

Oracle Endeca Commerce

Workbench Administrator's Guide

Version 3.1.1 • December 2012



Contents

Preface.....	7
About this guide.....	7
Who should use this guide.....	7
Conventions used in this guide.....	8
Contacting Oracle Support.....	8
Chapter 1: Introduction to Oracle Endeca Workbench.....	9
Logging in to Workbench as an administrator.....	9
Changing the administrator's password.....	9
About configuring Workbench resource caching in a Web browser.....	10
Chapter 2: Managing Users in Oracle Endeca Workbench.....	11
About users and permissions in Workbench.....	11
About the predefined admin user.....	12
Best practices.....	12
Adding administrators to Oracle Endeca Workbench.....	12
Adding groups to Oracle Endeca Workbench.....	13
Adding business users to Oracle Endeca Workbench.....	14
Modifying users and groups.....	16
Changing a group's membership.....	16
Deleting users from Oracle Endeca Workbench.....	17
Configuring the user inactivity logout.....	17
Chapter 3: Integrating LDAP with Oracle Endeca Workbench.....	19
About LDAP integration with Workbench.....	19
User authentication in Workbench with LDAP enabled.....	19
Permissions for LDAP users and groups.....	20
About granting administrator privileges in Workbench to LDAP users and groups.....	20
Enabling LDAP authentication in Workbench.....	20
About disabling LDAP authentication in Workbench.....	21
About the LDAP login configuration file.....	21
Templates used in the LDAP login profile.....	22
About configuration parameters for the LDAP login profile.....	22
Configuration parameters for the LDAP login profile.....	23
Configuration parameters for identity information stored in LDAP.....	25
LDAP path parameters.....	26
About specifying multiple values for parameters in the LDAP login profile.....	26
Specifying the location of the LDAP login configuration file.....	27
Specifying the location of the LDAP login configuration file using Windows Services.....	28
Troubleshooting user authentication in Workbench with LDAP enabled.....	28
Chapter 4: Configuring Communication with Other Endeca Components.....	31
About Workbench interaction with the Endeca Configuration Repository.....	31
Specifying Workbench authentication credentials for the Endeca Configuration Repository.....	31
Configuring the shared key for Workbench extensions hosted in the Endeca Configuration Repository.....	33
About Workbench interaction with the MDEX Engine.....	34
About specifying the MDEX Engine that Workbench queries.....	34
About specifying which Dgraphs to update with configuration changes.....	34
About accessing files on remote servers.....	35
Chapter 5: Configuring SSL for Oracle Endeca Workbench.....	37
About configuring SSL in Workbench.....	37
Enabling the SSL version of Oracle Endeca Workbench.....	37
Modifying the server.xml for the Endeca Tools Service.....	38

Chapter 6: Configuring Workbench System Logs.....	41
About Workbench system logs.....	41
Configuring the Oracle Endeca Workbench logs.....	41
Chapter 7: Customizing Oracle Endeca Workbench.....	43
The navigation menu and launch page.....	43
Navigation menu nodes.....	43
Node titles for multiple locales.....	44
Predefined menu nodes in Oracle Endeca Workbench.....	44
About navigation menu leaf items.....	45
Predefined menuitem elements.....	46
Updating the Oracle Endeca Workbench menu and launch page.....	47
Workbench extensions.....	47
About configuring extensions in Oracle Endeca Workbench.....	48
Extension element attributes.....	48
Enabling extensions in Oracle Endeca Workbench.....	49
URL tokens and Workbench extensions.....	50
URL token reference.....	50
Token-based authentication for Workbench extensions.....	51
Troubleshooting Workbench extensions.....	53

Copyright and disclaimer

Copyright © 2003, 2012, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Preface

The Oracle Endeca Commerce solution enables your company to deliver a personalized, consistent customer buying experience across all channels — online, in-store, mobile, or social. Whenever and wherever customers engage with your business, the Oracle Endeca Commerce solution delivers, analyzes, and targets just the right content to just the right customer to encourage clicks and drive business results.

Oracle Endeca Commerce is the most effective way for your customers to dynamically explore your storefront and find relevant and desired items quickly. An industry-leading faceted search and Guided Navigation solution, Oracle Endeca Commerce enables businesses to help guide and influence customers in each step of their search experience. At the core of Oracle Endeca Commerce is the MDEX Engine™, a hybrid search-analytical database specifically designed for high-performance exploration and discovery. The Endeca Content Acquisition System provides a set of extensible mechanisms to bring both structured data and unstructured content into the MDEX Engine from a variety of source systems. Endeca Assembler dynamically assembles content from any resource and seamlessly combines it with results from the MDEX Engine.

Oracle Endeca Experience Manager is a single, flexible solution that enables you to create, deliver, and manage content-rich, cross-channel customer experiences. It also enables non-technical business users to deliver targeted, user-centric online experiences in a scalable way — creating always-relevant customer interactions that increase conversion rates and accelerate cross-channel sales. Non-technical users can control how, where, when, and what type of content is presented in response to any search, category selection, or facet refinement.

These components — along with additional modules for SEO, Social, and Mobile channel support — make up the core of Oracle Endeca Experience Manager, a customer experience management platform focused on delivering the most relevant, targeted, and optimized experience for every customer, at every step, across all customer touch points.

About this guide

This guide describes the tasks involved in configuring and administering Oracle Endeca Workbench and some features of your Endeca application. Oracle Endeca Workbench contains configuration and administrative functionality for system administrators as well as business logic functionality for business users.

Who should use this guide

This guide is intended for system administrators and others who are managing the day-to-day operations of an Oracle Endeca Workbench implementation.

Oracle Endeca Workbench is a Web-based tool intended for business users and system administrators.

Within Workbench, system administrators can perform any of the following tasks:

- Perform system operations such as running baseline updates or starting and stopping the MDEX Engine or Log Server.

- Monitor the status of system components such as Dgidx, the MDEX Engine, Log Server, and Report Generator.
- Configure SSL settings
- Configure report generation
- Provision the hosts available to an Endeca implementation.
- Provision the applications available to an Endeca implementation.
- Provision scripts, such as a baseline update script to an Endeca implementation.

Workbench and Developer Studio rely on the Endeca Application Controller (EAC) to control and communicate with other hosts and components in an Endeca implementation.

Conventions used in this guide

This guide uses the following typographical conventions:

Code examples, inline references to code elements, file names, and user input are set in `monospace` font. In the case of long lines of code, or when inline monospace text occurs at the end of a line, the following symbol is used to show that the content continues on to the next line: ↵

When copying and pasting such examples, ensure that any occurrences of the symbol and the corresponding line break are deleted and any remaining space is closed up.

Contacting Oracle Support

Oracle Support provides registered users with important information regarding Oracle Endeca software, implementation questions, product and solution help, as well as overall news and updates.

You can contact Oracle Support through Oracle's Support portal, My Oracle Support at <https://support.oracle.com>.

Introduction to Oracle Endeca Workbench

This section introduces Oracle Endeca Workbench and describes how to access it via a Web browser.

Logging in to Workbench as an administrator

Upon installation, Workbench has a predefined administrator user with full administration privileges.

To log in to Workbench as an administrator:

1. In a Web browser, navigate to the Workbench login page.
By default, this is `http://localhost:8006`.
2. Specify a username and password.
The predefined administrator account has a username of `admin` and a password of `admin`.
3. Click **Log In**.

After your initial login, you can change the password of the predefined admin user or create and manage users and administrators.

Changing the administrator's password

Oracle recommends that you change the predefined administrator's password.

Follow these steps to change the password for the predefined admin user. You cannot use this procedure to change the password of any Oracle Endeca Workbench administrators that are in the LDAP directory.

1. In the upper right corner of Oracle Endeca Workbench, click the down arrow next to **admin** and then click **Change Password**.
2. Enter the current admin password in the **Old Password** field.
If this is the first time that you are changing the admin password, type `admin`.
3. Enter your new password in the **New Password** and **Confirm Password** fields.
4. Click **Save**.

About configuring Workbench resource caching in a Web browser

If users are connecting to Workbench over a high-latency network, setting certain cache settings in the Web browser may improve performance.

Workbench pages reference a number of assets such as images, CSS files, and JavaScript files. These assets are relatively static and are typically cached by the browser. They are served with HTTP headers that instruct the browser not to check for new versions of these files for a period of six hours. This results in better page load performance for users who connect to Workbench over a high-latency network.

Oracle recommends setting your browser to check for new versions of pages automatically, and allocating 256MB or more of disk space to temporary files.

Chapter 2

Managing Users in Oracle Endeca Workbench

This section describes the Workbench users and permissions model, and how to manage users and groups within Workbench.

About users and permissions in Workbench

Workbench users and permissions are defined by a Workbench administrator.

Workbench users log in with standard user name and password authentication, and permissions dictate which Workbench tools and content within an application are available to them. Workbench administrators create accounts for users that define this authentication and permission information. Administrators can also create groups and then add users to the groups. Creating groups is the preferred method of managing user permissions. For example, you can place all users that need access to a specific content collection into a group. By granting access to the group rather than to each user, you avoid the need to assign permissions to each user one-by-one. You can also make groups members of other groups which further aids in assigning permissions.

Once groups and users are added to Workbench, their names and passwords are associated with all applications across Workbench. Permissions, on the other hand, are associated with single applications, and must be specified for each application in Workbench.

Within an application, administrators provide permissions at the tool level or at a more granular content level. For example, you might provide a group with permissions to access the Experience Manager tool. This high-level access provides group members with write access to all the content within the Experience Manager. The administrator can also limit access to only a single content collection within the Experience Manager.

User management and LDAP

LDAP, Lightweight Directory Access Protocol, is a centralized directory used by programs to look up user information. Using LDAP, one password for a user can be shared between many services. If you have Workbench configured to use LDAP for user authentication, an administrator can create a member profile where the password and identity information is stored and managed in an LDAP directory. LDAP integration also allows you to assign permissions across an entire LDAP group rather than configuring each user individually. For more information about configuring Workbench with LDAP, see *Integrating LDAP with Oracle Endeca Workbench*.

Special characters

User and group names cannot contain the following characters: / \ : [] | * ? " < >.

About the predefined admin user

Workbench has a predefined administrator with full administration privileges.

An administrator is granted all permissions in the system. The predefined Workbench administrator uses the username `admin` with default password `admin`. After logging in to Workbench you can modify the password, but not the user name.

