An Oracle White Paper
March 2011

Oracle Real User Experience Insight Release 11.1 Quick Start Guide

Introduction

This document aims to get your Oracle Real User Experience Insight (RUEI) installation up and running as quickly as possible. It is not intended to replace the product documentation, but rather to identify the most important issues that need to be considered in order to start data monitoring and reporting.

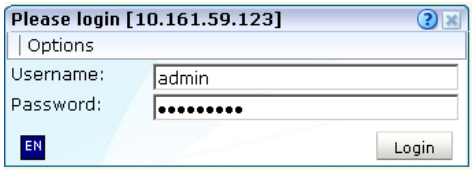
In addition, a number of third-party tools that can assist you in the configuration process are highlighted. Note that the use of these tools is not mandatory, and all featured tools are free of charge.

Running the Initial Setup Wizard

The first time you log on, you will receive a warning that the Web server was unable to verify the identity of the site's certificate. Depending on your security policies, you can either choose to accept this certificate permanently, temporarily for this session, or reject the certificate. Alternatively, you can contact your network or system administrator to acquire the appropriate a certificate from a Certificate Authority (CA). Please note that this security warning is related to your browser and the Web server on which RUEI is running. It is not a security warning of RUEI as a software component itself.

Start the Initial setup wizard by pointing your browser at `https://Reporter/ruei`. The dialog shown in Figure 1 appears.

Figure 1: Logon dialog box.

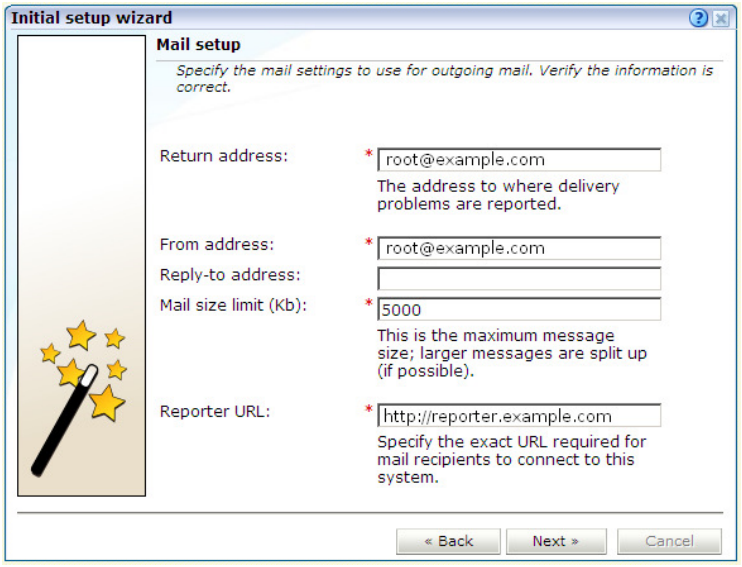


If you experience problems logging on, ensure that any pop-up blocking facility within your browser has been disabled. Specify the `admin` user, and the password defined with the `set-admin-password` script. When ready, click **Login**. The dialog shown in Figure 2 appears.

Figure 2: Initial setup wizard dialog.

When ready, click **Next**. The dialog shown in Figure 3 appears.

Figure 3: Mail setup dialog.



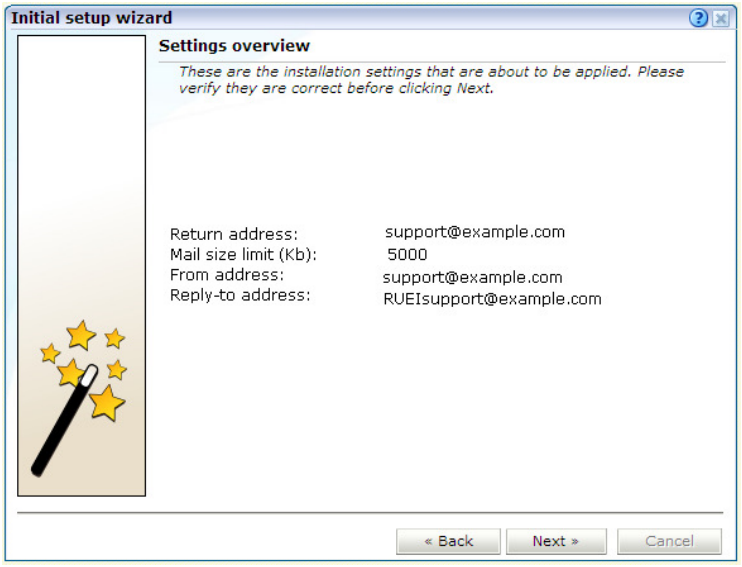
Specify the requested information as described in Table 1. The E-mail information is used to configure RUEI's interface to your internal network, and will be used for reporting problems.

Table 1: Mail Setup Information.

FIELD	DESCRIPTION
Return address	Specifies the E-mail address to which failed or problem E-mails are reported. It is *strongly* recommended that this an address that is regularly checked.
From address	Specifies the address the recipient sees in their mail client.
Reply-to address	Specifies the address that users can click within an E-mail to reply to an E-mail. If this is not specified, the From address setting is used.
Mail size limit	Specifies the maximum message size (in kilobytes) allowed for E-mails. Note that if an E-mail contains reports that exceed this limit, the system will try to split up the reports into individuals E-mails to overcome this limitation. Reports that are too large to be sent individually are not sent, and the user is informed of the problem. The default mail size limit is 5000 Kb.
Reporter URL	Specifies the exact URL required for E-mail recipients to connect to the Reporter system. Typically, this is the same URL used by RUEI users to access the Reporter system.

When you have entered the required information, click **Next**. The dialog shown in Figure 4 appears.

Figure 4: Settings overview dialog.



Check the information specified in the settings overview is correct. You can use **Back** and **Next** to move between dialogs as necessary. When ready, click **Next**. The next dialog indicates how far the system has got in applying your specified settings. Typically, this process takes a maximum of 15 minutes. When finished, click **Finish** to close the dialog.

Initial Configuration Steps for Data Collection

Once the RUEI application has started successfully, you will need to perform an initial configuration in order for RUEI to start monitoring and analyzing traffic. Without this initial configuration, RUEI reports, dashboards, KPIs, and the Data Browser will not be populated with any information. Initial configuration includes the following:

- Specifying the scope of monitoring.
- Specifying the cookie technology.
- Configuring user identification.
- Uploading and activating SSL keys.
- Specifying Web applications, services, and suites.
- Authorizing initial users.
- Securing sensitive data.

Additional Information

Each of these topics are discussed in the following sections. While configuration of these items can be performed in any order, it is strongly recommended that you complete the final section on evaluating your configuration. There is also a helpful webcast on this topic at http://ilearning.oracle.com/ilearn/en/learner/jsp/offering_details_home.jsp?classid=544554206.

In addition, information about the third-party tools featured in this document is available at http://webiv.oraclecorp.com/cgi-bin/webiv//do.pl/Get?WwwID=note:741329.1.

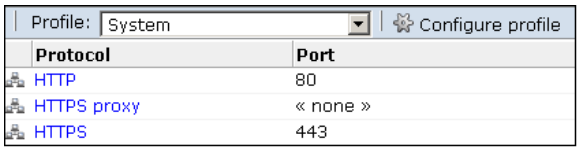
Specifying the Scope of Monitoring

Within RUEI, you control the scope of traffic monitoring by specifying which TCP ports it should monitor. Obviously, no information is available for unmonitored ports.

Do the following:

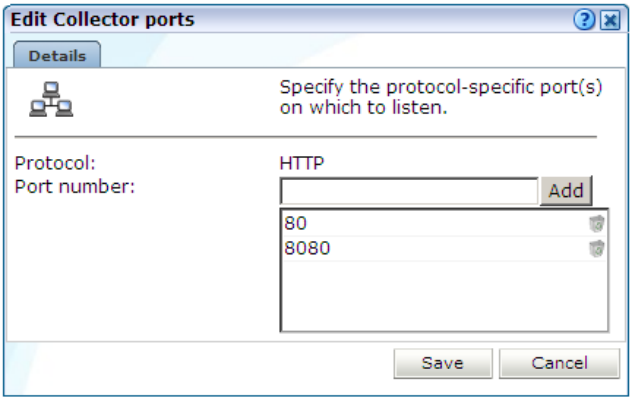
1. Select **Configuration** > **Security** > **Protocols**. The currently monitored ports are listed. An example is shown in Figure 5.

Figure 5: Monitored protocol ports.



2. Use the **Profile** menu to select the required Collector profile. Upon installation, a predefined Collector profile (System) is created. The local Collector instance is assigned to this profile.
3. Click the protocol whose port settings you want to modify. Upon installation, the HTTPS port 443 is defined as the default monitored port. The dialog shown in Figure 6 appears.

Figure 6: Edit collector ports dialog.



4. To add a new port number, enter the required number in the Port number field, and click **Add**. To remove a port from the list, click the **Remove** icon to the right of the port. When ready, click **Save**.

Important: the port numbers specified within each protocol must be mutually exclusive. That is, a port number should only appear in one protocol's list of assigned port numbers.

