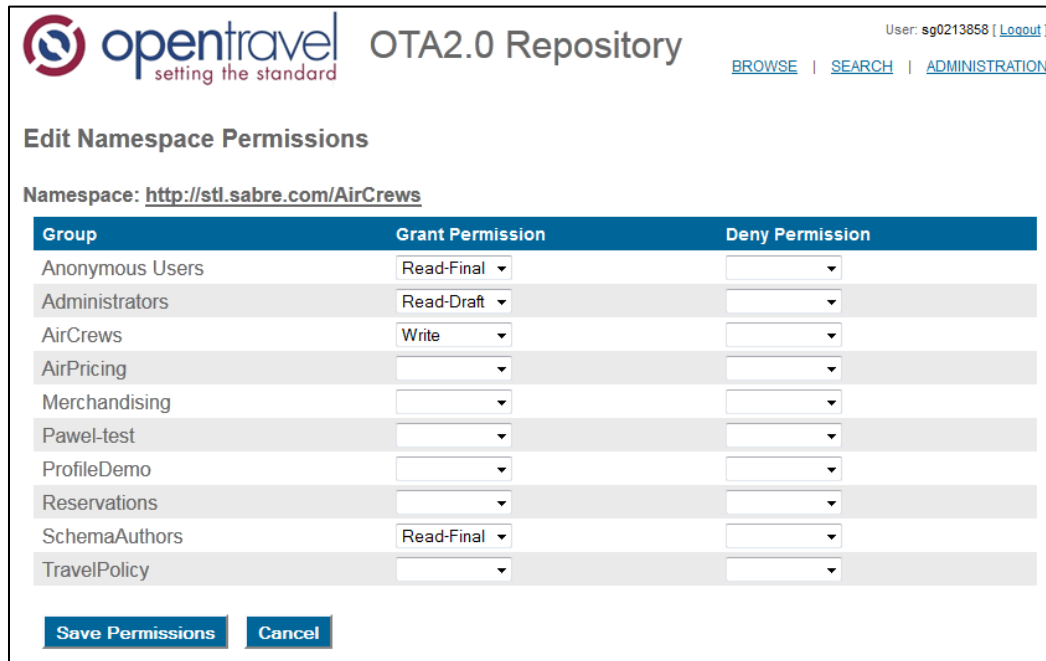
http://stl.sabre.com/AirCrews [Show Permissions](#)

Namespace: <http://stl.sabre.com/AirCrews> [Edit](#) [Test](#)

Group	Grant Permission	Deny Permission
Anonymous Users	Read-Final	
Administrators	Read-Draft	
AirCrews	Write	
SchemaAuthors	Read-Final	

Figure 14: Manage/View Namespace Permissions

To edit the permissions in the selected namespace, click the 'Edit' button next to the namespace URI in the next section of the page. Then select the appropriate permission for each group in the selected namespace. Note that the permissions that are entered on this page are explicit grants and denials for the namespace. If a groups permissions are to be inherited from a higher-level namespace, that permission drop-down should remain blank (unassigned) for the selected namespace.



User: sg0213858 [Logout]

[BROWSE](#) | [SEARCH](#) | [ADMINISTRATION](#)

Edit Namespace Permissions

Namespace: <http://stl.sabre.com/AirCrews>

Group	Grant Permission	Deny Permission
Anonymous Users	Read-Final	
Administrators	Read-Draft	
AirCrews	Write	
AirPricing		
Merchandising		
Pawel-test		
ProfileDemo		
Reservations		
SchemaAuthors	Read-Final	
TravelPolicy		

[Save Permissions](#) [Cancel](#)

Figure 15: Edit Permissions for a Namespace

2.7 Refreshing the Free-Text Search Index

Updating the free-text search index is a relatively simple task. Just click on the ‘Refresh Free-Text Search Index’ link on the main administration page. This should only be necessary if the any repository artifacts have been manually updated or the content of the repository has been restored from a backup.