The `admin` user can create additional users and administrators in Workbench (only an administrator can create other administrators). An administrator can also delete other administrators, but not the predefined administrator. If you have LDAP authentication enabled, see the section *About granting administrator privileges in Workbench to LDAP users and groups* for additional information on creating Workbench users.

Best practices

Oracle recommends the following best practices for managing users in the Workbench.

- Consider adding all users to groups to make managing permissions simpler.
- Do not share the predefined admin user account. This makes it difficult to track who has made changes to Workbench. Create an account for each user or group that administers Workbench.
- Do not let business users share accounts. Again, this makes it difficult to track changes in Workbench.
- Create administrators in addition to the predefined admin. If one administrator loses a password, another administrator needs to reset it.
- If you use LDAP, consider creating an LDAP group of administrators to add to the Administrators group in Workbench.

Adding administrators to Oracle Endeca Workbench

Administrators can configure other administrators on the **User Management** page.

You add administrators to the Oracle Endeca Workbench by adding a users or groups to the administrators group.

1. From Administrative Tools, click **User Management**.
2. Click the **Users** or **Group** tab, and then click **Add User** or **Add Group**.
3. Select a **Source**. If your site uses LDAP, then **LDAP** is already selected.
4. Enter the name of the user or group that you want to add in the **User ID** or **Group ID** field.
5. If your **Source** is LDAP, click **Validate** to determine if you have entered a valid LDAP user name.



Note: **Validate** does not display if your **Source** is Workbench.

If you entered a valid LDAP user or group name, then Workbench retrieves available information and populates the name and email fields. This information as well as the user password is not editable.


6. If your **Source** is Workbench, complete the following fields.

For a single user:

- First Name

- Last Name
- Email
- Password
- Confirm Password

For a group:

- Name
 - Email
7. Select **Administrators** from the **Select a group** drop-down, and click the **Include in Group** button .
- You do not need to give this new administrator any additional permission since administrators have all available permissions already.
8. Click **OK**.

Adding groups to Oracle Endeca Workbench

Administrators and administrative users with permissions can configure groups on the **User Management** page.

You can add a group in one of these ways:

- Create a group and assign permissions to the group.
- Add a group that is stored in LDAP and assign permissions to the group.

The LDAP options are only available if you have configured Oracle Endeca Workbench to use LDAP for user authentication. For more information about using Oracle Endeca Workbench with LDAP, see the *Integrating LDAP with Oracle Endeca Workbench*.


To add a group to Oracle Endeca Workbench:

1. From Administrative Tools, click **User Management**.
2. Click the **Groups** tab, and then click **Add Group**.
The **Create Group** dialog displays.
3. Select a **Source**. If your site uses LDAP, then **LDAP** is already selected; if not, there is no **Source** to select.
4. Enter a name in the **Group ID** field.
5. If your **Source** is LDAP, click **Validate** to determine if you have entered a valid LDAP group name.




Note: **Validate** does not display if your **Source** is Workbench.

If you entered a valid LDAP group name, then Workbench retrieves available group information and populates the name and email fields. This information is not editable.

6. If your **Source** is Workbench, complete the following optional fields:
 - Name
 - Email
7. If you want to make this group a member of another group, select a group in which to add the group from the **Select a group** drop-down, and click the **Include in Group** button . Repeat this for each group in which you want to add the group as a subgroup.
The group is granted any permissions of the group in which it is a subgroup.

Permissions are cumulative. For example, if the new group belongs to two groups: group A and group B, and group A has permission to use a tool and group B does not, then the new group has access to the tool due to his or her membership in group A.

8. To populate this group with users and groups, click the **Membership** tab.
 - a) Select a user or group from the **Include a user or group...** drop-down list, and then click the **Include in Group** button  .
The user or group displays in the list.
 - b) Repeat the first step until you have added all the members that you want to the group.
9. If you want to give the group additional permission to use tools or to access content that group membership does not provide, then follow these steps. You cannot use this procedure to remove permissions that a group is granted by membership in another group.
 - a) Click the **Permissions** tab.
 - b) Select the application in which the group will be working.
Any tools that the group's parent group grants permissions for are already selected.
 - c) Select the additional tools to which you want to give the group access.
Giving the group access gives it full access: the group members can read, write and edit.
 - d) If you selected Experience Manager in the previous step, or if it was already selected, you can update the content that the group can access. Expand **Experience Manager**.
Any content that the group's parent group grants permissions for are already selected. The group's members have read access to any unselected content.
 - e) Select or deselect the pages, folders, and content collections to which you want to update group access.
Giving the group access gives it full access: the group members can read, write, and edit. Content permissions are inherited, so a content collection has the same permissions as the folder to which it belongs.



Note: Deselecting pages, folders or content collections removes write access. The group still has read access.

10. Click **Save** .

The new group profile displays on the **Groups** tab of the **User Management** page.

Adding business users to Oracle Endeca Workbench

Administrators and administrative users with permissions can configure users on the **User Management** page.

You can add a user in one of these ways:

- Add a user manually in Oracle Endeca Workbench and assign permissions.
- Add a user that is stored in LDAP and assign permissions.

If a user is a member of an LDAP that has been added to Workbench, the user is automatically added the first time that he or she logs in to Workbench.

The LDAP options are only available if you have configured Oracle Endeca Workbench to use LDAP for user authentication. For more information about using Oracle Endeca Workbench with LDAP, see the *Integrating LDAP with Oracle Endeca Workbench*.


To add a user to Oracle Endeca Workbench:

1. From Administrative Tools, click **User Management**.
2. Click the **Users** tab, and then click **Add User**.
The **Create User** dialog displays.
3. Select a **Source**. If your site uses LDAP, then **LDAP** is already selected.
4. Enter a name in the **User ID** field.
5. If your **Source** is LDAP, click **Validate** to determine if you have entered a valid LDAP user name.



Note: **Validate** does not display if your **Source** is Workbench.

If you entered a valid LDAP user name, then Workbench retrieves available user information and populates the name and email fields. This information as well as the user password is not editable.

6. If your **Source** is Workbench, complete the following fields:
 - First Name
 - Last Name
 - Email
 - Password
 - Confirm Password
7. Select a group in which to add the user from the **Select a group** drop-down, and click the **Include in Group** button . Repeat this for each group in which you want to add the user.



Note: You cannot add an LDAP user to an LDAP group.

The user inherits the permissions of a group in which it is a member.

Permissions are cumulative. A user who is a member of groups defined in Workbench is assigned the broadest permission associated with any of the groups to which that user belongs. For example, if the user belongs to two groups: group A has permission to use Experience Manager and group B does not, then the user has access to Experience Manager due to his or her membership in group A.

8. If you want to give the user additional permission to use tools or to access content in your application that group membership does not provide, or if the user is not a member of any group, then follow these steps. You cannot use this procedure to remove permissions that a user is granted by membership in a group as permissions are additive. These permissions display in grey to indicate they cannot be removed.
 - a) Click the **Permissions** tab.
 - b) Select the application in which the user will be working.
Any tools for which the user's group membership grants permissions are already selected.
 - c) Select the additional tools to which you want to give the user access.
 - d) If you selected **Experience Manager** in the previous step, or if it was already selected, you can update the content that the user can access. Expand **Experience Manager**.
 - e) Select or deselect the pages, folders, and content collections to which you want to update user access.

Giving users access gives them full access: the user can read, write, and edit. Content permissions are inherited, so a content collection has the same permissions as the folder to which it belongs. You can also add permissions explicitly if the folder has no permissions.



Note: Deselecting pages, folders or content collections removes write access. The user still has read access.

9. Click **Save** .

The new profile displays on the **Users** tab of the **User Management** page.


Modifying users and groups

Administrators and administrative users with permission can modify aspects of a user or group including password, identity information, and permissions. You cannot change password and identity information for LDAP users and groups.

You cannot change the user or group name. To change a user or group name, create a new member with the new name and the same permissions, then delete the existing user or group.

See *Changing a group's membership* to add or remove members from a group.

To modify a user or group:

1. From the **User** tab or **Group** tab on the **User Management** page, click the user or group name of the user or group whose profile you want to modify.
2. Modify any first or last name, group name, email, and password information that is required. You cannot update this information for LDAP users and groups.
3. Optionally, select a group to which you want to add the user or group and then click the **Include in Group** button  .
4. Click the **Permissions** tab and update user or group access to tools and content. You cannot remove permissions if the user or group inherits the permission from a parent group. You can, however, add permissions.
5. Click **Save**.

If you want to add or remove members from a group, see *Changing a group's membership*.


Changing a group's membership

You can add and remove members from a group with the following exception: you cannot add and remove LDAP members from an LDAP group.

When you add members to an existing group, the members are granted all the permissions that the group has. If a member belongs to multiple groups that have different permissions, then the member has the broadest permissions. If you remove members from a group, be sure that the members have all the necessary permissions that they need either as an individual user or as a member of another group.

To add and remove members:

1. From Administrative Tools, click **User Management**.
2. Click the **Groups** tab, and then click the group in which you want to add or remove members.
3. Click the **Membership** tab.

4. From the **Include a user or group...** drop-down list, select the user or group that you want to add to the current group, and click the **Add to group** button .
5. To remove a member of group from this group:
 - a) Enter the name in the **Find** field or locate the name in the list that displays in the dialog.
 - b) Click the **X** icon in the Remove column for each user or group that you want to remove.
6. Click **Save**.

The changes take affect the next time a user or group member attempts to use a tool.

Deleting users from Oracle Endeca Workbench

An administrator can delete users and groups from Oracle Endeca Workbench.

There are restrictions on deleting users:

- You cannot delete yourself.
- You cannot delete the predefined `admin` user.
- You cannot delete the Administrators group.

If your site has deployed LDAP, deleting users does not delete them from the LDAP system.

To delete a user from Workbench:

1. On the **User Management** page, click the **Delete** icon for the user or group that you want to remove.
2. Click **Delete** in the confirmation dialog.

Configuring the user inactivity logout

You can configure how much time elapses before an inactive user is logged out of Oracle Endeca Workbench in the `webstudio.properties` file.

Follow these steps to set how much time that Workbench users must be inactive before they are automatically logged out. You can also specify how much time elapses before a timeout warning message appears to inactive users.

