Oracle® Fusion Middleware

Using Oracle WebCenter Portal Performance Pack

12*c* (12.2.1.3.0)
F10128-01
October 2018

Oracle Fusion Middleware Using Oracle WebCenter Portal Performance Pack, 12*c* (12.2.1.3.0)

F10128-01

Copyright © 2017, 2018, Oracle and/or its affiliates. All rights reserved.

Primary Author: Oracle Corporation

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, then 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.

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

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience iv
Documentation Accessibility iv
Related Documents iv
Conventions iv

1 Getting Started with Oracle WebCenter Portal Performance Pack

About WebCenter Portal Performance Pack 1-1
Before You Begin with WebCenter Portal Performance Pack 1-2
How to Install WebCenter Portal Performance Pack 1-2
How to Undeploy WebCenter Portal Performance Pack 1-3
Quick Tour of the Landing Page 1-3

2 Working with Events

What can I do with the Events List? 2-1
How do I locate hot paths in my code? 2-2

3 Working with Method Statistics and Dump Threads

How do I find methods for optimization? 3-1
How do I locate cause of serious performance issues? 3-2

4 Configuring Instrumentation

How do I change configuration settings to suit my system needs? 4-1
How do I add methods to the instrumentation configuration? 4-3

Preface

This guide explains how to use Oracle WebCenter Portal Performance Pack to diagnose Oracle WebCenter Portal performance. It provides instructions for tasks such as using the event listing, finding methods for performance optimization, and changing configuration settings.

Topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

This guide is intended for administrators and developers who want to identify slow requests and drill down to specific hot paths and methods to diagnose performance issues.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

Documentation for Oracle WebCenter Portal is available in the Oracle Fusion Middleware library on the Oracle Help Center.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Getting Started with Oracle WebCenter Portal Performance Pack

This section describes how to get started with the Oracle WebCenter Portal Performance Pack tool.

Topics

- About WebCenter Portal Performance Pack
- Before You Begin with WebCenter Portal Performance Pack
- How to Install WebCenter Portal Performance Pack
- How to Undeploy WebCenter Portal Performance Pack
- Quick Tour of the Landing Page

> **Important:**
>
> Oracle WebCenter Portal Performance Pack is entitled as part of Management Pack for WebCenter license. Oracle WebCenter Portal Performance Pack is available as an add-on as part of Oracle WebCenter Portal 11g and 12c.

About WebCenter Portal Performance Pack

WebCenter Portal Performance Pack is a performance diagnostics tool that can be integrated seamlessly into the development phase where detailed analysis is often required.

Using WebCenter Portal Performance Pack you can quickly identify and address critical performance bottlenecks in your application. We've pre-configured WebCenter Portal Performance Pack to instrument the main concerning areas for Oracle WebCenter Portal 11g or 12c; however, you can easily modify the configuration to fit any J2EE web application without restarting the running system.

WebCenter Portal Performance Pack:

- Makes it easy for you to identify slow requests
- Lets you drill down to specific hot paths and methods.
- Doesn't overwhelm you with too much data; instead provides you with only the most critical information.
- Aggregates data in different ways, which makes it easy for you to spot problem areas.
- Provides hints for optimization opportunity

Before You Begin with WebCenter Portal Performance Pack

WebCenter Portal Performance Pack has no prerequisites. However, it's helpful to be familiar with Java technology, The Hypertext Transfer Protocol (HTTP), WebLogic Server, as well as Oracle WebCenter Portal or any webapp running in the target JVM.

Before you begin using WebCenter Portal Performance Pack, it's helpful if you're familiar with the following technologies:

- Java, the foundation for virtually every type of networked application and the global standard for developing and delivering embedded and mobile applications, games, Web-based content, and enterprise software.

  www.java.com
- HTTP, the underlying protocol used by the World Wide Web to define how messages are formatted and transmitted.

  en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- Oracle WebLogic Server, provides a standard set of APIs for creating distributed Java applications that can access a wide variety of services, such as databases, messaging services, and connections to external enterprise systems. End-user clients access these applications using Web browser clients or Java clients.