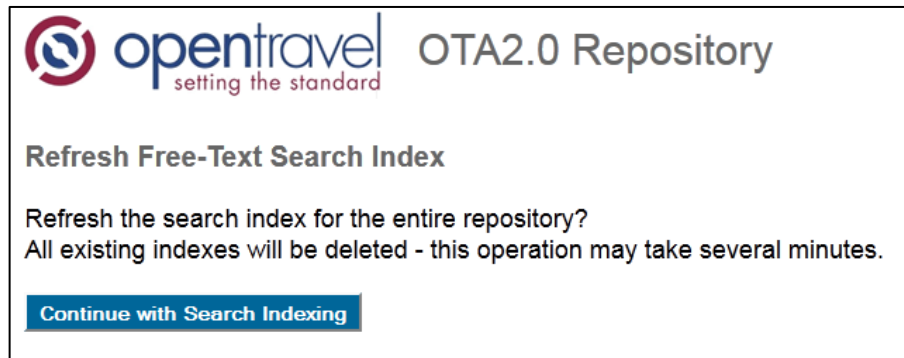


Figure 16: Refresh the Repository’s Free-Text Search Index

2.8 Managing OTM Artifacts

For managed OTM libraries, a number of functions are available to administrators from the Item Details page. This page is accessible via browsing or free-text searches in the web console application. The links that are available to administrator users are not visible to non-administrators.

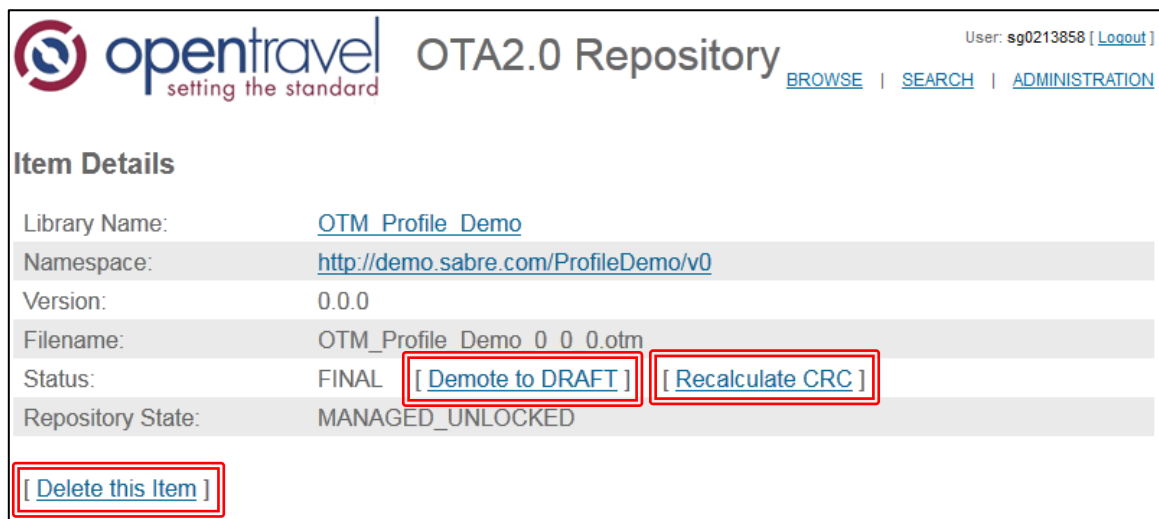


Figure 17: Item Details Page with Links to Administrator Functions

The following functions are available to administrators from this page:

- **Finalize Item** – Items in Draft status may be promoted to Final. From the web console, this operation is only available to administrators. For user's of the OTM-DE GUI application, this operation can be performed by any user with write access to the item.
- **Demote to Draft** – After an item has been promoted to Final status it cannot be changed (even by users with write or administrative access). If the promotion was done by mistake or needs to be undone for any reason, OTM administrators can demote artifacts back to Draft status.
- **Recalculate CRC (Final items only)** – When an item promoted to Final status, a CRC value is inserted into the library to ensure that the file cannot be edited (even hand edits are not allowed). If the CRC value becomes corrupted for any reason, administrators can force a recalculation of the value.
- **Unlock Item (coming soon)** – Normally, only the user who obtained a lock on a managed artifact can release that lock. The repository web console, however, does allow administrators to release locks that have been obtained by any user. WARNING: Any work-in-process changes that have been made by the locking user will be lost if the item is unlocked from the web console.
- **Delete Item** – Only administrators have the ability to permanently delete items from the repository. This action is not recoverable unless SVN file management has been configured for the repository.