1. Stop the Endeca Tools Service.
2. Navigate to `%ENDECA_TOOLS_CONF%\conf` (on Windows) or `$ENDECA_TOOLS_CONF/conf` (on UNIX).
3. Open the `webstudio.properties` file, and locate the `com.endeca.webstudio.timeout.warning` property.

```
# The warning for impending auto-logout
com.endeca.webstudio.timeout.warning=3300
```

4. Change the value to the number of seconds of inactivity that you want to elapse before an impending automatic logout warning appears to an inactive user.
5. Locate the `com.endeca.webstudio.timeout.logout` property.

```
# The time where a user will be automatically logged out due to inactiv-
ity
com.endeca.webstudio.timeout.logout=3600
```

6. Change the value to the number of seconds of inactivity that you want to elapse before an inactive user is logged out of Workbench.
7. Save and close the file.
8. Start the Endeca Tools Service.

Integrating LDAP with Oracle Endeca Workbench

This section describes how to configure Workbench to use LDAP for user authentication.

About LDAP integration with Workbench

LDAP integration allows you to share user identity information and passwords defined in LDAP with Workbench. You can also assign permissions for LDAP users or across an entire LDAP group rather than configuring Workbench users individually.

By configuring Workbench to use LDAP for user authentication, you enable administrators to create Workbench user profiles that are associated with users in an LDAP directory.

Workbench does not write data to the LDAP directory. Passwords and identity information, such as names and e-mail addresses, are maintained in the LDAP directory. Permissions assigned to an LDAP user or group profile in Workbench are stored in the Endeca Configuration Repository.

LDAP user and group profiles can be used in combination with non-LDAP Workbench user and groups that are manually configured by an administrator. Users can authenticate using either method on the same instance of Workbench.

Optionally, you can enable SSL for communication between Workbench and your LDAP server.

Supported versions of LDAP

Workbench supports integration with all LDAP servers that comply with LDAP version 3.

User authentication in Workbench with LDAP enabled

Once you have integrated LDAP with your Workbench installation, you can authenticate users either through LDAP or by configuring them manually in Workbench.

Workbench does the following when a user attempts to log in:

1. Workbench checks whether the user name matches the name of any manually configured Workbench user in the current application. If such a user exists, Workbench attempts to authenticate the user against the password stored for that user.
2. If no manually configured user exists for the name entered, Workbench attempts to authenticate the user against the LDAP directory.
3. If the user is configured for LDAP authentication in Workbench, any associated permissions are applied.

4. If the individual user is not configured for LDAP authentication in Workbench, Workbench checks the LDAP directory for any groups of which the user is a member.

If a profile exists in Workbench for any of these groups, the user is automatically made a member of each matching group. They inherit the permission of these groups. For more information about inheritance of LDAP group permissions, see “Permissions for LDAP users and groups.”

Permissions for LDAP users and groups

The User Management tool in Workbench allows you to assign permissions to an LDAP user or group.

A user that exists in the LDAP directory but is not associated with a Workbench user profile, either individually or as a member of an LDAP group, cannot log in to Workbench.

A user who authenticates via LDAP is assigned the union of all permissions associated with all groups of which that user is a member. A user who is a member of multiple LDAP groups defined in Workbench is assigned the broadest permission associated with any of the LDAP groups to which that user belongs. Every time users log in to Workbench, their group membership is synchronized with LDAP so that their permissions are current with any group membership changes.

If you create an LDAP user profile in Workbench for an individual who is also a member of one or more LDAP groups defined in Workbench, that user is assigned any permissions that you specify in the User Management tool in addition to any permissions that the user inherits from membership in LDAP groups. If you specify permissions for an LDAP user who is also a member of an LDAP group, then the user is assigned either the permission specified in the User Management tool or the broadest permission associated with any of the user's LDAP groups, whichever is broader.

About granting administrator privileges in Workbench to LDAP users and groups

With LDAP enabled, you can create user profiles for both LDAP users and LDAP groups that grant administrator privileges in Workbench.



Note: For administrators, the same precedence rules apply when logging in to Workbench as for non-administrators, so that if a manually configured user profile exists in Workbench, a user will not be able to log in via LDAP with the same user name.

Administrators can delete other administrators, but they cannot delete the predefined admin user or the Administrators group. This restriction ensures that changing LDAP permissions or disabling LDAP authentication for Workbench does not disable all administrator logins.

Enabling LDAP authentication in Workbench

LDAP authentication in Workbench is disabled by default.

Because LDAP configuration is unique to each LDAP server and directory, enabling LDAP authentication in Workbench is a manual process.

To enable LDAP authentication in Workbench:

1. Stop the Endeca Tools Service.

2. Navigate to `%ENDECA_TOOLS_CONF%\conf` (on Windows) or `$ENDECA_TOOLS_CONF/conf` (on UNIX).
3. Open the `webstudio.properties` file, and locate the `com.endeca.webstudio.useLdap` property:

```
# LDAP Authentication
com.endeca.webstudio.useLdap=false
```

4. Change the value of the property to `true`:

```
com.endeca.webstudio.useLdap=true
```

5. Save and close the file.
6. Open the `Login.conf` file.

This file contains a sample configuration for LDAP authentication.



Note: By default, Workbench uses the authentication profile in this location. You can specify an alternate configuration file. For more information, see “Specifying the location of the LDAP login configuration file.”

7. Uncomment and modify the login profile according to your LDAP configuration.
For details about profile parameters, see “About configuring the LDAP login profile.”
8. Save and close the file.
9. Start the Endeca Tools Service.

About disabling LDAP authentication in Workbench

If you disable LDAP authentication in Workbench by setting the property `com.endeca.webstudio.useLdap=false` in the `webstudio.properties` file, the options to create a user profile for an LDAP user or an LDAP group do not display in Workbench.

All new user profiles you create must be manually configured in Workbench. Any users who were configured as LDAP users or as members of an LDAP group lose access to Workbench. Existing user profiles for LDAP users or LDAP groups remain in Workbench in an inactive state, and can be edited by an administrator.

About the LDAP login configuration file

Workbench uses the Java Authentication and Authorization Service (JAAS) to authenticate users against an LDAP directory.

Workbench stores LDAP login configuration information in the `%ENDECA_TOOLS_CONF%\conf\Login.conf` file. A sample profile is included in this location by default, but you should modify its parameters as needed for your LDAP configuration. You can also specify an alternate location for the configuration file.

If you want to configure JAAS authentication for other applications running in the Endeca Tools Service (for example, your own Endeca application or Workbench extensions), you can create additional profiles with unique names in the `Login.conf` file.

Templates used in the LDAP login profile

Workbench allows templates to be supplied for certain configuration parameters in the LDAP login profile.

These templates, indicated by `%{ }` escapes, allow values from the authentication operation (such as a user or group name entered in Workbench or specific values from the user or group objects in LDAP) to be substituted into the parameter value. Templates also allow you to extract information from the LDAP user or group object (such as the exact user or group name as specified in the LDAP directory) or identity information that is stored in LDAP. The `%{ }` escapes are expanded as follows:

Escape	Description
<code>%{#username}</code>	The name of the LDAP user as entered in the User Settings tool in Workbench, or the user name entered by a user at the Workbench login page.
<code>%{#groupname}</code>	The name of the LDAP group as entered in the User Settings tool in Workbench.
<code>%{#dn}</code>	The distinguished name of the user or group object in the LDAP directory.
<code>%{#dn:n}</code>	The value of the path field at index <i>n</i> in the distinguished name of the user or group object in LDAP. For example, if the value in the <code>%{#dn}</code> field is <code>cn=joe,ou=People,dc=foo,dc=com</code> , then the value "People" will be substituted for <code>%{#dn:1}</code> , while "joe" will be substituted for <code>%{#dn:0}</code> . Note that unlike the value of <code>%{#dn}</code> , which is the raw value returned from the LDAP server, the values returned by this template are not LDAP escaped.
<code>%{#fieldname}</code>	The value in the specified field of the user object (or group object when used in the <code>groupTemplate</code> or <code>findGroupTemplate</code> parameter) under consideration.

About configuration parameters for the LDAP login profile

You specify the values of configuration parameters for LDAP authentication as quoted strings.

If there are any quotation marks (") or backslashes (\) in the string, they must be escaped. For example, if you have the following string:

```
"A string with an "embedded quote" and a \backslash"
```

In the profile, it should be specified as follows:

```
"A string with an \"embedded quote\" and a \\backslash"
```

For most parameter values, single quotation marks (') do not need to be escaped and the values you specify for the parameters can include non-ASCII UTF-8 characters. For additional restrictions on the `userPath`, `groupPath`, and `findGroupPath` parameters, see "LDAP path parameters."

For a full list of the parameters that can be specified in the profile, see the section "Configuration parameters for the LDAP login profile."


Configuration parameters for the LDAP login profile

This section provides a reference of parameters that can be specified in the LDAP login profile.

The following is a full list of the parameters that can be specified in the profile:

Parameter	Description
serverInfo	A URL specifying the name and port of the LDAP server to be used for authentication. You can specify multiple LDAP servers. Note that the protocol portion of the URL (that is, <code>ldap://</code>) must be in all-lowercase.
userPath	<p>The query that is passed to the LDAP server to find an individual user. You can use the <code>%{#username}</code> template to insert the name entered in the User Settings tool or the name entered in the Workbench login page into the query. Be sure to set the appropriate <code>objectClass</code>.</p> <p>For example:</p> <pre>userPath="/ou=users,dc=example,dc=com??sub?-&(objectClass=person)(uid=%{#username})"</pre>
userTemplate	<p>A template that specifies how to produce the username from the user object returned by the <code>userPath</code> query.</p> <p>This template allows Workbench to automatically correct the case (capital or lowercase) of the username to match the name exactly as specified in the LDAP directory. The correction occurs when you add an LDAP user to Workbench. Therefore, the value returned by this template should match the name entered in the User Settings tool, except for possible differences in case.</p>
groupPath	The query that is passed to the LDAP server to find all the groups of which a user is a member. This query is executed when a user logs in to Workbench after looking up the user with the <code>userPath</code> query. Thus, you can use templates to insert any information from the user object that is returned by the previous query, such as the distinguished name of the user or any other LDAP attributes, into the <code>groupPath</code> query. You can specify multiple values for <code>groupPath</code> .
groupTemplate	A template that specifies how to produce individual group names from the set of groups returned by the <code>groupPath</code> query. The value returned by this template should match the name of the LDAP group as defined in the Workbench user profile. You can specify multiple values for <code>groupTemplate</code> .
findGroupPath	The query that is passed to the LDAP server to find a specific group. You can use the <code>%{#groupname}</code> template to insert the name of the group as entered in the User Settings tool into the query. Be sure to set the appropriate <code>objectClass</code> .