  Understanding Oracle WebLogic Server
- Oracle WebCenter Portal, a web platform that allows you to create intranets, extranets, composite applications, and self-service portals.

  Introduction to WebCenter Portal in *Using Oracle WebCenter Portal*

How to Install WebCenter Portal Performance Pack

WebCenter Portal Performance Pack is made available as part of the WebCenter installation.

To install WebCenter Portal Performance Pack:

1. Go to `<INSTALL_HOME>/wcportal/webcenter/archives/perfSense.zip`
2. Install WebCenter Portal Performance Pack:
   a. Unzip `perfSense.zip` into a directory on the server where your target JVM is located.
   b. Run `./deploy.sh <WLS_SERVER_NAME>` to install the perfSense agent and webapp.

      For example, `./deploy.sh WC_PORTAL` .
      Note that you'll be prompted to enter the admin server's Administrator **Username** and **Password** to install webapp.
3. Access WebCenter Portal Performance Pack using this URL: `http://<host>:<port>/perfsense`
4. Sign in using a **Username** and **Password** that belongs to the Administrators role defined in WebLogic server.

You're directed to the landing page, which provides access to all the resources required to monitor and diagnose performance issues related to the selected Oracle WebCenter Portal instance.

Notes:

- The `deploy.sh` command installs two components:
  - **perfSense agent:** Instruments JVM
  - **perfSense webapp:** Displays the performance data for the instrumented JVM
- If you restart the target JVM, you must reinstall the perfSense agent.

  Run `./deploy.sh –a <WLS_SERVER_NAME>`
- If you restart the target JVM, the perfSense webapp remains installed. However, if you need to reinstall the webapp:

  Run `./deploy.sh –w <WLS_SERVER_NAME>`
- You can run `./deploy.sh` to see the script usage.

How to Undeploy WebCenter Portal Performance Pack

Undeploy is a command-line tool that you can use to clean up the WebCenter Portal Performance Pack footprint from within the target JVM.

When you first deploy the WebCenter Portal Performance Pack app to the target JVM, WebCenter Portal Performance Pack is attached as a Java agent in the target JVM. Once attached, instrumentation of the target classes/methods occurs so that event and method analysis can be performed. When you undeploy WebCenter Portal Performance Pack, any changes that were made to the target JVM are rolled back and the JVM returns to its initial clean state.

To undeploy, run `./undeploy.sh <WLS_SERVER_NAME>`

This is what happens:

- The WebCenter Portal Performance Pack webapp is undeployed from the target JVM
- All currently instrumented classes are rolled back to their initial state.

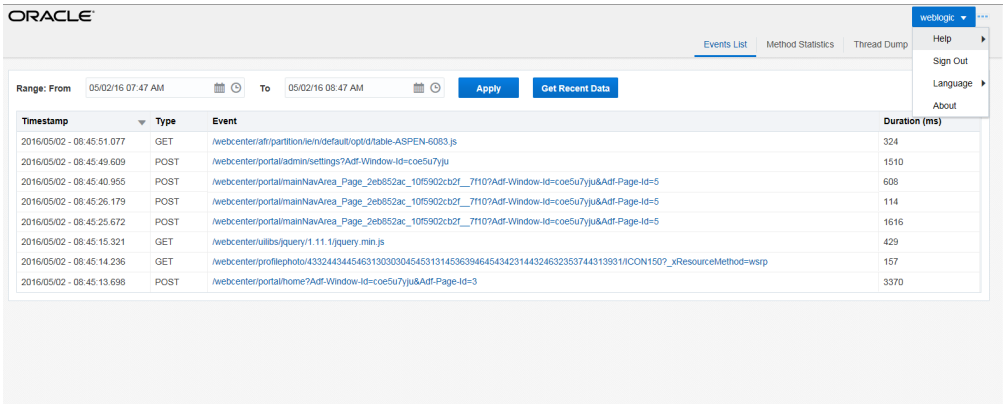
  classloaders are reloaded with non-instrumented classes.
- All WebCenter Portal Performance Pack related threads are removed from the target JVM.