3 Appendix: Repository File System Layout

The layout of the OTM repository file system is as follows:

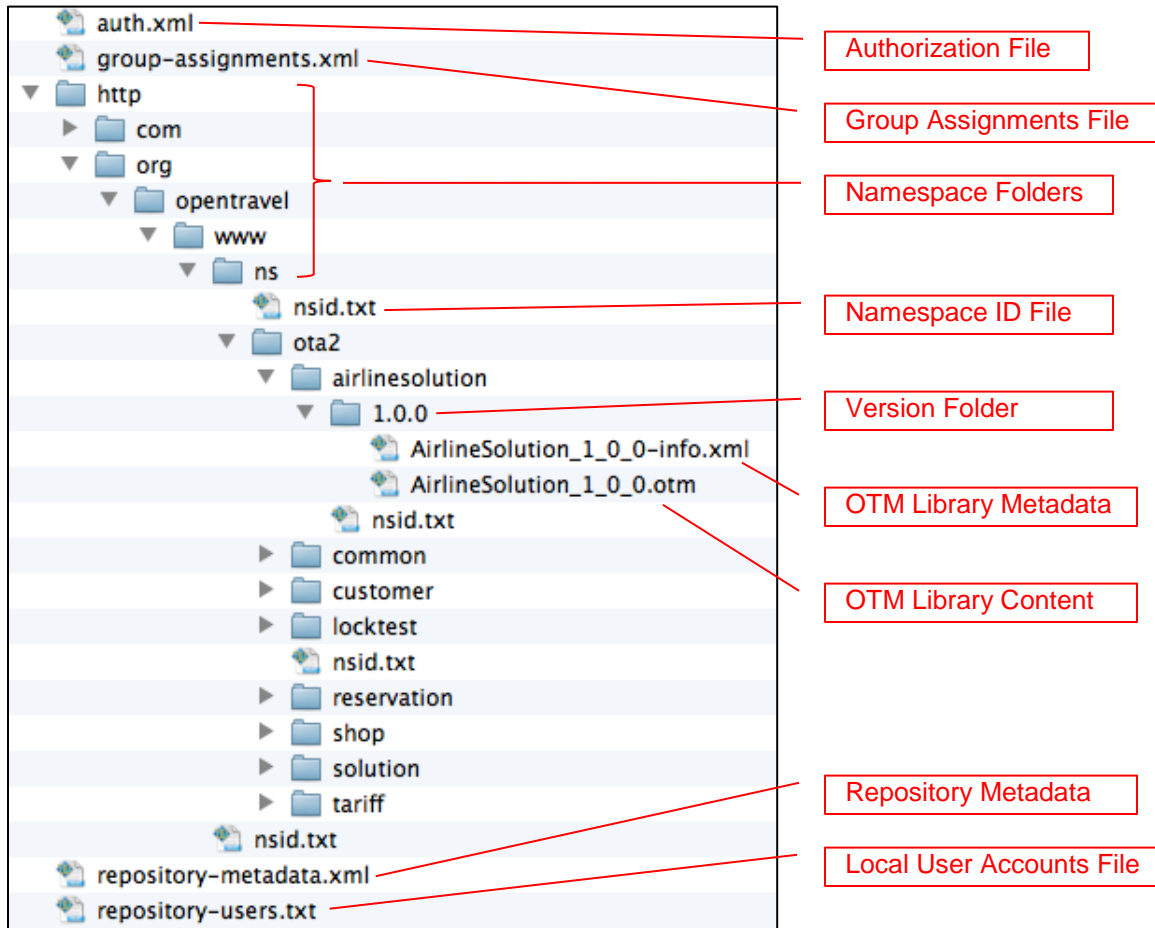


Figure 18: OTM Repository File and Folder Structure

Repository Metadata [repository-metadata.xml]

The repository metadata file contains the identity of the repository and the list of managed root namespaces. The only aspect of this file that cannot be edited from the administration console is the unique ID for the repository.

```
<otp:RepositoryInfo xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:otp="http://www.OpenTravel.org/ns/OTA2/RepositoryInfo_v01_00">
  <otp:ID>test-repository</otp:ID>
  <otp:DisplayName>OTA2.0 Local Development Repository</otp:DisplayName>
  <otp:RootNamespace>http://www.OpenTravel.org</otp:RootNamespace>
  <otp:RootNamespace>http://services.sabre.com</otp:RootNamespace>
  <otp:RemoteRepositories/>
</otp:RepositoryInfo>
```

Local User Accounts File [repository-users.xml]