Parameter	Description
	<p>For example:</p> <pre>findGroupPath="/ou=groups,dc=example,dc=com- ??sub?(&(objectClass=group)(cn=#{groupname}))"</pre>
findGroupTemplate	A template that specifies how to produce the group name from the group object returned by the <code>findGroupPath</code> query. Like the <code>userTemplate</code> , this template is used to correct the case of a group name when you add LDAP group profiles in Workbench. Therefore, the value returned by this template should match the name entered in the User Settings tool, except for possible differences in case.
serviceUsername	<p>The user name of an administrator login to the LDAP server specified in the <code>serverInfo</code> parameter. For example: "Manager@example.com" or "cn=Manager,dc=example,dc=com".</p> <p>If no value is specified for this option, Workbench attempts to authenticate anonymously.</p>
servicePassword	The password to use in conjunction with the <code>serviceUsername</code> value.
serviceAuthentication	Specifies the method of authentication that should be used in connecting to the LDAP server as the administrator account. The permitted values are <code>none</code> , <code>simple</code> , or <code>EXTERNAL</code> .
authentication	Specifies the method of authentication that should be used in binding to the LDAP server as a user account. The permitted values are <code>none</code> , <code>simple</code> , or <code>EXTERNAL</code> .
ldapBindAuthentication	<i>Not supported in Workbench 3.1.1</i> Optional. By default this is set to <code>true</code> , and Workbench authenticates users by rebinding as the user to the LDAP system, thereby employing the LDAP system's own authentication mechanism.
loginName	Optional. A template login name that will be used to bind to the LDAP server. Default value is <code>%{dn}</code> .
passwordAttribute	<i>Not supported in Workbench 3.1.1</i> Optional. The name of the attribute on the user object that contains the user's password. Used only if <code>ldapBindAuthentication</code> is set to <code>false</code> . The field specified must contain the user's password in clear text. By default this is set to <code>userPassword</code> .

Parameter	Description
checkPasswords	<i>Not supported in Workbench 3.1.1</i> Optional. Determines whether Workbench checks passwords during logins. Default value is <code>true</code> . If set to <code>false</code> , Workbench uses only the user name to authenticate from the LDAP directory.
useSSL	Optional. Default value is <code>false</code> . If set to <code>true</code> , Workbench attempts to make mutually authenticated SSL connections to the LDAP server. If you set the parameter, ensure that you have configured the LDAP server to use SSL and that the value of <code>serverInfo</code> has the protocol specified as <code>ldaps://</code> with an SSL port.
keyStoreLocation	Used only if <code>useSSL=true</code> . The location of the Java keystore, which stores keys and certificates. The keystore is where Java gets the certificates to be presented for authentication. The location of the keystore is OS-dependant, but is often stored in a file named <code>.keystore</code> in the user's home directory.  Note: Even if this location is on a Windows system, the path uses forward slashes, (/) not backslashes (\).
keyStorePassphrase	Used only if <code>useSSL=true</code> . The passphrase used to open the keystore file.

Configuration parameters for identity information stored in LDAP

The LDAP configuration profile allows you to specify templates to extract identity information from LDAP user or group objects.

Workbench does not store any identity information such as first name, last name, or email address for LDAP users or groups. Instead, Workbench looks up this information in the LDAP directory when needed. The LDAP configuration profile allows you to specify templates to extract identity information from LDAP user or group objects, but they are not required for authentication via LDAP.

Workbench looks up the identity information for a user or group when you use the `Check Name` function on the **Add User** page to confirm that you are adding the correct LDAP user or group. If you do not specify templates for retrieving identity information, the fields are not filled in when you use `Check Name`.

Parameter	Description
firstNameTemplate	A template that specifies how to produce the user's first name from the user object, for example, <code>%{#firstNameAttribute}</code> .
lastNameTemplate	A template that specifies how to produce the user's last name from the user object, for example, <code>%{#lastNameAttribute}</code> .

Parameter	Description
<code>emailTemplate</code>	A template that specifies how to produce the user's email address from the user object, for example, <code>%{#emailAttribute}</code> , or <code>%{usernameField}@companydomain.com</code> .
<code>findGroupEmailTemplate</code>	A template that specifies how to produce the email address associated with a group in LDAP from the group object.

LDAP path parameters

The `userPath`, `groupPath`, and `findGroupPath` parameters, when appended to the URL in the `serverInfo` parameter, must conform to RFC 2255.

This means that certain characters must be encoded in order for the path parameters to form a valid LDAP URL when appended to the value of the `serverInfo` parameter. Both LDAP and URL encoding may apply to these strings depending on your data. If possible, verify the URL by passing it to your LDAP server before specifying it in the configuration for Workbench.

LDAP encoding affects reserved characters such as the comma (,), equals sign (=), and question mark (?). These characters must be escaped by prepending a backslash (\) when they are not used for their reserved purpose, for example if they appear within a common name or organizational unit.

URL encoding affects characters that are invalid for URLs, such as non-ASCII characters and any unsafe characters as defined in RFC 1738. This includes reserved LDAP characters when they are not used for their reserved purpose. These characters must be replaced with the % sign followed by the appropriate hex code.

For example, if you have the following string as part of your `userPath`:

```
ou=Endeca Technologies, Inc.
```

Applying LDAP encoding produces the following result:

```
ou=Endeca Technologies\, Inc.
```

Applying URL encoding to the LDAP-encoded string produces:

```
ou=Endeca%20Technologies%5C%2C%20Inc.
```

Any non-ASCII characters or any other characters that are not valid in an LDAP URL must also be properly encoded in the string that you specify in the LDAP login profile.

About specifying multiple values for parameters in the LDAP login profile

You can specify multiple LDAP servers and multiple values for the `groupPath` element.

If you specify multiple LDAP servers, the servers are assumed to be equivalent. The choice of which LDAP server to contact is made randomly. If an LDAP server cannot be reached, the `LoginModule` plug-in proceeds through the remaining servers in order of configuration, wrapping if necessary. For example, if five servers are configured and Server 3 is the first to be contacted, the remaining order of contact is Server 4, Server 5, Server 1, and finally Server 2.

You can specify multiple LDAP servers with multiple instances of the `serverInfo` parameter, by using the format:

```
serverInfo.n = "ldap://server_url:port_number"
```

For example:

```
serverInfo.0="ldaps://globalcatalog.corp.example.com:3269"
serverInfo.1="ldap://globalcatalog.us.example.com:3009"
```

You can also specify multiple values for the `groupPath` attribute by using the same format, for example:

```
groupPath.0="/ou=groups,dc=example,dc=com??sub?(member=%{#dn})"
groupPath.1="/dc=example,dc=com?memberOf?sub?(AccountName=%{#username})"
groupPath.2="/dc=example,dc=com?memberOf?sub?(CN=%{#dn})"
```

If you specify more than one `groupPath`, Workbench sends all the queries to the LDAP server to discover the groups of which a user is a member.

You can specify corresponding values for `groupTemplate` for each `groupPath`. In this case, the value for `groupTemplate.0` is applied to the results of the `groupPath.0` query, `groupTemplate.1` is applied to the results of `groupPath.1`, and so on.

For example:

```
groupTemplate.0="%{#dn:0}"
groupTemplate.1="%{#memberOf:0}"
groupTemplate.2="%{#memberOf:0}"
```

Specifying the location of the LDAP login configuration file

By default, Workbench uses `%ENDECA_TOOLS_CONF%\conf\Login.conf` (on Windows) or `$ENDECA_TOOLS_CONF/conf/Login.conf` (on UNIX) as the LDAP login configuration file.

If you are running the Endeca Tools Service as a Windows service, see the section "Specifying the location of the LDAP login configuration file using Windows Services."

You can substitute any configuration file that includes a LDAP login profile named `webstudio`. The file does not have to be named `Login.conf`, but it must be saved in UTF-8 format.

If you want to store the configuration file in a different location, you can pass this location to the Java JVM. How you specify the location depends on how you run the Endeca Tools Service.

If you are running the Endeca Tools Service on Windows from the command line or on UNIX:

1. Navigate to `%ENDECA_TOOLS_ROOT%\server\bin` (on Windows) or `$ENDECA_TOOLS_ROOT/server/bin` (on UNIX).
2. Open the `setenv.bat` or `setenv.sh` file.
3. Locate the line that sets `JAVA_OPTS`:

```
set JAVA_OPTS=-Xmx1024m -XX:MaxPermSize=128m -Djava.security.auth.login.config=%ENDECA_TOOLS_CONF%/conf/Login.conf
```

4. Change the `-Djava.security.auth.login.config` parameter to point to the location of your configuration file on the file system.

Specifying the location of the LDAP login configuration file using Windows Services

By default, Workbench uses `%ENDECA_TOOLS_CONF%\conf>Login.conf` (on Windows) or `$ENDECA_TOOLS_CONF/conf/Login.conf` (on UNIX) as the LDAP login configuration file.

If you are running the Endeca Tools Service on UNIX or on Windows from a command line, see "Specifying the location of the LDAP login configuration file."

You can substitute any configuration file that includes a LDAP login profile named `webstudio`. The file does not have to be named `Login.conf`, but it must be saved in UTF-8 format.

If you want to store the configuration file in a different location, you can pass this location to the Java JVM. How you specify the location depends on how you run the Endeca Tools Service.

If you are running the Endeca Tools Service on Windows from the command line or on UNIX:

1. Open the Registry Editor.
2. Navigate to the `HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software Foundation\Procrun 2.0\EndecaToolsService\Parameters\Java\Options` key.
3. Right click **Options** in the right pane and select **Modify**. The **Edit Multi-String** dialog box displays.
4. Locate the following parameter:

```
-Djava.security.auth.login.config=%ENDECA_TOOLS_CONF%/conf/Login.conf
```

5. Change the path to point to the location of your configuration file.
6. Click **OK**.

Troubleshooting user authentication in Workbench with LDAP enabled

If a user cannot log in to Workbench, one of several error messages displays.

Incorrect Username or Password

If the user is entering the correct LDAP user name and password, there may be a manually configured Workbench user in the same application with the same user name or a Workbench administrator with the same user name.

A user with a manually configured profile always takes precedence over a user authenticating via LDAP. For more details about the behavior of users with the same name, see "User profiles for LDAP users and groups."