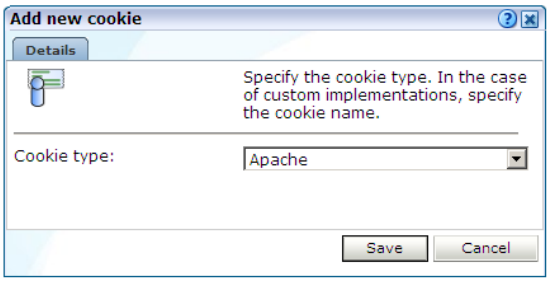
Specifying the Cookie Technology

RUEI needs to know and understand the cookie technology (or technologies) your Web application is using. This will either be a standard technology (such as ASP or ColdFusion), or a custom implementation. Note that a maximum of five cookie technologies can be defined.

If you do not specify a cookie technology, the network IP address and browser combination are used to track the visitor session. This can lead to unreliable session information. In the case of multiple users behind the same proxy server visiting your Web site, they will all be recorded in one single session. Therefore, use of a cookie tracking technology is *strongly* advised. Do the following:

1. Select **Configuration** > **Applications** > **Session tracking**. Click **Add new cookie**. The dialog shown in Figure 7 appears:

Figure 7: Cookie dialog.



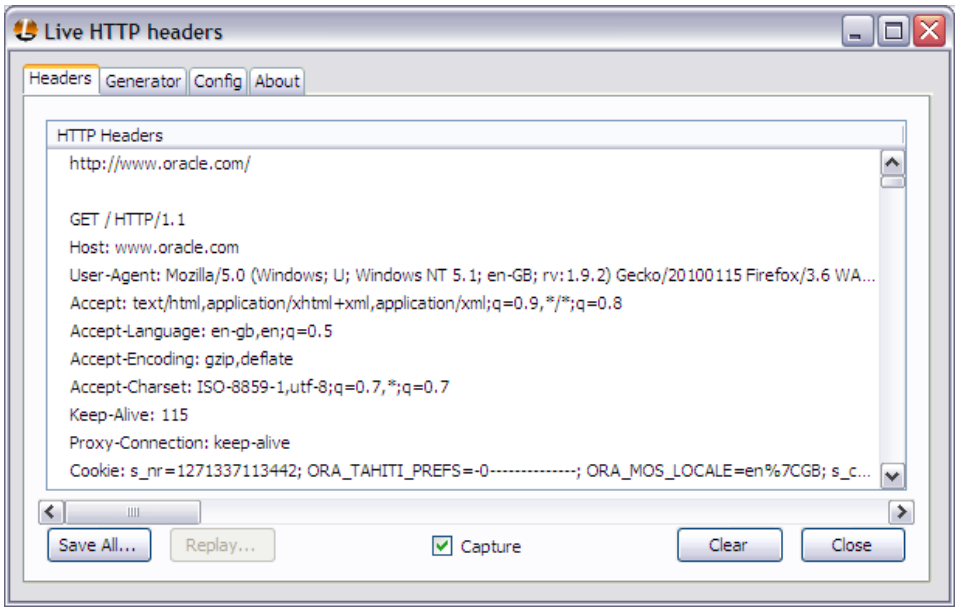
2. Select the required cookie technology. If you are using a non-standard technology, select "custom". If you selected the "custom" option, you are required to specify the name of the cookie used by your Web application. Note that the cookie technology in use should provide a tracking cookie which is unique and non-persistent between sessions. When ready, click **Save**.

Additional Information

To discover the cookie technology in use, together with further details about user name logins (required in the next section), you can use browser plugins such as Live HTTP Headers or FireBug (for Mozilla Firefox), and the Fiddler or HTTPWatch (for Microsoft Internet Explorer).

For example, try to login to the Oracle.com Web site (http://www.oracle.com). The LiveHTTP Headers progress is shown in Figure 8.

Figure 8: User identification.



First, we see the cookie technology being used in the last line. In fact, multiple cookie technologies are being used. However, not all of them will be useful. The cookies or cookie values that are set are semicolon separated. In this example, the 's_nr' seems to be the appropriate technology.

Note: some cookie strings can be very long. In addition, not all of them contain user-related information, or information to uniquely identify a user's session. Try to see if you can find any of the standard cookie technologies within an application that are (by default) supported by RUEI. Refer to

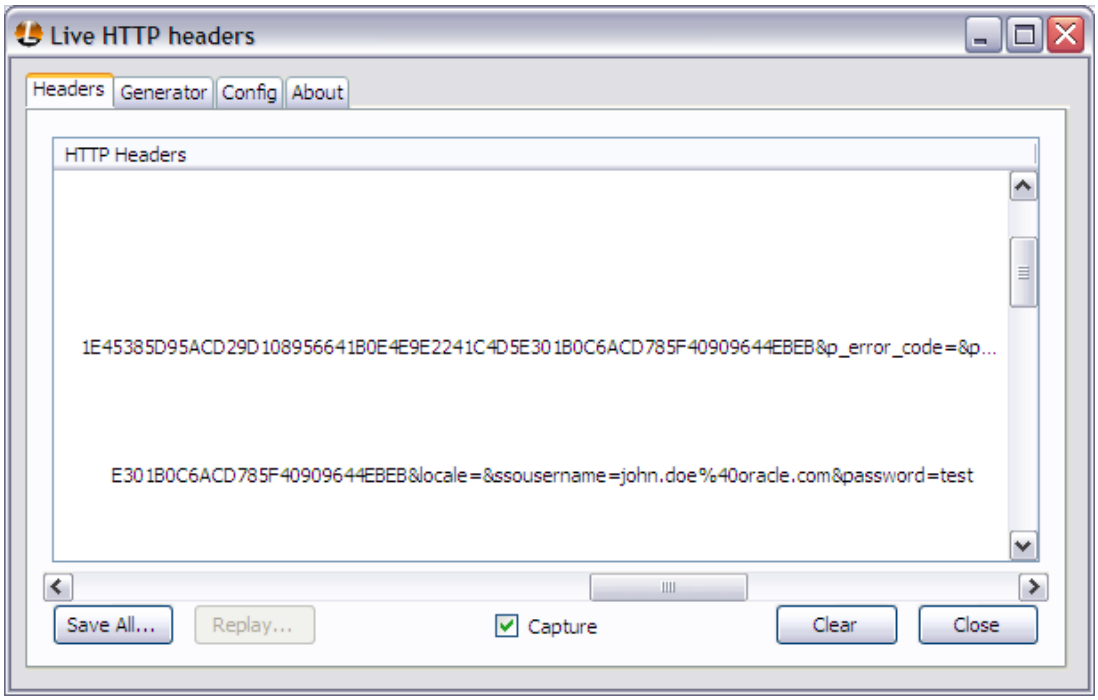
the *Oracle Real User Experience Insight User's Guide* for information about the required conventions for cookie technologies.

Identifying User Names

Besides tracking user sessions, visitors to your Web application can be identified by their user name if they provide these credentials at some point (for example, a login screen). Let's attempt a login by following the Sign In link top middle at the Oracle.com Web site. Note that for this type of investigation, you don't require a working user name and password.

Track your attempt to login again with LiveHTTP Headers as shown in Figure 9.

Figure 9: Logon attempt.



From LiveHTTP headers, we can detect the following:

- Login actions are POSTed against https://login.oracle.com/sso/auth.
- The field that contains the user's name is 'ssousername'.
- The field that contains the user's password is 'password'.

When it is not configured, RUEI will use the SSL client certificate (when available). The common name (CN) portion of it is used. If this is not found, the client ID is reported as Anonymous.

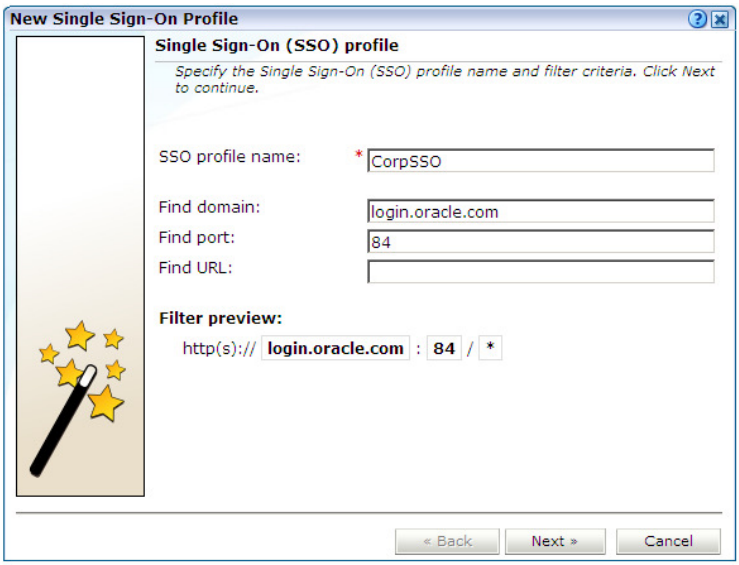
Working with SSO Profiles

In order to facilitate the correct monitoring of SSO-enabled applications, you need to configure the authentication server(s) used within your environment. This is done through the creation of an SSO profile.