The local user accounts file is only necessary if local user management has been configured for the repository. The format of the file is very simple; each line contains the user's ID and their encrypted password, separated by a colon.

```
admin:A0xvAVbKZKfyElheEsu2kX3gfJtB89kZnPHC3/muKZ4r=
user1:uRdzy7tQDgg/HV3igEtypORNUJB9YrXlzQm7NLg+yjV4x
user2:l+yyEuTRWrfD+lWVNI9CAxlvfVVVXYmsiallL8Aa6BQ==
```

Group Assignments File [group-assignments.xml]

The group assignments file defines all of the groups for the repository, as well as the users who are assigned to each group.

```
<sec:GroupAssignments xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:sec="http://www.OpenTravel.org/ns/OTA2/Security_v01_00">
  <sec:Group name="Administrators">
    <sec:Member>admin</sec:Member>
    <sec:Member>user1</sec:Member>
  </sec:Group>
  <sec:Group name="SchemaAuthors">
    <sec:Member>admin</sec:Member>
    <sec:Member>user1</sec:Member>
  </sec:Group>
  <sec:Group name="SchemaObservers">
    <sec:Member>user1</sec:Member>
  </sec:Group>
</sec:GroupAssignments>
```

Namespace Folders

The folder structure of the repository content is structured according to the namespaces of the artifacts contained within the repository. The basic folder scheme is as follows:

- **Top Level:** Scheme of the namespace URI (e.g. "http" or "https")

- **Authority Level(s)**: Reverse order of the URI authority (e.g. “/org/opentravel” or “/com/sabre/services”)
 - **Namespace Extension Level(s)**: Path structure of the URI (e.g. “ns/ota2/airlinesolution”)

All namespace folders are converted to lowercase letters on the local file system.

Namespace ID File [nsid.txt]

Since all namespace folders in the repository are converted to lowercase, a namespace ID file is placed in every managed directory that specifies the original case-sensitive name for the namespace component. For example, the contents of the namespace ID file for the /airlinesolution folder might be as follows:

```
AirlineSolution
```

Authorization File [auth.xml]

Authorization files contain the granted and/or denied permissions for each namespace in the repository. Since repository permissions are hierarchical in nature, each namespace folder can potentially contain an authorization file. The authorization file in the root folder of the repository's content contains the global permissions that are inherited by all managed namespaces within the repository.

```
<sec:NamespaceAuthorizations xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:sec="http://www.OpenTravel.org/ns/OTA2/Security_v01_00">
  <sec:Grant permission="ReadDraft">
    <sec:Principal>anonymous</sec:Principal>
  </sec:Grant>
  <sec:Grant permission="Write">
    <sec:Principal>SchemaAuthors</sec:Principal>
  </sec:Grant>
</sec:NamespaceAuthorizations>
```

Version Folder

Within each managed namespace, multiple versions of an artifact may exist. To account for this, all OTM artifacts are maintained within a version folder that is named according to each item's version identifier (e.g. "1.5.2"). All version folder names include the major, minor, and patch version numbers of the version identifier.

OTM Library Metadata

A metadata file that contains basic information about the file accompanies each OTM managed artifact that is published to the repository. The name of this file is always the same as the managed artifact itself, with a “-info.xml” suffix in place of the original file suffix.

```

<otp:LibraryInfo xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:otp="http://www.OpenTravel.org/ns/OTA2/RepositoryInfo_v01_00">
  <otp:Namespace>http://www.OpenTravel.org/ns/OTA2/AirlineSolution_v01_00</otp:Na
namespace>
  <otp:BaseNamespace>http://www.OpenTravel.org/ns/OTA2/AirlineSolution</otp:BaseN
amespace>
  <otp:Filename>AirlineSolution_1_0_0.otm</otp:Filename>
  <otp:LibraryName>AirlineSolution</otp:LibraryName>
  <otp:Version>1.0.0</otp:Version>
  <otp:VersionScheme>OTA2</otp:VersionScheme>
  <otp:Status>Draft</otp:Status>
  <otp:State>ManagedUnlocked</otp:State>
  <otp:LastUpdated>2013-04-26T07:21:58.221-05:00</otp:LastUpdated>
  <otp:OwningRepository>test-repository</otp:OwningRepository>
</otp:LibraryInfo>
  
```

OTM Library Content

The raw content of each managed OTM library artifact is exactly as it was saved and published using the OTM-DE GUI application.