An error occurred while trying to validate your credentials

This error displays when any error occurs other than a user name-password mismatch or an absence of permissions. It can indicate anything from a connectivity issue with the LDAP server to a mistake in the configuration in the LDAP login configuration file located in `%ENDECA_TOOLS_CONF%\conf>Login.conf`. For more information about the login profile, see "Configuration parameters for the LDAP login profile."

Check the Workbench log, located in `%ENDECA_TOOLS_CONF%\logs\webstudio.#.log` for more information about the causes of authentication failures. In most cases, the solution is to adjust the

LDAP query strings to return the desired results. If possible, test the query URLs against your LDAP server using an independent tool in order to confirm that they behave as expected and that each query for a user or group that exists in the directory returns a unique user or group object.

Configuring Communication with Other Endeca Components

This section discusses configuring Workbench to communicate with other Endeca components.

About Workbench interaction with the Endeca Configuration Repository

The Endeca Configuration Repository is a Web application that runs in the Endeca Tools Service.

The Endeca Configuration Repository uses a JSR-170-compliant Java Content Repository to store configuration related to Endeca applications. It also hosts several tools that are accessed via Workbench, including Experience Manager and the Thesaurus.

Specifying Workbench authentication credentials for the Endeca Configuration Repository

By default, Workbench authenticates itself to the Endeca Configuration Repository using the JCR repository's default admin user credentials.

To specify the Workbench authentication credentials for the Endeca Configuration Repository:

1. Change the admin password in the Endeca Configuration Repository by submitting a POST request to `http://<WorkbenchHost>:8006/ifcr/system/userManager/user/admin.changePassword.html` with the following parameters:

Parameter	Value
<code>oldPwd</code>	The current password. The default password for the admin user is admin.
<code>newPwd</code>	The desired value for the new password.
<code>newPwdConfirm</code>	The desired value for the new password.

The following is an example using the curl tool:

```
curl -FoldPwd=admin -FnewPwd=newpassword -FnewPwdConfirm=newpassword \
http://admin:admin@localhost:8006/ecr/system/userManager/user/admin.changePassword.html
```

2. Update the credentials that Workbench uses to connect to the Endeca Configuration Repository.
 - a) Stop the Endeca Tools Service.
 - b) Navigate to %ENDECA_TOOLS_CONF%\conf (on Windows) or \$ENDECA_TOOLS_CONF\conf (on UNIX).
 - c) Open the webstudio.properties file.
 - d) Locate the ifcr.password property, for example:

```
# The password Workbench uses to authenticate as the admin user for
the Endeca Configuration Repository
#ifcr.password=admin
```
 - e) Uncomment the line and set the value of the property to the new password, for example:

```
ifcr.password=newpassword
```
 - f) Save and close the webstudio.properties file.
 - g) Start the Endeca Tools Service.
3. Update the credentials used by Deployment Template scripts to connect to the Endeca Configuration Repository:

- a) Navigate to the script directory in your deployed application, for example, C:\Endeca\apps\discover\script.

- b) Locate the SiteHandler bean near the end of the file:

```
<spr:bean id="SiteHandler" class="com.endeca.dt.ifcr.SiteHandler">
  <spr:property name="appName" value="discover"/>
  <spr:constructor-arg type="java.lang.String">
    <spr:value>http://example.com:8006/ifcr</spr:value></spr:constructor-arg>
  <spr:constructor-arg type="java.lang.String">
    <spr:value>admin</spr:value></spr:constructor-arg>
  <spr:constructor-arg type="java.lang.String">
    <spr:value>admin</spr:value></spr:constructor-arg>
</spr:bean>
```

- c) Update the value of the last string argument to the new password, for example:

```
<spr:bean id="SiteHandler" class="com.endeca.dt.ifcr.SiteHandler">
  <spr:property name="appName" value="discover"/>
  <spr:constructor-arg type="java.lang.String">
    <spr:value>http://example.com:8006/ifcr</spr:value></spr:constructor-arg>
  <spr:constructor-arg type="java.lang.String">
    <spr:value>admin</spr:value></spr:constructor-arg>
  <spr:constructor-arg type="java.lang.String">
    <spr:value>newpassword</spr:value></spr:constructor-arg>
</spr:bean>
```

For further information about Sling user management, consult the Apache documentation.

Configuring the shared key for Workbench extensions hosted in the Endeca Configuration Repository

Several of the core tools provided with Workbench are implemented as Workbench extensions and hosted in the Endeca Configuration Repository. These tools make use of the token-based authentication mechanism for Workbench extensions.

The authentication mechanism for Workbench extensions depends on a shared secret that is used to build an authentication token. The extension can then use this token to ensure that a given request comes from Workbench. All of the core Workbench tools that are hosted in the Endeca Configuration Repository use the same shared secret.

To update the shared secret used to authenticate users to Endeca Configuration Repository-hosted tools:

1. Stop the Endeca Tools Service.
2. Navigate to %ENDECA_TOOLS_CONF%\conf (on Windows) or \$ENDECA_TOOLS_CONF\conf (on UNIX).
3. Open the `ws-extensions.xml` file.
4. For each extension hosted in the Endeca Configuration Repository (with a URL that includes the path `/ifcr/sites/${EAC_APP}`), edit the value of the `sharedSecret` attribute, for example:

```
<extension id="groupmanager"
  defaultName="Group Manager"
  defaultDescription="Modify the Groups to organize content in
your application."
  url="/ifcr/sites/${EAC_APP}/zones.html/?times-
tamp=${TS}&auth=${AUTH}"
  sharedSecret="n3wSecre+"
  role="admin"
  height="800"/>
```

Workbench uses this value for the shared secret when building the authentication token that it passes to the tool.

5. Save and close the `ws-extensions.xml` file.
6. Open the `webstudio.properties` file.
7. Locate the `sharedSecret` property, for example:

```
# Shared secret used for all Endeca Configuration Repository-hosted tools
# Value should match the shared secret defined for each tool
# in ws-extensions.xml
sharedSecret=123456789
```

8. Set the value of the property to match the value you specified in `ws-extensions.xml`, for example:

```
sharedSecret=n3wSecre+
```

Workbench tools hosted in the Endeca Configuration Repository use this value for the shared secret to verify that a request is coming from a user who is logged in to Oracle Endeca Workbench.

9. Save and close the `webstudio.properties` file.
10. Start the Endeca Tools Service.

About Workbench interaction with the MDEX Engine

Workbench both queries the MDEX Engine for information and publishes configuration to it.

Experience Manager queries the MDEX Engine for record and dimension information that a content administrator can use to configure dynamic content. Examples include:

- specifying a navigation state as part of a location trigger
- selecting records or a navigation state for content spotlighting
- selecting records or dimension values for boost and bury

Workbench publishes configuration to the MDEX Engine, including:

- Content item configuration from Experience Manager
- Content collections
- Thesaurus entries

About specifying the MDEX Engine that Workbench queries

You can designate a specific Dgraph or Agraph that Workbench should always query by defining a `WebStudioMDEX` property on the appropriate component and setting it to `true`.

By default, Oracle Endeca Workbench queries the first MDEX Engine returned by the EAC. If you have multiple MDEX Engine components in your environment, you can designate a specific MDEX Engine that Workbench uses for all MDEX queries.

There are two ways to designate a specific Dgraph for use with Workbench:

- Specify the property on the Dgraph component in the EAC provisioning file, or via the `AppConfig.xml` file in the Deployment Template. For details, see the *Endeca Application Controller Guide* or the *Deployment Template Usage Guide*, respectively.
- Specify the property on the Dgraph component using the EAC Admin Console in Oracle Endeca Workbench. This option is provided as a convenience for development or staging environments. In a production environment, Oracle recommends using the Deployment Template.

Priority order for selecting an MDEX Engine

Workbench chooses an MDEX Engine to query based on the following order:

1. The first Dgraph component returned by the EAC that has `WebStudioMDEX = true`
2. The first Dgraph component returned by the EAC that does not have `WebStudioMDEX = true`

About specifying which Dgraphs to update with configuration changes

By default, Workbench updates all Dgraphs that are defined in your application whenever a user saves changes to the application configuration (including changes to content items, thesaurus entries, and so on).

You can specify which Dgraphs are updated with configuration changes made in Workbench. Omitting some Dgraphs from the update process can offer performance improvements when saving changes. It also allows you to control which servers can be updated directly by business users in Workbench. You specify that a Dgraph should not be updated by Workbench by defining a custom EAC property of `WebStudioSkipConfigUpdate` set to `true` on the appropriate component. You can do this using one of the following methods:

- Specify the property on the Dgraph component in the EAC provisioning file or via the `AppConfig.xml` file in the Deployment Template. For details, see the *Endeca Application Controller Guide* or the *Deployment Template Usage Guide*.
- Specify the property on the Dgraph component using the EAC Admin Console in Workbench. This option is provided as a convenience for development or staging environments. In a production environment, Oracle recommends using the Deployment Template.

Because only Dgraphs are updated with configuration changes, this property does not apply to Agraph components.

About accessing files on remote servers

Experience Manager occasionally needs to access files hosted on a different server. Certain security issues may apply.

Experience Manager makes an anonymous request to the file server to fetch resources. That is, even though content administrators are authenticated when they log in to Experience Manager, the tool does not use their credentials when requesting images, editors, or other files.

Experience Manager also respects the cross-domain policy file of the server hosting the external files. To ensure that Experience Manager can load these files, place a `crossdomain.xml` file on the file server. This file allows you to enable access to media on this server from a specific IP address, a specific domain, or any domain. If this policy file does not allow access from the Experience Manager server, a security error similar to the following displays when Experience Manager attempts to load the resource:

```
Error #2044: Unhandled securityError: . text=Error #2048: Security sandbox violation: http://pagebuilder.mycompany.com/tmgr/tmgr.swf cannot load data from http://www.example.com/images/3column.gif.
```

The following example of a `crossdomain.xml` file enables access from any domain to files hosted on `www.example.com`:

```
<?xml version="1.0"?>
<!-- http://www.example.com/crossdomain.xml -->
<cross-domain-policy>
  <allow-access-from domain="*" />
</cross-domain-policy>
```

You can also restrict access to specific domains or IP addresses, for instance, for the server on which Experience Manager is running. Wildcards are allowed in domain names but not IP addresses. The following example shows a policy file for `www.example.com` that allows access from anywhere in the `example.com` domain, `www.customer.com`, and `105.216.0.40`. It includes a `by-content-type` meta-policy that allows policy files with a Content-Type of exactly `text/x-cross-domain-policy`:

```
<?xml version="1.0"?>
<!-- http://www.example.com/crossdomain.xml -->
<cross-domain-policy>
  <site-control permitted-cross-domain-policies="by-content-type" />
  <allow-access-from domain="*.example.com" />
  <allow-access-from domain="www.customer.com" />
  <allow-access-from domain="105.216.0.40" />
</cross-domain-policy>
```



Important: In addition to cross-domain policy files, you must set up a meta-policy for each server. These are configuration settings that manage what cross-domain policies are allowed on a server. A meta-policy file is required with the use of Flash 10.