To define a SSO profile, do the following:

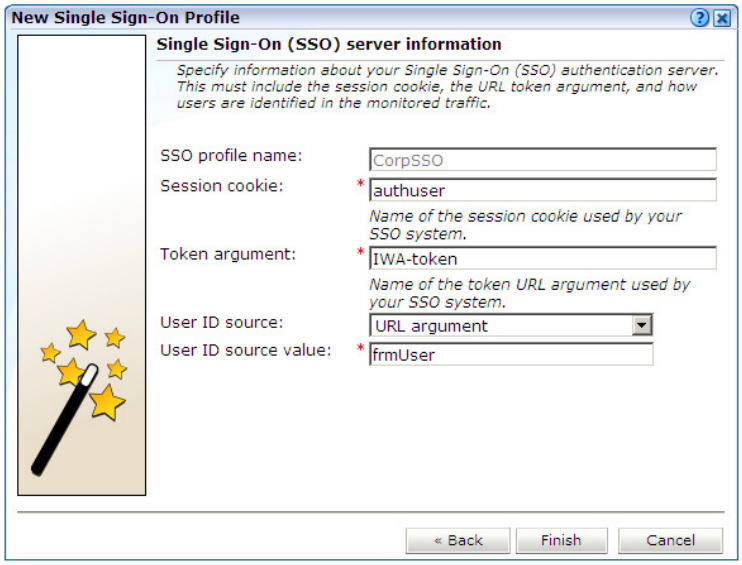
1. Select **Configuration** > **Applications** > **Single Sign-On**, and Click **New SSO profile**. The dialog shown in Figure 10 appears.

Figure 10: New Single Sign-On dialog.



2. Specify a name for the SSO profile. This must be unique across suites, services, applications, and SSO profiles. Note that SSO profiles cannot be renamed later.
3. Use the remaining fields to specify the scope of the SSO profile. This is defined in terms of partial page URLs. Note that you as enter this information, you can see the effect of your definition through the **Filter preview** column. When ready, click **Next**. The dialog shown in Figure 11 appears.

Figure 11: Single Sign-On server information dialog.



4. Use this dialog to specify information about the SSO authentication server you are using. You need to specify the session cookie name, the URL argument which contains the authentication token, and how users are identified in the monitored traffic. Normally, this is defined in terms of a URL argument and value. However, it can also be specified in terms of cookies, request or response headers, or XPath expressions. When ready, click **Finish**.

Note: it is important to understand that SSO profiles and applications, although closely related, are reported as separate entities within RUEI. For this reason, SSO profile and application definitions should be mutually exclusive. That is, each should be based on separate domain and cookies. Otherwise, the monitored traffic is reported as application-related traffic, and potential benefits to enhanced reporting are not realized.

Adding/Loading SSL Keys

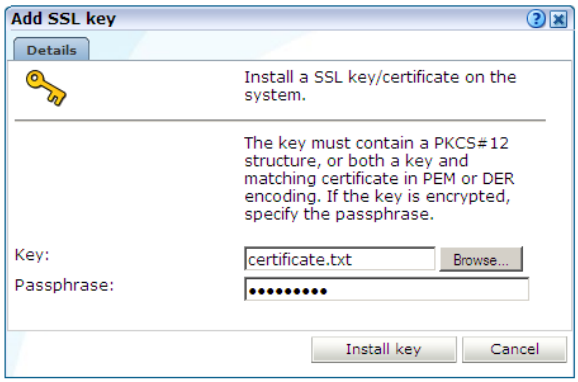
RUEI can be configured to monitor encrypted data (such as HTTPS and SSL). In order to do this, a copy of the Web server's private SSL keys needs to be imported into the system. In the example above, this is clearly required because the login of users is executed against an HTTPS URL. Therefore, in order for RUEI to track and decrypt user names that are posted against this domain, it would need the SSL key for it.

Do the following:

1. Select **Configuration** > **Security** > **SSL keys** > **SSL keys management**. Use the **Profile** menu to select the required Collector profile. This should be "System". Click **Add new key** to define a

new key. Note that existing SSL key definitions cannot be modified. The dialog shown in Figure 12 appears:

Figure 12: Add SSL key dialog.



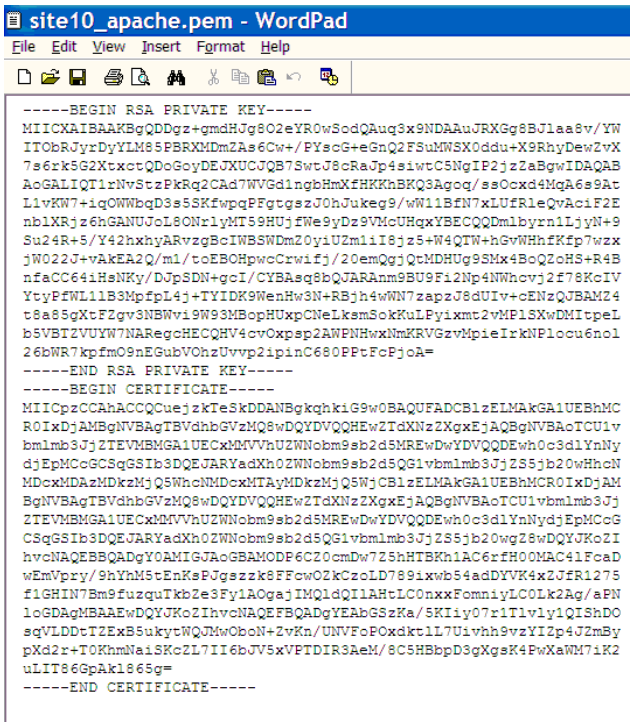
2. Use the **Key** field to specify the file containing the key. You must also specify the passphase used to encrypt the key. The supplied file can be in PEM, DER, or PKCS12 format, and must include the key and matching certificate. The key must be an RSA key. Note that encryption protocols that use 40-bit keys (such as DES_40, RS2_4-0, and RC4_40) are not supported. When ready, click **Install key**.

Additional Information

Sometimes, the Web server's private key and the Verisign private certificate can only be delivered separately. In this case, they need to be combined into one file. This is simple procedure, and should be done immediately on receipt of key and certificate files.

It is recommended that you use Microsoft Wordpad or any other similar DOS/Hex editor. Do *not* use Windows Notepad because it alters the saved file in an unusable way. The file structure should be similar to that shown in Figure 13:

Figure 13: Combined key and certificate file.



That is, the file starts with the RSA Private Key field, followed by the Certificate field.

When using a Linux system, you can create a PKCS#12-compliant file by issuing the following command:

`openssl pkcs12 -export -in` *`certificate.cer`* `-inkey` *`key`*`.key -out` *`pkcs12file.p12`* `-passout pass:`*`yourpassphrase`*

Where:

- *`certicate.cer`* is your CA root certificate file.
- `key.key` is the server's SSL key file.
- *`pkcs12file.p12`* is the output file name for the PKCS#12-encoded file.
- *`yourpassphrase`* is the passphrase you want to use to protect the file from unwanted decryption.

For example, consider the situation where the CA root certificate file name is `ca_mydomainroot.cer`, the Web server's SSL key is `appsrv12.key`, you want the output file to be called `uxssl.p12`, and want to protect this file with the passphrase `thisismysecretpassphrase`. The following command is required:

```
Openssl pkcs12 -export -in ca_mydomainroot.cer -inkey appsrv12.key -out
uxssl.p12 -passout  pass:thisismysecretpassphrase
```

Naming Pages and Web Services

Page identification within RUEI is based on *applications*. Essentially, an application is a collection of Web pages. Each page within an application has an assigned name, and belongs to a group. For example, "MyShop » Contact » About us" refers to the About us page in the Contact group, within the MyShop application.

Each application has a page naming scheme associated with it, which defines its scope. This can be specified in terms of a partial domain name, XPath expressions, URL structure, or a combination of these. It can also be specified through the use of page tagging or the title part of the HTML page.

URL Structure Considerations

If you are planning to use the URL structure as the basis for page identification, it is *strongly* recommended that your Web environment has a clear URL structure. For example, consider the following URL from the Oracle Web site:

```
http://www.oracle.com/en/career/jobs/
```

This clearly identifies the page as the English language version of the page "jobs" in the group "career". Compare this with the following URL:

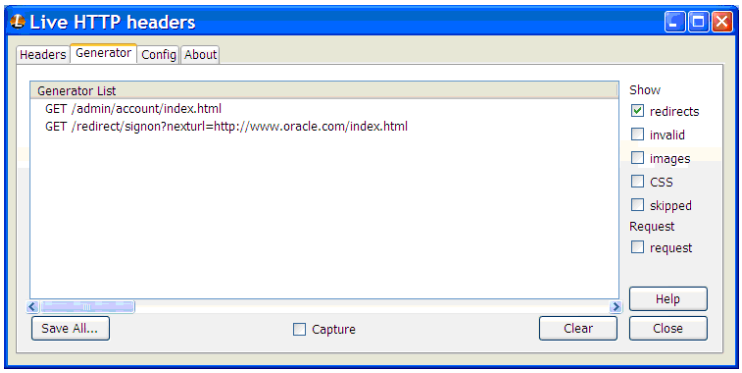
```
http://www.MyShop.com/shop/catalog?Page=WIN&CC=C99&HC=H1&SC=Y6
```

This second URL uses parameters to identify an item in its catalog, which are then resolved by an internal application. However, RUEI cannot identify the page beyond the first part of the URL. Therefore, large numbers of visited pages will be recorded as the same page: shop » catalog.