The target application can still function normally without you having to perform a restart.

Quick Tour of the Landing Page

When you start WebCenter Portal Performance Pack, you're prompted to enter your login credentials. You should enter the username and password that belongs to the Administrator's role in the WebLogic Server. You're then directed to the landing page. This page provides access to all the resources required to monitor and diagnose performance issues related to the selected Oracle WebCenter Portal instance.

When you start WebCenter Portal Performance Pack, the target is automatically set to the JVM WebCenter Portal Performance Pack it's installed against. Also note the location of the user menu. This menu has options for viewing your language preferences, accessing help (including the Getting Started overlay), and signing out.



Use the landing page as a starting point to identify slow requests and drill down to diagnose performance bottlenecks. The tabbed pages make it easy to navigate between the available features. The information accessed from the **Help** button is based on the page you currently have open. The Events List is the default page.

Now that you have an overview of the landing page, let's take a closer look at the key features that you'll be using.

Feature	Description	More Info
Events List	Displays a list of events the server's processed during the time range specified. The initial range is set from the present to 1 hour back. From this page you can: • Change the specified time range or click **Get Recent Data** to get events from the previous hour. • View the HTTP request type for each request • Sort requests by duration (in milliseconds) to find slower requests • Hover over an event to view the full event name. This includes the full URL and any query strings. • Drill down to a specific event details where you can diagnose problem areas based on instrumentation and method sampling.	See Working with Events

Feature	Description	More Info
Method Statistics	Displays all methods from the present to the last reset date. From this page you can: • Sort methods by method, hits, or duration • Identify problem methods based on the duration and number of hits for each method. Optimizing methods with higher hits and/or longer duration might have an higher impact on your system's performance and scalability, especially the lower-level methods. • Hover over a method to view the full method name. You can also copy the name to the clipboard. • Click **Reset Statistics** to erase all historical data and update the last reset date to the current date and time. This feature gives you a fresh start to begin collecting method statistics for a new test scenario.	See How do I find methods for optimization?
Thread Dump	Displays all threads returned from the JVM in the order provided. Each calling stack is a collapsible section that contains the method calls. From this page you can: • Filter threads based on thread health. (ACTIVE, STANDBY, SUSPENDED, STUCK) • Filter threads based on their current state. (WAITING, RUNNABLE, TIMED-WAITING, PENDING) • Expand specific threads to view the methods associated with that thread.. Select to watch specific methods. **Note:** You can't watch methods associated with WebCenter Portal Performance Pack as they're disabled.	See How do I locate cause of serious performance issues?

Feature	Description	More Info
Configuration	Displays the current configuration settings as well as key methods that have been configured to be instrumented. From this page you can: • Enable and/or disable instrumentation and sampling • Change configuration settings • Reset the configuration back to the previous saved configuration • Click **Update** to commit changes but not save to the configuration file • Click **Save** to save changes to the configuration • Add additional class/methods to be instrumented • Select to watch or unwatch pre-configured methods	See Configuring Instrumentation

2
Working with Events

This section discusses how to use the event listing as a starting point for performance analysis. It describes how to drill down into specific requests and diagnose potential bottlenecks and problem areas using instrumentation and sampling tree as well as analyzing method statistics.