For more information about meta-policies and cross-domain policy files, see the Adobe Flash documentation.

Configuring SSL for Oracle Endeca Workbench

This section describes how to configure your Workbench installation to use SSL for Web browser connections. Workbench does not support SSL communication with Endeca components (such as the EAC and MDEX Engine).

About configuring SSL in Workbench

SSL is disabled by default for Workbench as a server.

To enable SSL security between Workbench and its clients, you must do the following:

- Enable the SSL version of Workbench.
- Set up a certificate for the Workbench server. For details, see the *Endeca Security Guide*. The server certificate for Workbench must be issued to the fully qualified domain name of the server.
- Modify the `server.xml` file for the Endeca Tools Service to enable the HTTPS connector and point to the new keystore.

Clients can make secure connections to Workbench either by taking advantage of a redirect from the non-SSL port or, if you have disabled the non-SSL port or do not wish to use the redirect, by making an HTTPS connection directly to the SSL port.

Workbench supports version 3.0 of the Secure Sockets Layer (SSL) protocol for its communication endpoints.

Enabling the SSL version of Oracle Endeca Workbench

The non-SSL version of Oracle Endeca Workbench is installed by default.

To enable the SSL version of Workbench:

1. Stop the Endeca Tools Service.
2. Navigate to `%ENDECA_TOOLS_CONF%\conf\Standalone\localhost` (on Windows) or `$ENDECA_TOOLS_CONF/conf/Standalone/localhost` (on UNIX).
3. Open the `ROOT.xml` file.
4. Locate the line in which the `docBase` is defined.

For example:

```
docBase="${catalina.base}/../webapps/workbench-legacy-tools-3.1.1.war"
```



Note: The file name in the example may not match the one in your installation.

5. Change this to point to the SSL version of the WAR by adding `-ssl` to the filename. For example:

```
docBase="${catalina.base}/../workbench-legacy-tools-3.1.1-ssl.war"
```

6. Save and close the file.
7. Start the Endeca Tools Service.

If you want to restore the non-SSL version at a later date, you can reverse the process by editing the `ROOT.xml` file accordingly.

Modifying the `server.xml` for the Endeca Tools Service

Before you can use SSL with Workbench, you must edit its `server.xml` file as described.

This procedure assumes you have already generated server certificates for Workbench as described in the *Endeca Security Guide* and uploaded them to the Endeca Workbench server.

To enable the HTTPS connector:

1. Stop the Endeca Tools Service.
2. Navigate to `%ENDECA_TOOLS_CONF%\conf` (on Windows) or `$ENDECA_TOOLS_CONF/conf` (on UNIX).
3. Open the `server.xml` file.
4. Locate and remove the comments around the Connector element for port 8446 as follows:

```
<Connector port="8446" SSLEnabled="true"
protocol="org.apache.coyote.http11.Http11Protocol"
maxPostSize="0"
maxThreads="150" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="conf/eac.ks" keystorePass="eacpass"
truststoreFile="conf/ca.ks" truststorePass="eacpass"
/>
```

5. Optionally, change the port number to something other than 8446 if you do not want to use that default.

If you do not use the default port, update the `redirectPort` attribute on the non-SSL HTTP connector to point to the new port as in the following example:

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8006 -->
<Connector port="8006" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="true" redirectPort="8446" acceptCount="10"
connectionTimeout="60000" disableUploadTimeout="true" debug="0"
URIEncoding="UTF-8" />
```

6. If you want to disable the redirect from the non-secure port to the secure port, comment out the non-SSL connector in the `server.xml` file. By default, the redirect is enabled.



Caution: If you choose to disable the non-SSL connector, the Deployment Template and the `emgr_update` utility cannot communicate with Oracle Endeca Workbench and you must manually update your application's instance configuration.

7. Update the `keystoreFile`, `keystorePass`, `truststoreFile`, and `truststorePass` with the appropriate values for your certificates.

The `keystoreFile` and `truststoreFile` values should be the paths to the location where you uploaded your keystore and truststore files. These paths can be specified as absolute paths, or paths relative to `ENDECA_TOOLS_CONF`, although the files themselves can be located anywhere on the server.

8. Save and close the file.
9. Start the Endeca Tools Service.

Chapter 6

Configuring Workbench System Logs

This section discusses Workbench system logs and how to configure their behavior.

About Workbench system logs

The Oracle Endeca Workbench logs are located in `%ENDECA_TOOLS_CONF%\logs` (on Windows) or `$ENDECA_TOOLS_CONF/logs` (on UNIX).

The following logs can be found in this directory:

File name	Description
<code>webstudio.log</code>	System log for Oracle Endeca Workbench, including activity such as user logins, updates to instance configuration, and Oracle Endeca Workbench errors.
<code>webstudio_audit.log</code>	Audit log for activity such as dynamic business rule and search configuration changes. Logging for rules includes the name of the rule being modified, when it was modified, who modified it (based on Endeca Workbench user name), and any note associated with the change.

By default, the Oracle Endeca Workbench system log and audit log have a maximum size of 1MB. Each of the logs is part of a four-log rotation.

Configuring the Oracle Endeca Workbench logs

By editing the configuration file, you can control the log level, the maximum file size, and the number of files in the log rotation. You can also optionally direct the output of any Workbench logger to the console or to another file.

Both the Workbench system log and audit log are configured by the `webstudio.log4j.properties` file, located in `%ENDECA_TOOLS_CONF%\conf` (on Windows) or `$ENDECA_TOOLS_CONF/conf` (on UNIX).

To configure the behavior of the Oracle Endeca Workbench logs:

1. Stop the Endeca Tools Service.
2. Navigate to %ENDECA_TOOLS_CONF%\conf (on Windows) or \$ENDECA_TOOLS_CONF/conf (on UNIX).
3. Open the `webstudio.log4j.properties` file.
4. Modify the configuration file as needed. For more information, see the comments in `webstudio.log4j.properties` and the log4j documentation at <http://logging.apache.org/log4j/>.
5. Save and close the file.
6. Start the Endeca Tools Service.

Customizing Oracle Endeca Workbench

This section describes how to customize the Oracle Endeca Workbench interface and how to add extensions to Oracle Endeca Workbench.

The navigation menu and launch page

You can configure the items in the navigation menu on the left and on the launch page of Oracle Endeca Workbench by modifying the `ws-mainMenu.xml` file in `%ENDECA_TOOLS_CONF%\conf` (on Windows) or `$ENDECA_TOOLS_CONF/conf` (on UNIX).

By editing `ws-mainMenu.xml`, you can do any of the following:

- Add a new menu item.
- Remove an item from the menu.
- Specify the order in which the menu items display.
- Specify whether an item is in the top-level menu or in a submenu.
- Specify whether a menu item displays on the launch page.

Navigation menu nodes

A menu item is either a leaf or a node. A node is a top-level menu item that does not link directly to any pages.

Instead it has children that are leaf items and are displayed in a submenu. Each node is defined in a `<menunode>` element in `ws-mainMenu.xml` that takes the following attributes:

Attribute name	Attribute value
<code>id</code>	The <code>id</code> of a predefined node in Oracle Endeca Workbench or a unique string identifying a custom node. For more information on predefined nodes, see “Predefined menu nodes in Oracle Endeca Workbench.”
<code>defaultTitle</code>	The display name for this node that appears in the navigation menu. This attribute is required for all custom nodes.

A `menunode` element requires one or more child `menuitem` elements.

This example of a `ws-mainMenu.xml` file defines a custom menu node with extensions as its child items.

```
<?xml version="1.0" encoding="UTF-8"?>
<mainmenu xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="mainMenu.xsd">
  <menunode id="myextensions" defaultTitle="My Extensions">
    <menuitem id="extensionA"/>
    <menuitem id="extensionB"/>
  </menunode>
</mainmenu>
```

Related Links

[Predefined menuitem elements](#) on page 46

This section is a reference table listing all of the predefined pages and corresponding ids available in the `ws-mainMenu.xml` file.

[About navigation menu leaf items](#) on page 45

A leaf is a menu item that links to a page, and also has an entry on the launch page.

Node titles for multiple locales

If you customize a menu for multiple locales in Workbench, you can optionally specify localized titles for custom menu nodes in a `titles` element within `menunode` that contains one or more title elements.

The title element requires a `locale` attribute whose value is a valid ISO language code.

This example of a `ws-mainMenu.xml` file defines a custom menu node with titles in both English and French.

```
<?xml version="1.0" encoding="UTF-8"?>
<mainmenu xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="mainMenu.xsd">
  <menunode id="myextensions" defaultTitle="My Extensions">
    <titles>
      <title locale="en">Access Extensions</title>
      <title locale="fr">Accéder aux extensions</title>
    </titles>
    <menuitem id="extensionA"/>
    <menuitem id="extensionB"/>
  </menunode>
</mainmenu>
```

Workbench checks for a title that matches the locale defined in the current installation of Workbench. If no matching localized title is found, the `defaultTitle` value is used.

Predefined menu nodes in Oracle Endeca Workbench

There are several predefined menu nodes in Oracle Endeca Workbench. You can specify the placement of the predefined nodes in the menu and what items display under them, but you cannot modify the titles or specify localized titles.

The predefined nodes in Oracle Endeca Workbench are as follows:

Node id	Node description
reports	Reports
settings	Application Settings
administration	Administration

About navigation menu leaf items

A leaf is a menu item that links to a page, and also has an entry on the launch page.

A leaf can be either in the top-level menu or in a submenu as the child of a node. Leaf items cannot have child items. Menu items display in the order in which they are listed in `ws-mainMenu.xml`.

Each leaf in the menu is defined in a `menuItem` element in `ws-mainMenu.xml` that takes the following attributes:

Attribute name	Attribute value	Required?
id	The <code>id</code> of a predefined page in Endeca Workbench or the <code>id</code> of an extension as defined in <code>ws-extensions.xml</code> . For more information about extensions, see "Workbench extensions."	yes
onLaunchPage	<i>Deprecated and ignored in release 3.1.1.</i> If set to true, the menu item displays on the launch page in the order in which it is listed in <code>ws-mainMenu.xml</code> . Default value is false.	no



Note: For a full list of predefined pages and their corresponding ids, see "Predefined menuItem elements."

This example of a `ws-mainMenu.xml` file defines a menu that shows top-level leaf items, items nested within a predefined node.