First contact with the Web Application

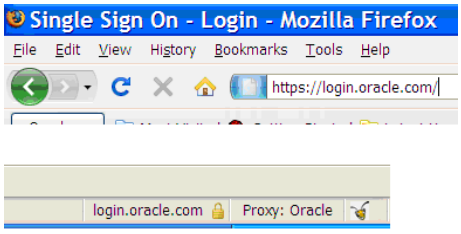
When initially analyzing your Web applications, the Mozilla Firefox browser plug-in is very useful in showing you how contact with the Web application is handled. A typical scenario to a login-secured application is that you are redirected from normal HTTP to an HTTPS encrypted site. Consider the example from Oracle.com shown in Figure 14.

Figure 14: oracle.com login.



After logging onto oracle.com, you are first taken to `/admin/account/index.html`. However, that page automatically takes you to a redirect. In this case, `/redirect/signon` (with some additional URL parameters). On the browser's address bar, it is clear that this redirect leads to `https://login.oracle.com`. Therefore, the official login application resides on the login.oracle.com domain and not the www.oracle.com domain itself. Finally, you can see that the login.oracle.com domain is on active SSL encryption, indicated by the S in HTTPS, and a security lock icon in your browser status bar (shown in Figure 15).

Figure 15: Browser taskbar and status bar.



This leads to the following conclusions:

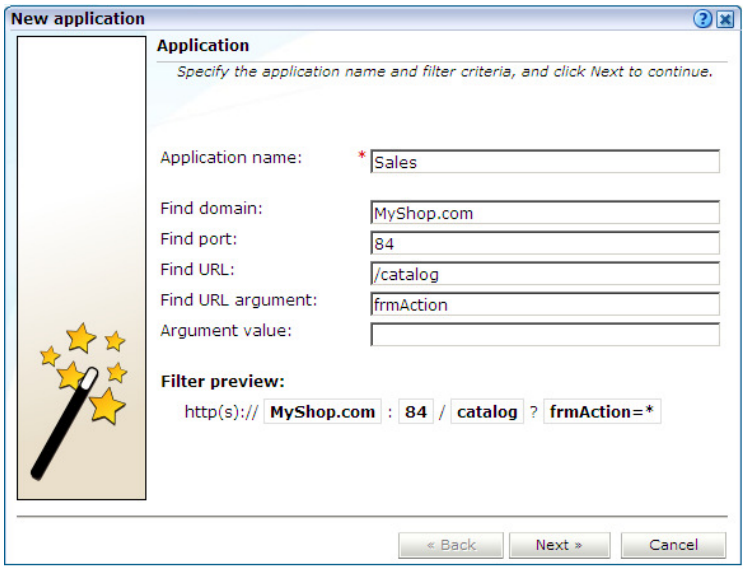
- Two domains (www.oracle.com and login.oracle.com) must be monitored.
- Access to the SSL private server key and SSL public certificate is required to monitor the login application part.

Page name assignment

For each page that the system detects, it uses the available application definitions to assign a name to it. Do the following:

1. Select **Configuration** > **Applications** > **Applications**, and click **New application**. The dialog shown in Figure 16 appears.

Figure 16: New application dialog.



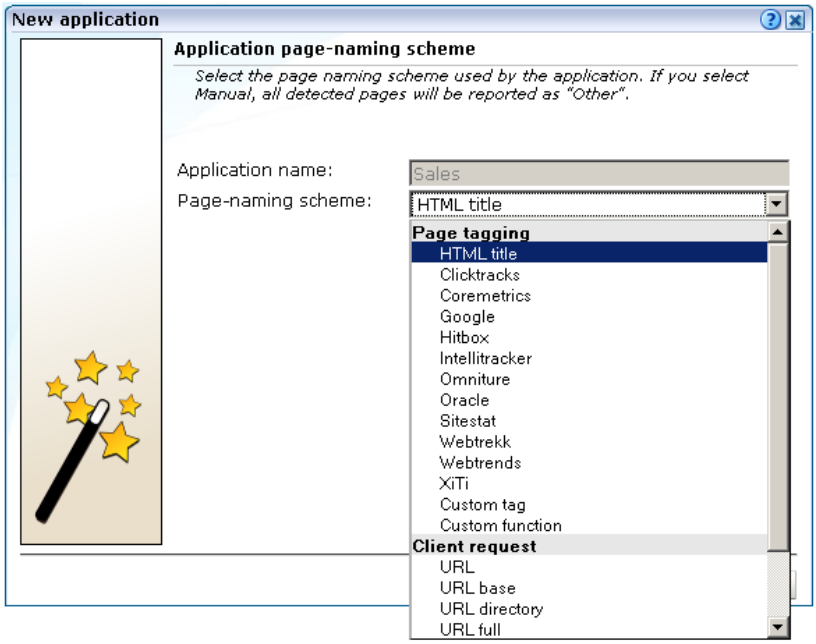
2. Specify a name for the application. This should be unique across suites, services, SSO profiles, and applications. Note that applications cannot be renamed later.

3. Use the remaining fields to specify the scope of the application. This is defined in terms of page URLs. Note that as you enter this information, you can see the effect of your definition through the **Filter preview** column.

   The highest level filter is the domain. It is not possible to specify an application name and leave all the other fields blank. That is, a blank filter. Note that a wildcard character (*) cannot be specified within the **Find Port** field, and only one port number can be specified. If you need to specify additional ports, these should be specified as additional filters after the new application has been created. Be aware that while the use of a wildcard character is supported within certain fields, all other specified characters are interpreted as literals. Finally, it is not possible to specify the wildcard character and no other information for domain and URL argument combinations.

   You can also specify an argument within the URL that must be matched. When ready, click **Next**. The Application page-naming wizard shown in Figure 17 appears.

Figure 17: Application page-naming scheme.



4. This dialog allows you to specify the automatic page-naming scheme used for pages within the application. Only one scheme can be specified per application. The following option groups are available:

   - **Page tagging**: specifies that a either a standard scheme (such as Coremetrics) or a custom scheme is being used. In the case of a custom scheme, you are required to specify the name of the tag. The HTML title option specifies that the text found within the page's <title> tag should be used to identify the page. Note if this is not defined on the page, the <H1>, <H2>, and <H3> heading tags are used.
   - **Client request**: specifies that pages are identified on the basis of their URL structure. The following options specify which portion of the URL is used:

     a. **URL advanced**: page naming is based on advanced URL matching rules. The use of this facility is described below.

     b. **URL-directory**: use only the directory. The various parts of the URL are highlighted in Figure 18.

Figure 18: URL structure.

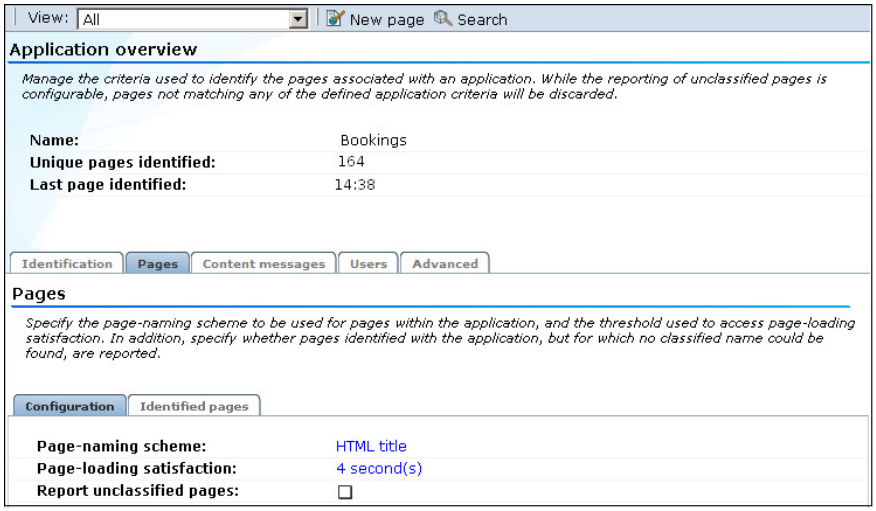
c. **Base-URL**: use the main directory and file name (without the file extension).

d. **Full-URL**: use the main directory, the file name (without the file extension), and the configured arguments. If you select this option, you are prompted for arguments that you want included in the page name. Within the dialog box, multiple arguments should be separated with an ampersand (&) character. For example, if the `frmAction` parameter has been defined, the URL shown in Figure 18 will result in the page name `myshop » shop » NL index frmAction=buy`.

- **Server response**: specifies that pages are identified on the basis of an XPath expression applied to the server response.
- **Manual**: specifies that the application pages will be manually defined rather than through automatic detection. Note that if you select this option, all pages associated with the application that you want monitored must be manually defined. This is the default option.

When ready, click **Finish**. The application definition you have specified is displayed. An example is shown in Figure 19.

Figure 19: Application overview.



5. This overview provides a summary of the defined application. This includes the application's name, the page-naming scheme it uses, the report unclassified pages setting, the page-loading satisfaction assigned to each of the application's associated pages, the source from which the client IP address is fetched, the number of unique pages that have so far been matched to it, and the date of the most recent page identified for it. The **Identification** section summarizes the match criteria currently defined for the application.