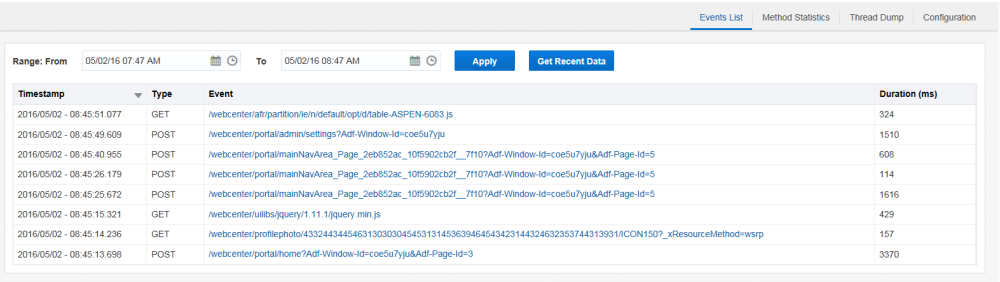
Topics

- What can I do with the Events List?
- How do I locate hot paths in my code?

What can I do with the Events List?

The Events List is the starting point for performance analysis. It provides you with a list of events the server has processed during a specified date and time range. Duration of time taken to process each event can help you determine what events require further analysis.

The Events List is the default page that you see when you sign in, as shown.



When you first access this page, the date and time range is set from the present to 1 hour back, and the list of events during that time are displayed by default. You can scope a different set of events by changing one or both date/time selections. Click **Apply** to refresh the view. Click **Get Recent Data** to get the set of events up to the last hour.

The following information is provided for each event shown. You can sort events by *Timestamp, Type,* or *Duration.*

Event	Description
Timestamp	Displays the date and time the event was processed. By default, the events are sorted to display the latest event first.
Type	Displays the HTTP method for this request (GET/POST)

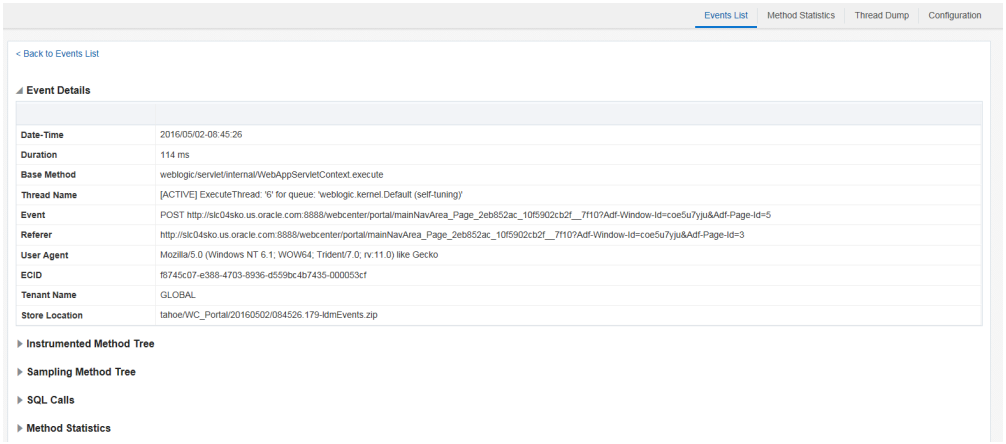
Event	Description
Event	Displays the event name as (.....path/resource); the host name and port part of the event may be truncated to only show the beginning of the path. You can hover over the name to display the full URL as well as any query strings. Click the name to view the details for this event. For more information about event details, see How do I locate hot paths in my code?.
Duration (ms)	Displays the time (in milliseconds) taken to process this event.

How do I locate hot paths in my code?

Use the Event Details page to drill down to a specific request's profile information and identify hot paths and methods using the instrumented method tree, sampling method tree, as well as SQL and method statistics.

WebCenter Portal Performance Pack is automatically configured to ignore non-essential information and track only key methods; therefore, when you drill down on a specific request you're not overwhelmed with a massive amount of data to search through. The Event Details page contains the following sections to help you pinpoint possible problem areas:

- **Event Details** — Shows top-level information about the event
- **Instrumented Method Tree** — Shows how much time is being spent on individual key methods
- **Sampling Method Tree** — Compliments the instrumented method tree by identifying cases where the configured methods may have missed other key methods
- **SQL** — Shows how much time was spent making calls to the database
- **Method Statistics** — Shows method timings associated with this method within the scope of this particular event (request)



You can collapse and expand each section so that you can zero in on the most critical information needed.