```
<?xml version="1.0" encoding="UTF-8"?>
<mainmenu xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="mainMenu.xsd">
  <menuItem id="xmgr"/>
  <menuItem id="thesaurus"/>
  <menunode id="reports">
    <menuItem id="reports.today"/>
    <menuItem id="reports.daily"/>
    <menuItem id="reports.weekly"/>
  </menunode>
  <menunode id="settings">
    <menuItem id="user-segments"/>
    <menuItem id="preview-settings"/>
    <menuItem id="report-settings"/>
  </menunode>
  <menunode id="administration">
```

```

    <menuitem id="locks" />
    <menuitem id="user-management" />
    <menuitem id="eac-settings" />
  </menunode>
  <menuitem id="eac-admin-console" />
</mainmenu>

```

Related Links

[Predefined menuitem elements](#) on page 46

This section is a reference table listing all of the predefined pages and corresponding ids available in the `ws-mainMenu.xml` file.

[Navigation menu nodes](#) on page 43

A menu item is either a leaf or a node. A node is a top-level menu item that does not link directly to any pages.




[Workbench extensions](#) on page 47

Extensions enable you to incorporate Web applications related to your Endeca implementation as plug-ins to Oracle Endeca Workbench.

Predefined menuitem elements

This section is a reference table listing all of the predefined pages and corresponding ids available in the `ws-mainMenu.xml` file.

The predefined pages and their corresponding ids are as follows:

Workbench page	Menu item id
Experience Manager  Note: This menu item is only available in installations of Oracle Endeca Commerce that include Experience Manager.	xmgr
Rule Manager  Note: This menu item is only available in the Oracle Endeca Commerce Guided Search package.	rmgr
Thesaurus	thesaurus
Today's Reports	reports.today
Daily Reports	reports.daily
Weekly Reports	reports.weekly
Report Scheduler	report-settings
Preview Settings  Note: This menu item is only available in installations of Oracle Endeca Commerce that include Experience Manager.	preview-settings

Workbench page	Menu item id
EAC Admin Console	eac-admin-console
EAC Connection Settings	eac-settings
User Segments	user-segments
User Management	user-management

Related Links

[Navigation menu nodes](#) on page 43

A menu item is either a leaf or a node. A node is a top-level menu item that does not link directly to any pages.

[About navigation menu leaf items](#) on page 45

A leaf is a menu item that links to a page, and also has an entry on the launch page.

[Extension element attributes](#) on page 48

This section provides a reference table of required and optional extension element attributes.

Updating the Oracle Endeca Workbench menu and launch page

The menu items on the launch page of Oracle Endeca Workbench are configurable.

You can configure the items in the menu and on the Workbench launch page by modifying the `ws-mainMenu.xml` file in `%ENDECA_TOOLS_CONF%\conf` (on Windows) or `$ENDECA_TOOLS_CONF/conf` (on UNIX).

To update the navigation menu and launch page:

1. Stop the Endeca Tools Service.
2. Navigate to `%ENDECA_TOOLS_CONF%\conf` (on Windows) or `$ENDECA_TOOLS_CONF/conf` (on UNIX).
3. Open `ws-mainMenu.xml` in a text editor and add or modify menu items as necessary.
4. Save and close the file.
5. Start the Endeca Tools Service.

Workbench extensions

Extensions enable you to incorporate Web applications related to your Endeca implementation as plug-ins to Oracle Endeca Workbench.

An extension can be as simple as a static Web page or it can provide sophisticated functionality to control, monitor, and configure your Endeca applications. Extensions can be hosted on the same server as Workbench or on another server.

Related Links

[About navigation menu leaf items](#) on page 45

A leaf is a menu item that links to a page, and also has an entry on the launch page.

About configuring extensions in Oracle Endeca Workbench

Extensions are defined in the `ws-extensions.xml` file in `%ENDECA_TOOLS_CONF%\conf` (on Windows) or `$ENDECA_TOOLS_CONF/conf` (on UNIX).

The default `ws-extensions.xml` file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>

<extensions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="extensions.xsd">

</extensions>
```

Each extension is defined in an extension element within extensions. You can specify as many additional extensions as you need by adding more extension elements. For a full list of list of required and optional attributes, see "Extension element attributes."

This example of a `ws-extensions.xml` file defines a simple extension that enables a link to the Endeca Web site.

```
<?xml version="1.0" encoding="UTF-8"?>

<extensions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="extensions.xsd">
  <extension id="endecaHome"
    defaultName="Endeca home page"
    defaultDescription="Visit the Endeca home page"
    url="http://www.endeca.com" />
</extensions>
```

Related Links

[Updating the Oracle Endeca Workbench menu and launch page](#) on page 47

The menu items on the launch page of Oracle Endeca Workbench are configurable.

[Enabling extensions in Oracle Endeca Workbench](#) on page 49

You enable Workbench extensions by editing the `ws-extensions.xml` file.

Extension element attributes

This section provides a reference table of required and optional extension element attributes.

The extension element takes the following attributes:

Attribute name	Attribute value	Required?
<code>id</code>	A unique string identifying this extension. Do not define an extension with the same <code>id</code> as one of the predefined Oracle Endeca Workbench pages. For a list of predefined Workbench pages and their <code>ids</code> , see the reference table in "Predefined <code>menuItem</code> elements."	yes
<code>defaultName</code>	The display name for this extension that appears in the navigation menu and launch page in Oracle Endeca Workbench.	yes

Attribute name	Attribute value	Required?
defaultDescription	A brief description of this extension that appears on the launch page in Oracle Endeca Workbench.	yes
url	An absolute or relative URL to this extension. The extension must be a Web application reachable through HTTP or HTTPS, but it does not have to run on the same server as Oracle Endeca Workbench. If the extension is hosted on the same server as Oracle Endeca Workbench, the extension URL can be site relative (omitting the host name and port and beginning with the leading slash).	yes
launchImageUrl	An absolute or relative URL to a custom image for this extension's entry on the launch page. (Relative URLs are relative to <hostname>:8006.	no
helpUrl	<i>Unused in release 3.1.1.</i>	
role	This attribute is only used when you want an extension to display for administrators on the Workbench Admin page. The only value allowed is "admin." Any other value is ignored.	no
height	<i>Deprecated and ignored in release 3.1.1.</i> The height in pixels of the frame in which the extension is displayed. The default value is 500 pixels.	no
sharedSecret	A shared key that Oracle Endeca Workbench uses to calculate the authentication token.	no

Related Links

[Enabling extensions in Oracle Endeca Workbench](#) on page 49

You enable Workbench extensions by editing the `ws-extensions.xml` file.

Enabling extensions in Oracle Endeca Workbench

You enable Workbench extensions by editing the `ws-extensions.xml` file.

To enable extensions in Oracle Endeca Workbench:

1. Stop the Endeca Tools Service.
2. Navigate to `%ENDECA_TOOLS_CONF%\conf` (on Windows) or `$ENDECA_TOOLS_CONF/conf` (on UNIX).
3. Open `ws-extensions.xml` in a text editor and add or modify extensions as necessary.



Note: In addition to adding an extension to Oracle Endeca Workbench, you must also enable links to the new extension in the navigation menu and the launch page.

4. Save and close the file.
5. Start the Endeca Tools Service.

Related Links

[Updating the Oracle Endeca Workbench menu and launch page](#) on page 47

The menu items on the launch page of Oracle Endeca Workbench are configurable.

[About configuring extensions in Oracle Endeca Workbench](#) on page 48

Extensions are defined in the `ws-extensions.xml` file in `%ENDECA_TOOLS_CONF%\conf` (on Windows) or `$(ENDECA_TOOLS_CONF)/conf` (on UNIX).

[Extension element attributes](#) on page 48

This section provides a reference table of required and optional extension element attributes.

URL tokens and Workbench extensions

Oracle Endeca Workbench can pass information to an extension through URL tokens in order to enable the extension to authenticate users, connect to the EAC Central Server, and maintain its state if a user navigates away from the extension and back again during the same session.

You use URL tokens by specifying them in the `url` attribute of the extension definition in `%ENDECA_TOOLS_CONF%\conf\ws-extensions.xml`. The name of the URL parameter does not have to match the id of the token as listed in the preceding table.

For example, the following extension definition creates a URL that passes the EAC host, port, and application to the extension:

```
<?xml version="1.0" encoding="UTF-8"?>
<extensions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="extensions.xsd">
  <extension id="testExtension"
    defaultName="Test Extension"
    defaultDescription="Demonstrates extensions with tokens."
    url="http://www.example.com:8989/TestExtension/index.jsp?eac-host=
    ${EAC_HOST}&eac-port=${EAC_PORT}&eac-app=${EAC_APP}"
  </extension>
</extensions>
```

Note the use of the `&` entity in the `url` attribute in place of the ampersand in the URL. In general, you should ensure that the `ws-extensions.xml` file validates against the provided schema before updating Oracle Endeca Workbench with the new configuration.

Related Links

[URL token reference](#) on page 50

This section provides a complete list of all tokens available to pass to Workbench extensions.

[Token-based authentication for Workbench extensions](#) on page 51

You can enable extensions to authenticate users coming from Endeca Workbench by including an authentication token in the URL.

URL token reference

This section provides a complete list of all tokens available to pass to Workbench extensions.