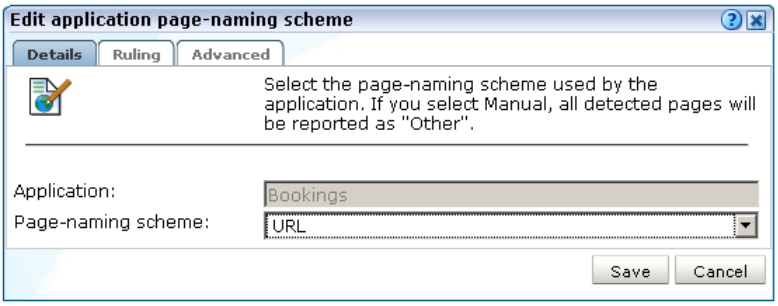
Note that information about any pages that could not be identified using these definitions is discarded and, therefore, not available through reports and the Data browser. You can overcome this by checking the 'Report unclassified pages' check box in the application configuration. However, once activated, you should try to discover why your pages are not reported correctly.

Using Advanced URL Matching Rules

Note that the Advanced URL matching rules facility should only be used by those who have a sound understanding of the Web application architecture and Web technologies generally. To specify the use of advanced URL matching rules for a selected application, do the following:

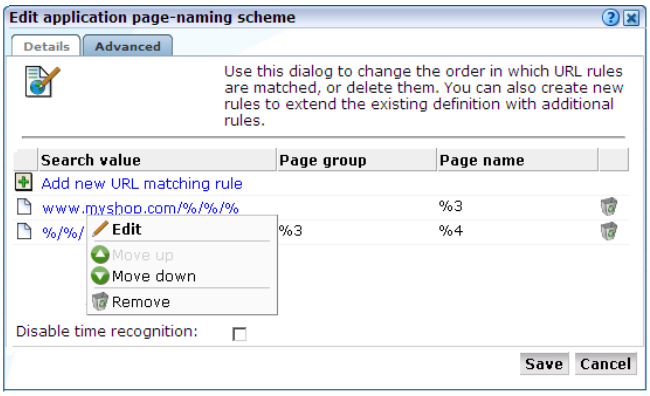
1. After you have initially defined your application (as described earlier), click the page-naming scheme setting shown in Figure 19. The dialog shown in Figure 20 appears.

Figure 20: Edit application page-naming scheme dialog.



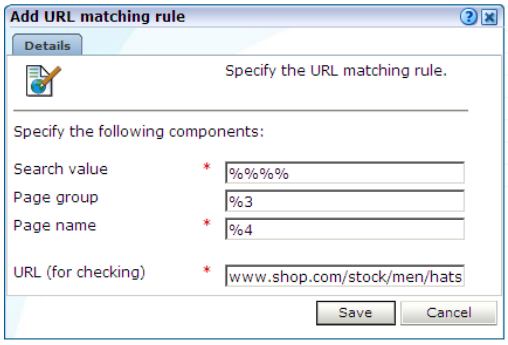
2. Use this dialog to either change the specified page-naming scheme, or click the **Ruling** tab to specify the URL matching rules, and the order in which they should be evaluated. The dialog shown in Figure 21 appears.

Figure 21: Advanced URL matching rules.



3. Click the **Add new URL matching rule** item to define new matching rules. The dialog shown in Figure 22 appears. Use this dialog to define new rules or delete existing ones. You can also right click a rule and use the context menu to modify the order in which they are applied, as well as edit and delete them. When ready, click **Save**.

Figure 22: Add URL matching rule dialog.



Each URL matching rule is expressed in terms of the following components:

- Search value: specifies the structure of the expected URL. Essentially, it provides a template for interpreting the received URL.
- Page group: specifies how the page group is identified from the received URL. Note if this is not specified, the page group is assigned the page name.
- Page name: specifies how the page name is identified from the received URL.
- URL (for checking): specifies a definition of the URL that should be matched. Typically, this is expressed in terms of required parameters, and the sequences that should comprise them.

If you do not specify anything in the advanced rules, the page is discarded and not reported. The rules are matched in the order specified for them. In addition to the use of parameters, the elements shown in Table 2 can also be used in URL matching rules.

Table 2: Advanced search constructions.

USAGE	DESCRIPTION
%	Match zero or more characters and fill one placeholder. Allowed placeholders are %1 - %9.
%[!...]	Find one value corresponding to any of the supplied name(s) in the URL argument, and fill one each for the original and matched placeholders.

%[&...]	Find all values corresponding to the supplied name(s) in the URL argument, and fill one parameter placeholder for the original and specified number of placeholders.
%[|...]	Find zero or more values corresponding to the supplied name(s) in the URL argument, and fill one placeholder for the original and specified number of placeholders.
%[c#]	Find the specified number of characters.
%[d]	Find directory path of the URL, and fill one placeholder.
%[f]	Find file name path of the URL without the file extension, and fill one placeholder.
%[h]	Find domain part of URL, and fill three placeholders (for example, a.b.name.co.uk would be matched as %1=a.b, %2=name, and %3=co.uk).
%[t...]	Match until one of the following characters is matched and fill one placeholder.
%[t^...]	Match until a character is found that does not match the specified list of characters.

Note the special characters specified in Table 1 must be preceded with a backslash if they should be interpreted literally. For example, \% specifies a literal % character, rather than a parameter. Also, be aware that a maximum of nine placeholders can be specified.

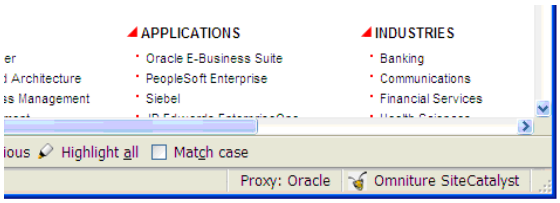
Important: because of the complex nature of URL matching rules, it is recommended this facility is only used by users with a sound understanding of URL structures. In addition, the selected application's underlying URL structure should be clearly understood.

Identifying the Page Naming Scheme

This section highlights the use of Web Analytics Solution Profiler (WASP). It is a Mozilla Firefox browser plug-in that helps you to immediately identify the Web analytics solution that a Web application is using.

In most public Web sites, so-called page labels are used to uniquely label each page. WASP will easily detect which labeling technology is being and, almost certainly, RUEI supports it. WASP is available via an icon on your browser's status bar. Figure 23 shows an example from the oracle.com Web site that contains the Omniture SiteCatalyst page labels.

Figure 23: Omniture SiteCatalyst page labels.



Note that page label identification can also performed via the Web page's source code. An example is shown in Figure 24:

Figure 24: Web page source code.

Usually, page labels are inserted just before the BODY opening or closing tag of a Web site. For quality reporting, and direct alignment with existing reporting, it is recommended to use any existing page labeling technology – if in place.

Manually Defined Pages

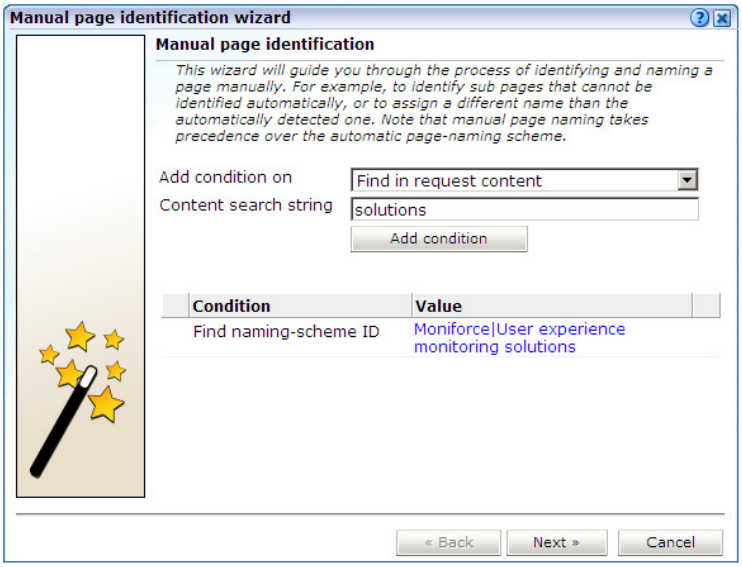
In addition to automatic detection, application pages can also be defined manually. This is particularly useful in the case of an inconsistent URL structure, or where identified pages contain sub pages, or you want to assign a different name to the one assigned automatically to it by the application. Note that these manually defined pages take precedence over pages identified automatically through application definitions.

To manually identify pages, you can either define the new page from scratch, or use an existing page (automatically detected or manually defined) as the basis for the new page.

To define a page, do the following:

1. To define the page from scratch, select the required application in the application overview, and click the **New page** button. To use an existing page as a basis for the new page, select the required application page, and click the **New page (based on current)** button. In either case, the dialog shown in Figure 25 appears.

Figure 25: Manual page naming wizard.

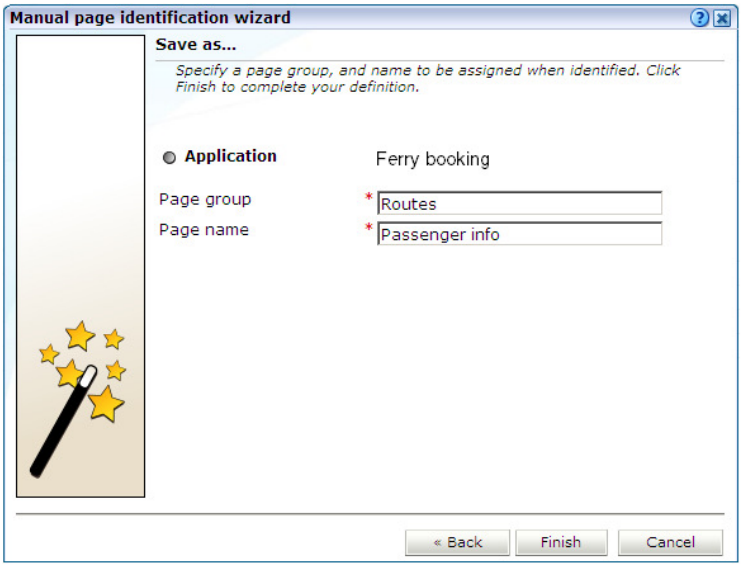


2. Use this dialog to specify the conditions that must be met for the page to receive the assigned name. These conditions can be defined in terms of the page's partial or exact URL, content, domain, or arguments. An XPath expression can also be specified. Click **Add condition** for each required condition.