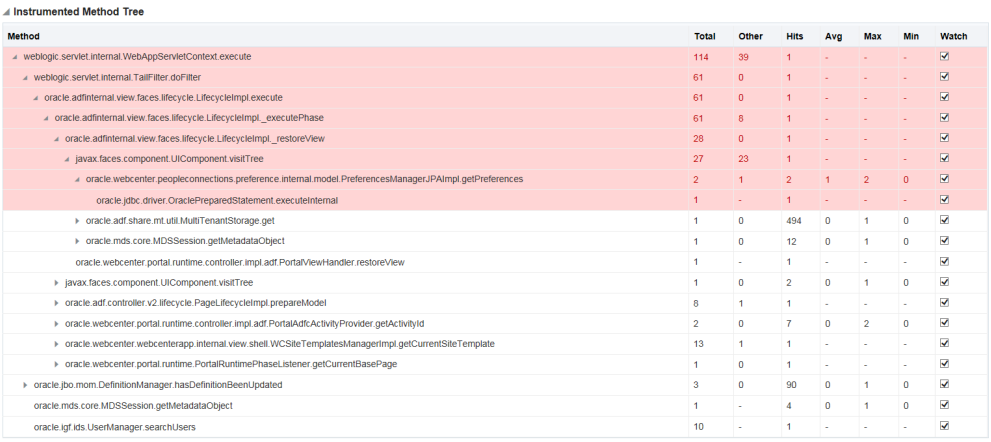
About Event Details

The event details provides information about the selected event, including but not limited to:

- **Timestamp** — Date and time the event was processed by the server
- **Duration** — Displays the total time taken to process the event.
- **Base Method** — Displays the root method as `package/class.method`
- **Event** — Displays the HTTP method (GET/POST) followed by a separator and the full URL, and any parameters
- **User Agent** — Displays the account that the customer used for authentication on the application (the target application being monitored)

About the Instrumented Method Tree

Because of the complexity of our code, a single request may contain a significant number of methods. With that in mind, we use a configuration algorithm to only include key methods in the instrumented method tree so that you can immediately focus on what is important for performance analysis. The instrumented tree is represented as a hierarchical table of data. Using this hierarchical tree format, you can easily determine the most expensive methods because those methods have a pink background and their statistics are highlighted in red. It's OK to ignore the high-level methods, which by nature are slower as they contain other methods. For example, the base method `weblogic/servlet/internal/WebAppServletContext.execute` appears slow because the time reflects the total timing for the whole request.



Although you want to ignore the actual high-level method, you should drill down further into this method by expanding the node and seeing if in fact it's a hot path with low-level methods eating up the time. The following information can help you determine areas that need further investigation.

Instrumented Tree Item	Description
Method	Displays as `package/class.method`. The full method is shown in a tooltip on hover.
Total Time	Displays the total time spent on this method.
Other Time	Indicates the time taken by methods that are not included in instrumentation.

Instrumented Tree Item	Description
Hits	Number of times this method is called during execution of the request
Min/Max/Avg	If the number of hits is greater than 1 for this method, the minimum, maximum, and average times are calculated.
Watch	By default, all methods in the tree are watched. Click here to remove the method from the instrumentation.

The sampling method tree works alongside the instrumented method tree because it can catch expensive methods that are not yet instrumented. Therefore, if you see a large Other Time statistic calculated, you should examine the methods included in the sampling.