The following tokens are available to pass to extensions:

Token ID	Token description
<code>\${AUTH}</code>	An MD5 hash value used to authenticate users coming from Oracle Endeca Workbench.
<code>\${EAC_APP}</code>	The name of the application that the Workbench user is logged in to.
<code>\${EAC_HOST}</code>	The host running the EAC Central Server to which Endeca Workbench is currently connected.
<code>\${EAC_PORT}</code>	The port on the EAC host through which Oracle Endeca Workbench and the EAC Central Server communicate.
<code>\${EXTENSION_ID}</code>	The id of the extension as defined in <code>ws-extensions.xml</code> .
<code>\${LOCALE}</code>	The locale of Oracle Endeca Workbench; this is the value of the <code>com.endeca.webstudio.locale</code> property in <code>webstudio.properties</code> .
<code>\${TS}</code>	The time, in milliseconds since 00:00:00 UTC January 1, 1970, when the user navigates to the extension.
<code>\${USERNAME}</code>	The username of the Workbench user accessing the extension.
<code>\${WEBSTUDIO_SESSIONID}</code>	The id of the user's current Workbench session. The extension can use this in combination with the <code>\${USERNAME}</code> token to maintain the state of the extension throughout a single Workbench session, for instance by storing the information in a cookie.

Related Links

[URL tokens and Workbench extensions](#) on page 50

Oracle Endeca Workbench can pass information to an extension through URL tokens in order to enable the extension to authenticate users, connect to the EAC Central Server, and maintain its state if a user navigates away from the extension and back again during the same session.

[About navigation menu leaf items](#) on page 45

A leaf is a menu item that links to a page, and also has an entry on the launch page.

[Token-based authentication for Workbench extensions](#) on page 51

You can enable extensions to authenticate users coming from Endeca Workbench by including an authentication token in the URL.

Token-based authentication for Workbench extensions

You can enable extensions to authenticate users coming from Endeca Workbench by including an authentication token in the URL.

Oracle Endeca Workbench calculates the value of the token by generating an MD5 hash from a portion of the URL and a shared secret. The portion of the URL that is used for the hash consists of everything after the host name and port, including the leading slash, but excluding the value of the AUTH token itself. The shared secret is a string that is specified in `ws-extensions.xml` and is also stored in the extension itself.

For example, the following `ws-extensions.xml` file defines an extension with a URL that uses the AUTH and TS tokens:

```
<?xml version="1.0" encoding="UTF-8"?>

<extensions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="extensions.xsd">
  <extension id="authExtension"
    defaultName="Authenticated Extension"
    defaultDescription="Demonstrates token-based authentication."
    url="http://localhost:8080/AuthExtension/index.jsp?timestamp=${TS}&
auth=${AUTH}"
    sharedSecret="secret!@#$$%^*(987654321" />
</extensions>
```

In this case, the value of the authentication token is the hash of a string that looks similar to this:

```
/AuthExtension/index.jsp?timestamp=1189702462936&auth=secret!@#$$%^*(987654321
```

The extension can verify that a user is coming from Oracle Endeca Workbench by calculating the hash of the same string and comparing the result to the value of the AUTH token. This ensures that the user visiting the extension has logged in to Oracle Endeca Workbench and has the permission (if any) that is required to access the extension.

Because the AUTH token is based in part on the URL, it is recommended that you include the time stamp of the request to introduce some variation in the value of the token. The time stamp can also be used to filter out stale requests and limit the possibility of an eavesdropper reusing the same URL to gain access to the extension.

The following Java code shows how the extension defined in the preceding example can authenticate users from Oracle Endeca Workbench:

```
// These values depend on what you defined in ws-extensions.xml
String extensionSecret="secret!@#$$%^*(987654321";
final String authTokenParameterName = "auth";
final String timeStampParameterName = "timestamp";

// Set the tolerance, in milliseconds, before a request is considered too
old
int allowedTimeStampSlackInMS = 5 * 60 * 1000;

// Calculate the hash of the substring of the URL and the shared secret
String url = request.getRequestURI() + "?" + request.getQueryString();
String findAuthToken = "&" + authTokenParameterName + "=";
url = url.substring(0, url.indexOf(findAuthToken) + findAuthToken.length());
String authCode = request.getParameter(authTokenParameterName);

MessageDigest md = MessageDigest.getInstance("MD5");
byte[] md5Hash = md.digest((url + extensionSecret).getBytes("UTF-8"));

StringBuffer hashCode = new StringBuffer();

for(int i : md5Hash)
{
  String str = Integer.toHexString(i+128);
```

```

if (str.length() < 2)
{
    str = "0" + str;
}
hashCode.append(str);
}

// Compare the hash to the value of the AUTH token
if (!hashCode.toString().equals(authCode))
{
    // Authentication fails because AUTH token did not match
}

// Compare the time stamp of the request to the current time stamp
long currentTime = new Date().getTime();
long ts = Long.parseLong(request.getParameter(timestampParameterName));

if ( Math.abs(ts - currentTime) > allowedTimeStampSlackInMS)
{
    // Authentication fails because request is too old
}

```

The example extension places the AUTH token at the end of the URL, making it more convenient to build the substring of the URL for the hash.

However, the AUTH token can be in any position in the URL. For instance, the URL can be defined in `ws-extensions.xml` is as follows:

```

url="http://localhost:8080/AuthExtension/index.jsp?auth=${AUTH}&timestamp=${TS}"

```

This would result in a URL similar to this:

```

http://localhost:8080/AuthExtension/index.jsp?auth=dc40570f2e7111fbelaf820a854ca817&timestamp=1189702462936

```

The value of the authentication token would be the hash of a string similar to this:

```

/AuthExtension/index.jsp?auth=&timestamp=1189702462936secret!@#%$%^*(987654321

```

In this case the code in the extension to remove the value of the authentication token from the URL would be more complex.

Related Links

[URL tokens and Workbench extensions](#) on page 50

Oracle Endeca Workbench can pass information to an extension through URL tokens in order to enable the extension to authenticate users, connect to the EAC Central Server, and maintain its state if a user navigates away from the extension and back again during the same session.

Troubleshooting Workbench extensions

This section provides troubleshooting information about Workbench extensions.

If the extension does not have a link in the navigation menu or launch page:

- Stop and restart the Endeca Tools Service. Changes to the XML configuration files for extensions and the navigation menu do not go into effect until the service is restarted.
- Ensure that you have the required Workbench user permission to access the extension.

- Ensure that a menu item for the extension is specified in `ws-mainMenu.xml` and that the `id` attribute matches the `id` of the extension as defined in `ws-extensions.xml`. Defining an extension in `ws-extensions.xml` does not automatically add a link to the navigation menu in Oracle Endeca Workbench.
- If you have no applications defined in Oracle Endeca Workbench, the only links that display in the navigation menu are for the EAC Admin Console and EAC Settings. To enable display of the full Workbench menu, you must first provision an application.

If the link displays in the menu but the extension does not display when you click the link:

- Ensure that the URL for the extension specified in `ws-extensions.xml` is a valid HTTP or HTTPS URL. A Workbench extension must be a Web application running in a Web server.

If an error message displays after updating `ws-extensions.xml`:

There may be a problem with your XML configuration files that prevents Oracle Endeca Workbench from starting up. The error messages in the Oracle Endeca Workbench log can help you identify whether one of the following is the case:

- One or more of the XML configuration files is missing. The following files must be present in `%ENDECA_TOOLS_CONF%\conf` (on Windows) or `$ENDECA_TOOLS_CONF/conf` (on UNIX):
 - `ws-extensions.xml` and its associated schema, `extensions.xsd`
 - `ws-mainMenu.xml` and its associated schema, `mainMenu.xsd`

The files are created in this location when you install Endeca. By default, the `ws-extensions.xml` file defines no extensions. The `ws-mainMenu.xml` file controls the display of the navigation menu and launch page.

If you have deleted one of these files, you can restore the default file by copying it from `%ENDECA_TOOLS_ROOT%\workspace_template\conf` (on Windows) or `$ENDECA_TOOLS_ROOT/workspace_template/conf` (on UNIX).

- One or more of the configuration files contains badly formed or invalid XML.

Ensure that the configuration files contain well-formed XML. In particular, check that any ampersand that is used within an attribute value is specified as the `&` entity.

Use an XML tool to validate any configuration files that you have edited against the associated schema in `%ENDECA_TOOLS_CONF%\conf` (on Windows) and `$ENDECA_TOOLS_CONF/conf` (on UNIX).

Index

A

- adding
 - business users 13, 14
 - administrators 12
- admin user
 - about 12
- audit logs
 - configuring 41
- authentication
 - token-based 52

B

- best practices
 - managing users 12

C

- changing
 - membership 16
- cross-domain policy file 35

D

- deleting
 - users 17
- Dgraph
 - updates 34

E

- Endeca Configuration Repository
 - about 31
 - authentication 31, 33
 - shared secret 33
- extensions
 - authentication 52
 - configuring 48
 - element attributes reference table 48
 - enabling 49
 - introduced 47
 - troubleshooting 53
 - URL token reference table 50
 - URL tokens 50

F

- findGroupPath 26

G

- groupPath 26

L

- launch page
 - configuring 43
- LDAP
 - about disabling 21
 - administrators 20
 - and Java Authentication and Authorization Service (JAAS) 21
 - ASCII characters 26
 - authenticating users 19
 - authentication troubleshooting 28
 - groupPath 26
 - groupTemplate 26
 - integration with Workbench 19
 - LDAP login profile 22
 - permissions 20
 - RFC 2255 26
 - serverInfo 26
 - servers 26
 - SSL integration 19
- LDAP login profile
 - configuration parameters 22
 - configuration parameters reference table 23
 - template reference table 22
- leaf 45

M

- MDEX Engine
 - specifying 34
 - Workbench interaction 34
- menu item
 - elements 43
 - leaf 43
 - node 43
 - nodes for multiple locales 44
- menu nodes
 - for multiple locales 44
- menuitem
 - predefined elements reference table 46
- menutems 45

N

- navigation menu
 - configuring 43

O

- Oracle Endeca Workbench
 - authentication troubleshooting 28
 - extension element attributes 48
 - extensions and URL tokens 50
 - extensions authentication 52
 - extensions introduced 47
 - extensions troubleshooting 53
 - extensions, enabling 49

P

- permissions
 - about 11
- predefined administrator
 - about 12
- property editors
 - image preview 35

S

- SSL
 - enable 37
- system logs
 - about viewing 41
 - configuring 41

T

- timeout 17
- troubleshooting
 - extensions 53
 - LDAP authentication 28

U

- URL tokens 50
- user profiles 20
- userPath 26
- users
 - about 11
 - best practices 12

W

- webstudio_audit.log 41
- webstudio.log 41
- WebStudioMDEX setting 34
- Workbench
 - LDAP login profile 22
 - using over high latency networks 10
- ws-mainMenu.xml 45