   Note that when specifying an exact URL (for example, http://www.oracle.com/contact.html) the domain and remaining URL structure are automatically assigned to the page conditions. For example, Find in domain (oracle.com) and Find exact URL (/contact.html).

   As you specify additional conditions, these are shown in the dialog. *All* specified conditions must be met for a match to be made. Note that conditions shown in blue can be removed by clicking them, while conditions shown in black cannot be removed. You must specify at least one condition for page identification. When ready, click **Next**. The dialog shown in Figure 26 appears.

Figure 26: Save as dialog.



3. Use this dialog to specify a group and name for the page. When ready, click **Finish**.

Working with Suites

If your monitored environment includes Siebel, E-Business Suite (EBS), JD Edwards, or PeopleSoft-based applications, it is *strongly* recommended that you make use of the relevant RUEI accelerator package. It not only saves you time in defining your applications, and makes applications within suites more compatible, but also ensures that these architectures are monitored correctly.

Important: Suite functionality is, by default, disabled. Packages are made available to enable it, and provide support for specific Oracle architectures. For information about package availability, please contact Customer Support or visit the Web site (http://www.oracle.com/enterprise_manager/user-experience-management.html.

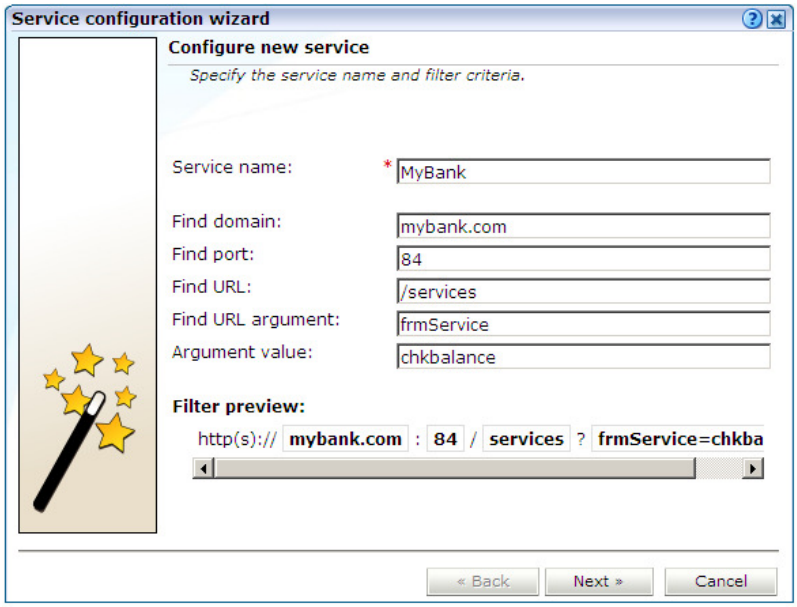
Working with Web Services

A Web service implements a clearly defined business function that operates independently of the state of any other service. Services are often loosely coupled - a service does not need to know the technical details of another service in order to work with it - and all interaction takes place through the interfaces. Using this technology, a service provider simply exposes a service on the Web, publishes the interface and service naming specifications, and waits for a connection.

To define a Web service, do the following:

1. Select **Configuration** > **Services**. Click **New services**. The dialog shown in Figure 27 appears.

Figure 27: Service configuration wizard.



2. Specify a name for the service. This is the name that will be used for the defined service within reports and the Data browser. The name must be unique across services, SSO profiles, suites, and applications. Note that services cannot be renamed later.

3. Use the remaining fields to specify the scope of the service. This is defined in terms of partial service URLs. Note that as you enter this information, you can see the effect of your definition through the **Filter preview** column.

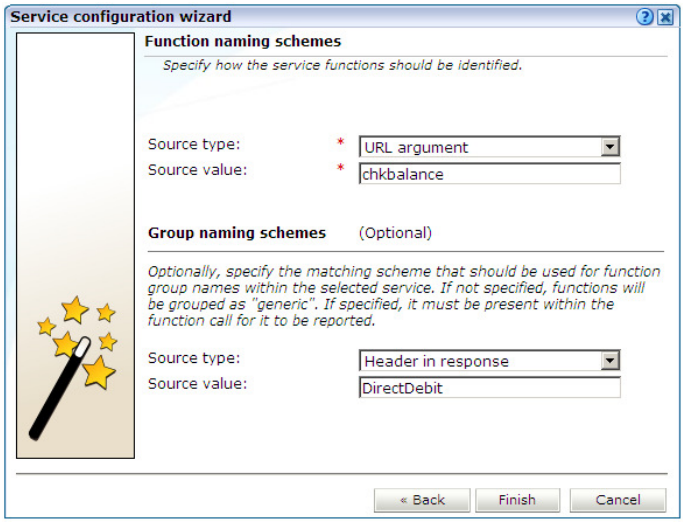
   The highest level filter is the domain. You can specify a partial URL instead of, or to refine, a domain. It is not possible to specify a service name and leave all the other fields blank. Note that a wildcard character (*) cannot specified within the **Find Port** field, and network traffic arriving on a non-standard port (that is, other than ports 80/443), is not associated with the service unless the port number is explicitly stated. You can only specify one port number within the **Find Port** field. If you want to specify additional ports, these should be specified as additional filters after the new service has been created.

   Important: it is recommended that filter definitions should be mutually exclusive across services, SSO profiles, applications, and suites. For example, do not define a service filtered on the domain us.oracle.com and then another service, suite, or application filtered on us.oracle.com/application_servlet. The use of non-mutually exclusive filter definitions can lead to unpredictable results.

   You can also specify an argument within the partial URL that must be matched. Note that if you

use this facility, both the argument and argument name must be complete in order for them to be matched to page URLs. That is, partial matching is not supported. When ready, click **Next**. The dialog shown in Figure 28 appears.

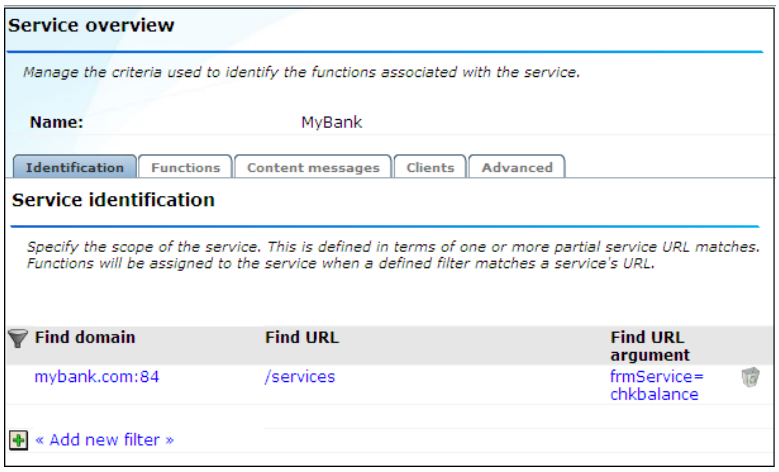
Figure 28: Function naming scheme dialog.



4. Use this dialog to specify how the service should be identified and reported. It is important to understand that while applications have the structure application » group » page, services have the structure service name » function group » function name. Note that functions that do not belong to a defined group are regarded as belonging to the default group "generic".

   When ready, click **Finish**. The service definition you have specified is displayed. An example is shown in Figure 29.

Figure 29: Service overview.



Securing Sensitive Data

Several directories on the Reporter system may hold sensitive data which was captured during monitoring. This is especially true if the Replay Viewer has been enabled. It is *strongly* recommended that you encrypt this data. You can select full disk encryption during the disk partitioning phase of the operating installation procedure. In this case, a password is required. This is fully explained in the *Oracle Real User Experience Insight Installation Guide*.

By default, the Replay Viewer is disabled. However, if enabled, it shows "raw" collected data. That is, no defined blinding filters are applied. Therefore, any sensitive information contained within the content becomes visible in the Replay viewer. Therefore, it is *strongly* recommended that you carefully review your information security requirements before enabling Replay view functionality.