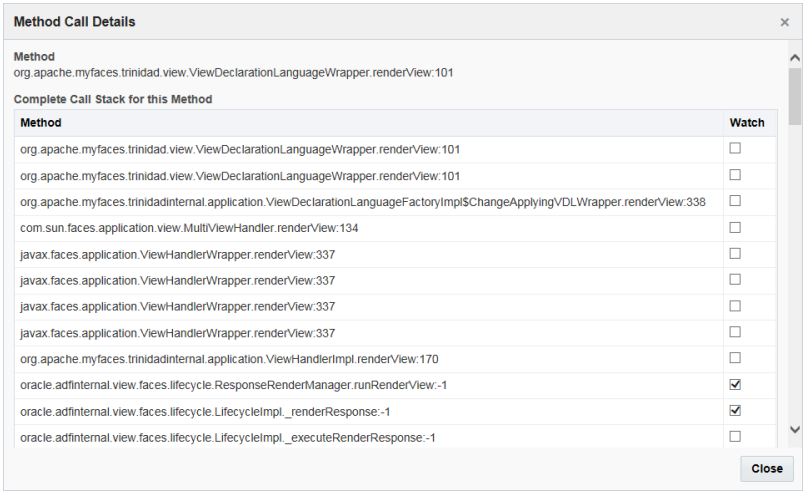
> **Note:**
>
> When you remove a method from instrumentation only that method is removed. Lower-level methods remain in the tree as they are selected based on the algorithm not because they are children of an instrumented method.

About the Sampling Method Tree

The sampling method tree operates in concert with the instrumented method tree. The stack traces are sampled periodically and this tree displays the methods that are slower than the sampling rate. For example, if a request takes 1 second to execute and the sampling rate is 50 milliseconds, sampling is done 20 times. The sampling method tree shows you the frequency (weight) a method was tracked during the lifecycle of the request and if it's taking too long to process. By default, the Sampling rate is set to 50 milliseconds, but you can change the rate based on your requirements.

For more information about changing the default rate and other pertinent configuration settings, see How do I change configuration settings to suit my system needs?.

The tree is represented as a hierarchical table of data. Just like the instrumented tree, you can expand or collapse the methods inside the tree. Instead the table indicates the number of child processes. To see these methods, just click on the method. This opens the Method Calls Details dialog, which displays the complete call stack for this method. From this dialog you can select to watch any methods in the calling stack.



If you find any methods in the sampling tree that you think should be instrumented for further analysis you can select to watch them. You don't have to stop and restart the tool to add a method to the configuration for instrumentation.

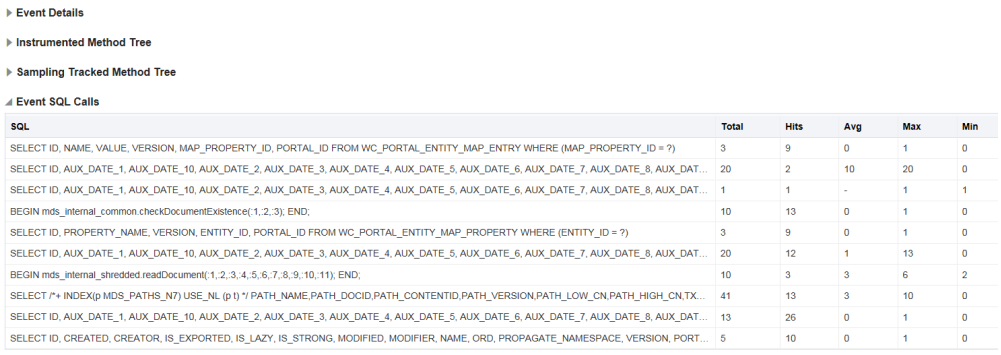
1. Select the **Watch** option.
2. Go to WebCenter Portal and execute the request.
3. Go to the Event List and drill down on the new request.

   You can see the method that you chose to watch is now included in the instrumented method tree. You don't have to stop and restart the target portal server.

> **Note:**
>
> The sampling tree also displays any methods in your custom code that is slowing down the system performance. You can choose to instrument these slow class.methods by simply selecting to Watch them.

About SQL

When processing requests, Oracle products, including WebCenter Portal, could make some calls to the database. The SQL section contains a listing of all these calls during the processing of this request. It shows how many times each call was made as well as the average time it takes to process. Hover over the SQL name to view the full SQL. You can copy the full SQL to the clipboard.