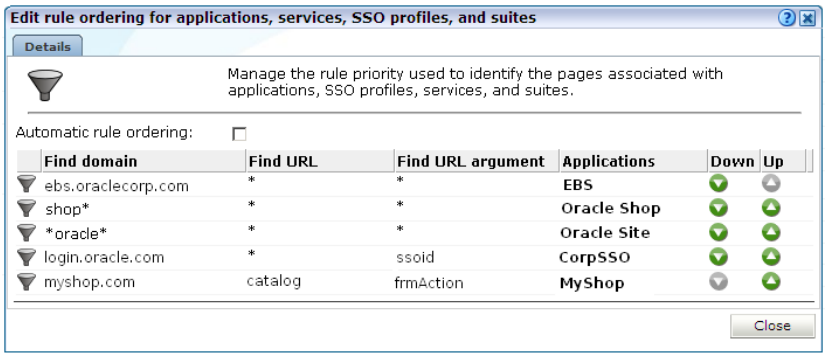
Controlling Rule Ordering

By default, the order in which application, SSO profile, suite, and service filters are matched within RUEI is determined by the level of detail specified in the definition. That is, the definitions with the most information specified for them are applied first. However, sometimes you may want to modify the order in which filters are applied.

To use the rule ordering facility, do the following:

1. Click the **Configuration** tab, select the **Configuration** menu option, and then the option **Edit ruling orders**. A dialog similar to the one shown in Figure 30 appears.

Figure 30: Edit rule ordering dialog.



2. Use the **Automatic rule ordering** check box to specify whether the rule ordering is automatically derived from the currently defined applications, SSO profiles, suites, and services. As explained earlier, by default, the definitions with the most information specified for them are applied first. This is check box is automatically unchecked if you use the **Up** and **Down** controls to specify the order in which the rules should be applied. If you re-check it, the filter ordering is automatically reset to the default. Note any changes you make are immediately put into effect. When ready, click **Close**.

Important: be aware that if you modify the default rule ordering, and then define a new application, SSO profile, suite, or service, its associated filter is immediately placed at the bottom of the current rule ordering. Therefore, you should always review the rule ordering after the creation of new filters.

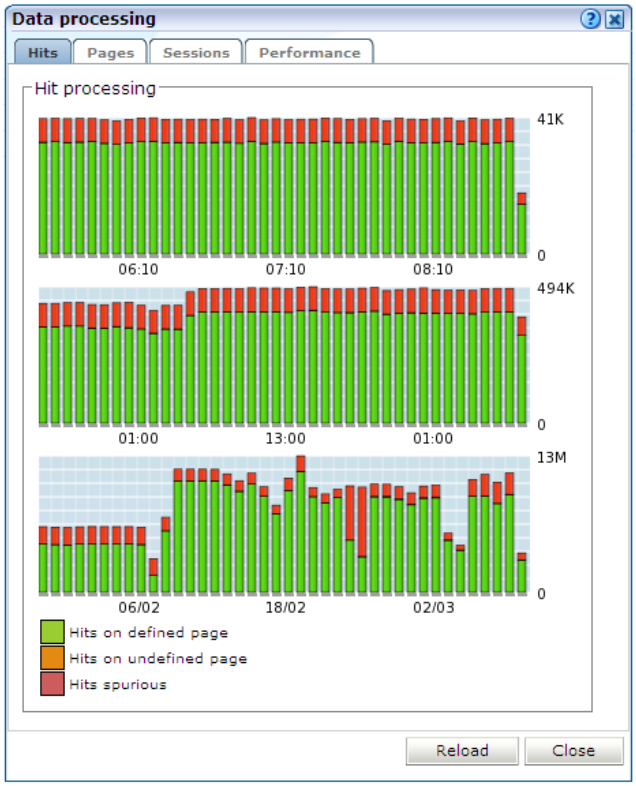
Verifying and Evaluating Your Configuration

To ensure the quality and quantity of data being collected and analyzed by your RUEI system, it is *strongly* advised that you verify the system's configuration using some core metrics. These are described in the following sections.

Viewing a traffic summary

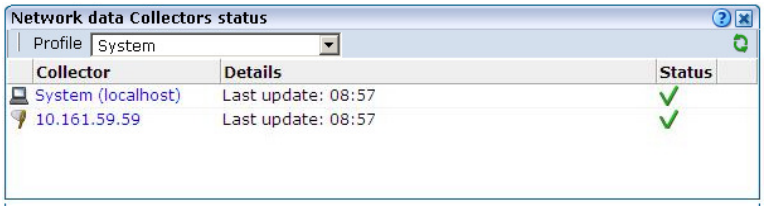
You can open an overview of the monitored network traffic by selecting **System** > **Status** > **Data processing**. This provides you with immediate information about hits, pages, and session processing, as well as the system load. An example is shown in Figure 31.

Figure 31: Data processing dialog.



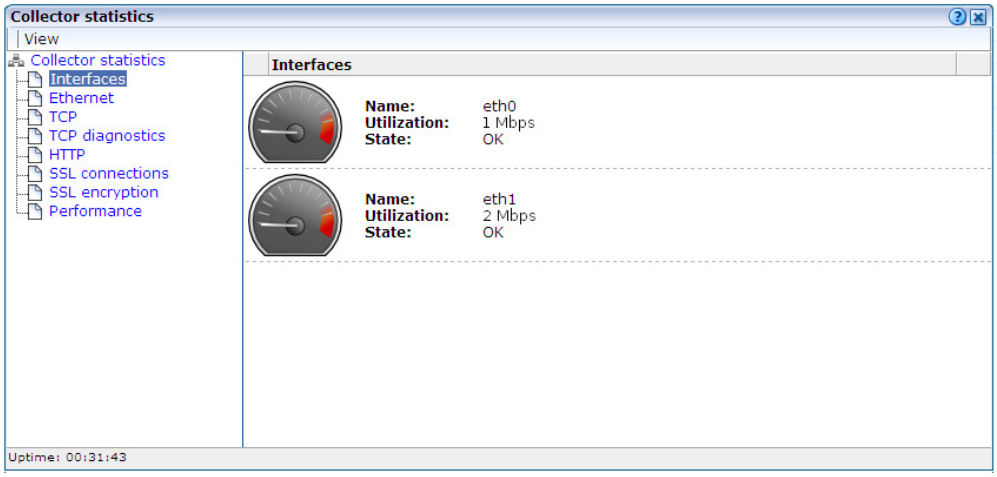
You can view the status of the Collectors attached to the system by selecting **System** > **Status** > **Collector status**. It opens the Network data Collectors window. An example is shown in Figure 32.

Figure 32: Network data collectors.



For each Collector, select **View statistics** from its context menu to view a detailed report of the traffic monitored by the Collector. An example is shown in Figure 33.

Figure 33: Collector statistics window.



The information shown in this window refers to the traffic monitored since midnight for the selected Collector, or the counters were reset. The Uptime field in the bottom left-hand corner of the window shows the time the Collector has been running. The uptime is reset when the Collector is restarted to update its configuration. You can reset all of counters shown in the window by selecting **Reset counters** from the **View** menu. Note that the counters will be reset the next time a network packet is detected. Hence, on an installation with no network traffic, the counters will never be reset. The display is automatically refreshed every two seconds.

Authorizing Initial Users

In order for users to start working with RUEI, you will need to authorize the required users. Only one user, the Administrator, is available after installation. All other required users must be created and assigned the necessary roles and access permissions through the Reporter GUI.

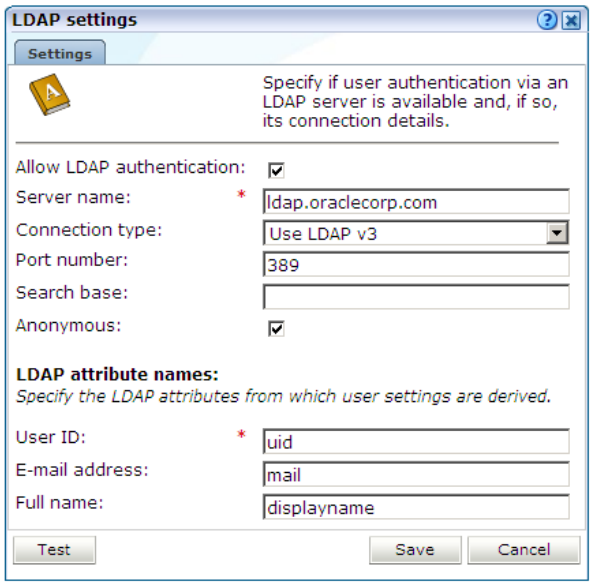
Configuring LDAP Server User Authentication

In order to provide enhanced security, RUEI can be configured to enable user authentication via an LDAP server, rather than through the settings held locally on your RUEI installation. If an LDAP server connection has been configured, you can specify the authentication method to be used for each defined user. Note because the Administrator user is predefined, and their password is set during initial configuration (see the *Oracle Real User Experience Insight Installation Guide*), only local authentication is available for this user.

If you plan to use LDAP authentication, it is recommended that you define your LDAP connection *before* the creation of user accounts. This is in order to prevent having to modify previously specified user settings. To enable LDAP server authentication, do the following:

1. Select **System >User Management**, and then click **Configure LDAP connection**. The dialog shown in Figure 34 appears.

Figure 34: LDAP settings dialog.



2. Use the **Allow LDAP authentication** check box to specify whether an LDAP server is available for user authentication. The default is unchecked (disabled).
3. Use the **Server name** field to specify the host name or IP address of the LDAP server to be used. Note that protocol information (such as `LDAP://`) should be omitted from the server name.
4. Use the **Connection type** menu to specify the LDAP version and connection method. The default is V2 (non-secure).
5. Use the **Port number** field to specify the port to which the LDAP server is listening. If necessary, discuss this with your System Administrator. The default port is 389 or 636 (for SSL encryption).
6. Use the **Search base** field to specify the location in the directory structure within which the user ID needs to be unique. This must be a valid DN. For performance reasons, this should be as specific as possible. The default is the root of the directory tree.
7. Use the **Anonymous** check box to specify if the LDAP server lookup should be performed using an anonymous user. If unchecked, then a valid Distinguished Name (DN) must be specified, and the password for that user is requested when a new user is created. The default is to use an anonymous lookup.