This information, along with the information gleaned from the instrumentation method tree and sampling method tree can be very useful in narrowing your performance issue.

About Method Statistics

The Method Statistics section provides aggregated statistics for all the methods executed for this request in a list. Instead of expanding and collapsing nodes to find specific methods, you can locate them more easily here. You can then see from the aggregated statistics if they are expensive and need to be optimized.

For more information about method statistics, see How do I find methods for optimization?.

3

Working with Method Statistics and Dump Threads

This section discusses how to find methods for optimization as well as how to find methods that are causing serious performance issues.

Topics

- How do I find methods for optimization?
- How do I locate cause of serious performance issues?

How do I find methods for optimization?

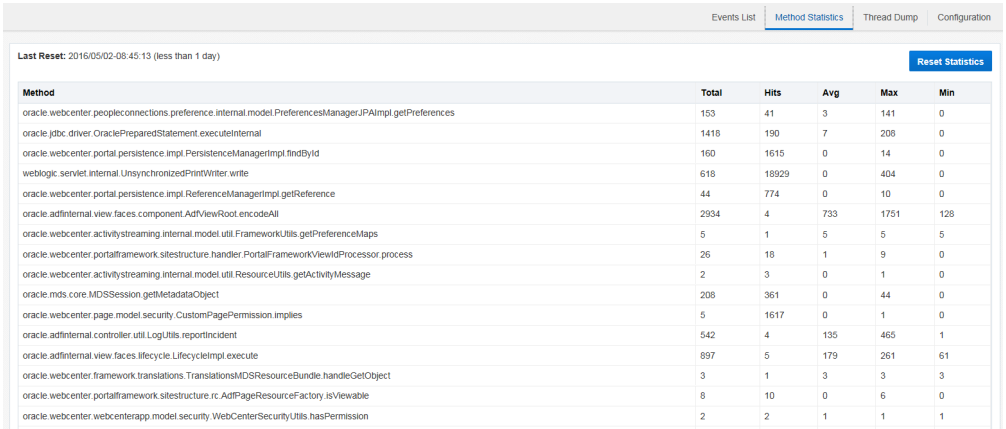
The Method Statistics page is used to help you identify methods that can be optimized based on aggregated statistics.

Let's say you have expanded your company and will be hiring 50 new employees. This means that additional users will soon be performing the set of tasks that current users perform in Oracle WebCenter Portal on a daily basis. You want to know if you can optimize any methods to make sure that performance isn't compromised by adding these new users.

To find methods for possible optimization:

1. Go to the Method Statistics page and click **Reset Statistics** to remove all the current methods listed.
2. Sign in to Oracle WebCenter Portal and perform the required set of tasks.
3. Go to the Method Statistics page.

   Every instrumented method that was executed during the test has been aggregated and is listed in the table.



Method	Total	Hits	Avg	Max	Min
oracle.webcenter.peopleconnections.preference.internal.model.PreferencesManagerJPAImpl.getPreferences	153	41	3	141	0
oracle.jdbc.driver.OraclePreparedStatement.executeInternal	1418	190	7	208	0
oracle.webcenter.portal.persistence.impl.PersistenceManagerImpl.findById	160	1615	0	14	0
weblogic.servlet.internal.UnsynchronizedPrintWriter.write	618	18929	0	404	0
oracle.webcenter.portal.persistence.impl.ReferenceManagerImpl.getReference	44	774	0	10	0
oracle.adfinternal.view.faces.component.AdfViewRoot.encodeAll	2934	4	733	1751	128
oracle.webcenter.activitystreaming.internal.model.util.FrameworkUtils.getPreferenceMaps	5	1	5	5	5
oracle.webcenter.portalframework.sitestructure.handler.PortalFrameworkViewIdProcessor.process	26	18	1	9	0
oracle.webcenter.activitystreaming.internal.model.util.ResourceUtils.getActivityMessage	2	3	0	1	0
oracle.mds.core.MDSSession.getMetadataObject	208	361	0	44	0
oracle.webcenter.page.model.security.CustomPagePermission.implies	5	1617	0	1	0
oracle.adfinternal.controller.util.LogUtils.reportIncident	542	4	135	465	1
oracle.adfinternal.view.faces.lifecycle.LifecycleImpl.execute	897	5	179	261	61
oracle.webcenter.framework.translations.TranslationsMDSResourceBundle.handleGetObject	3	1	3	3	3
oracle.webcenter.portalframework.sitestructure.rc.AdfPageResourceFactory.isViewable	8	10	0	6	0
oracle.webcenter.webcenterapp.model.security.WebCenterSecurityUtils.hasPermission	2	2	1	1	1