8. Use the **User ID**, **Email address**, and **Full name** fields to specify the attributes that should be used to extract user settings from the LDAP server. The defaults are based on standard LDAP functionality. If necessary, you should discuss these attributes with your LDAP administrator.

9. Optionally, you can click **Test** to verify whether a working connection to the LDAP server can be made. When ready, click **Save**.

Configuring Oracle SSO User Authentication

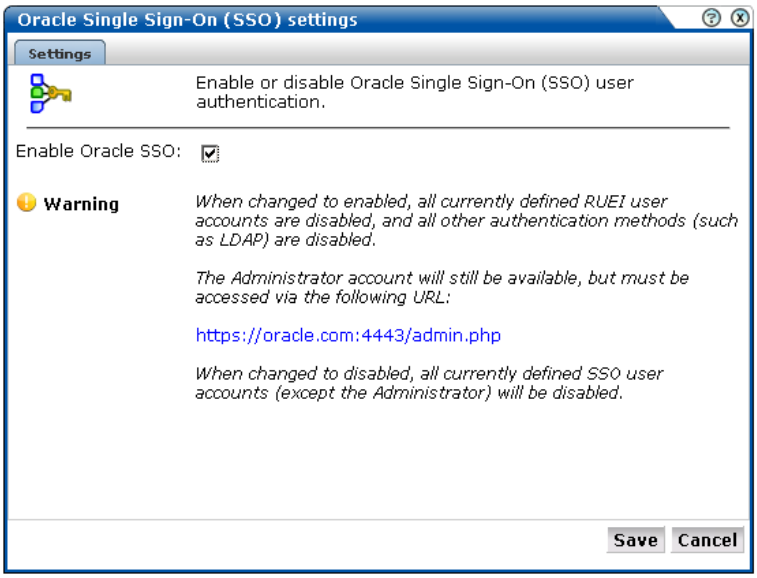
RUEI can be configured to enable user authentication via an Oracle Single Sign-On (SSO) server, rather than through the use of an LDAP server or the settings held locally on your RUEI installation. When enabled, RUEI users (other than the `admin` user) are automatically re-directed to the Oracle SSO logon page. They then logon to RUEI through this page, rather than the RUEI login dialog (shown in Figure 1).

The facility for user authentication via an Oracle SSO server is not available until the RUEI application has been registered with the Oracle SSO server. The procedure to install and configure the Oracle SSO server for RUEI user authentication is described in the *Oracle Real User Experience Insight Installation Guide*.

To activate the SSO server, do the following:

1. Select **System** > **User management** > **Configure SSO connection**. Note that if an Oracle SSO server connection has already been activated, the option is indicated as **Modify SSO connection**. The dialog shown in Figure 35 appears.

Figure 35: Oracle SSO settings dialog.



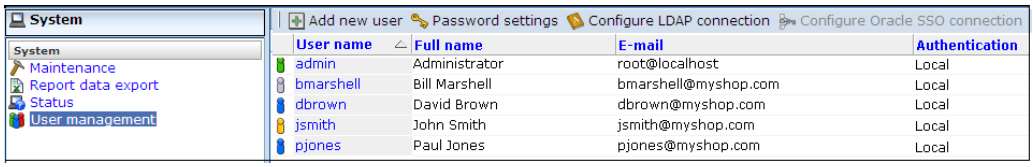
2. Use the **Enable/Disable Oracle SSO** check box to specify whether an SSO server is available for user authentication. The default is unchecked (disabled). When ready, click **Save**.
3. After enabling or disabling the Oracle SSO server, it is recommended that you logout and logon again to RUEI. This is to ensure that your RUEI installation reflects the change you have made.

Creating Required Users

Create additional users, do the following:

1. Select **System** > **User management**. The screen shown in Figure 36 appears.

Figure 36: User management.

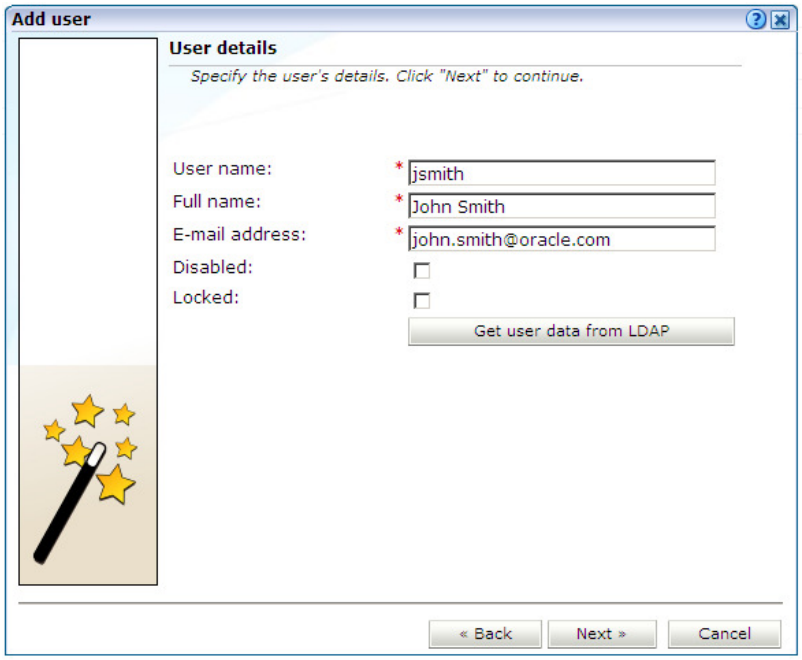


2. Click **Add new user**. If an LDAP server connection has been configured, the dialog box shown in Figure 37 appears. Otherwise, the dialog shown in Figure 38 appears, and you should continue from step 3.

Figure 37: Add new user.

3. Use the radio buttons shown in Figure 37 to specify whether the creation of the new user account, and it's associated user settings, should be authenticated against the settings held in the RUEI installation (this is the default), or against a configured LDAP server. When ready, click **Next**. A dialog similar to the one shown in Figure 38 appears.

Figure 38: User details dialog.

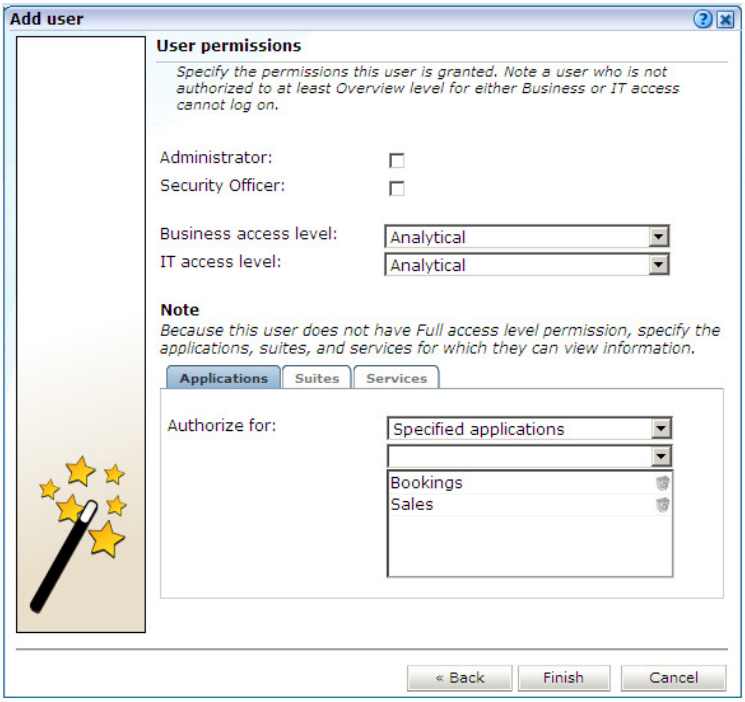


4. Use the dialog shown in Figure 38 to specify the following information for the new user:

    - The user name by which the user will be known within your RUEI installation. This must be a unique name.
    - The user's full name.
    - The user's e-mail address. This is the address to which reports and e-mail alerts will be sent. Ensure it is correct.
    - If the user will be authenticated against the settings held in the RUEI installation, you are required to specify and confirm a password for the new user. Note the new password must be changed within seven days or the user is locked out.
    - Optionally, use the **Disabled** check box to disable the user at this time. You are free to enable them later.

    If you selected user authentication against a configured LDAP server in Figure 37, you can click the **Get user data from LDAP** button to retrieve the user's settings from the configured LDAP server. When ready, click **Next** to continue. The dialog shown in Figure 39 appears.

Figure 39: User permissions dialog.



5. Use the check boxes and radio buttons to specify the permissions to be assigned to the new user. If the new user is assigned less than Full access level permission, you must use the **Authorize for** menu to specify the specific applications, suites, and services about which the user is authorized to view information. Click **Finish** to create the user definition. You are returned to the user list shown in Figure 36.

Oracle is committed to developing practices and products that help protect the environment

Oracle Real User Experience Insight Release 11.1 Quick Start Guide
March 2011
Author: Paul Coghlan, Jan van Tiggelen

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com