By scanning the list of methods and concentrating on the number of hits and total time for each method, you can easily determine methods that are the most expensive. By eliminating high-level methods you can then limit your search to low-level methods.

For these methods you should also take into consideration the minimum, maximum, and average times. By optimizing the methods that are most expensive you should not find any performance issues when new users are added.

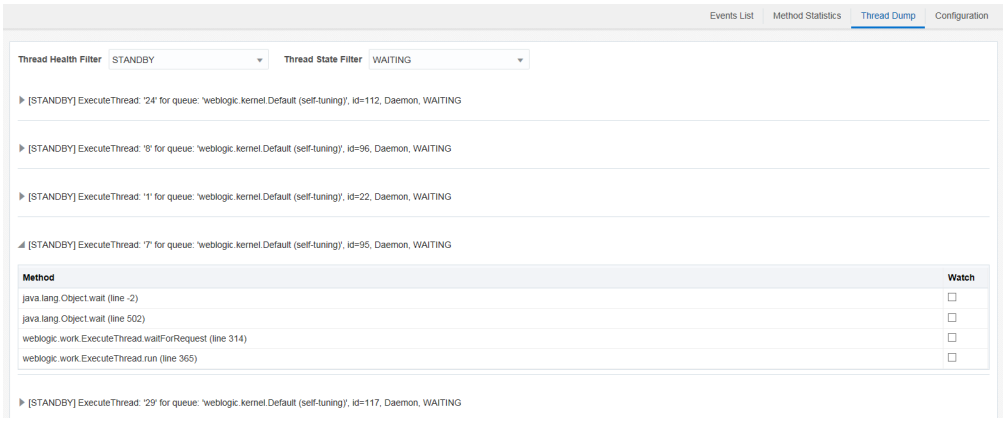
How do I locate cause of serious performance issues?

If users are experiencing serious slowness or the system has hung when using Oracle WebCenter Portal, you can perform a thread dump.

The thread health status is what's assigned to threads based on certain time thresholds that have been allotted for specific actions to complete. For example, let's say a thread is waiting for response from the database. While it's waiting for a new request, it's given the health status `STANDBY`. By default, the request timeout is set to 10 mins in the WebLogic Server. If a request has not finished processing after the time specified, the health status changes to `STUCK`.

Threads are also assigned a server status that you can see displayed at the end of the thread name.

To perform a thread dump, simply click the Thread Dump tab. The page opens with the results of the dump. You can filter the data by health status (ALL, ACTIVE, STANDBY, SUSPENDED, STUCK and/or server status (WAITING, RUNNABLE, TIMED-WAITING, and PENDING). Note that only states that are present in the current thread dump appear in the drop-down menus.



If your system has a serious performance issue, you can rapidly diagnose the problem causing the slowdown by filtering by the status `STUCK`.

You can expand specific threads to view the methods associated. Note that the methods may take awhile to load. You can then select to instrument suspect methods by selecting to watch them. You don't have to stop and restart WebCenter Portal Performance Pack or the target JVM to begin watching these suspect methods.

4

Configuring Instrumentation

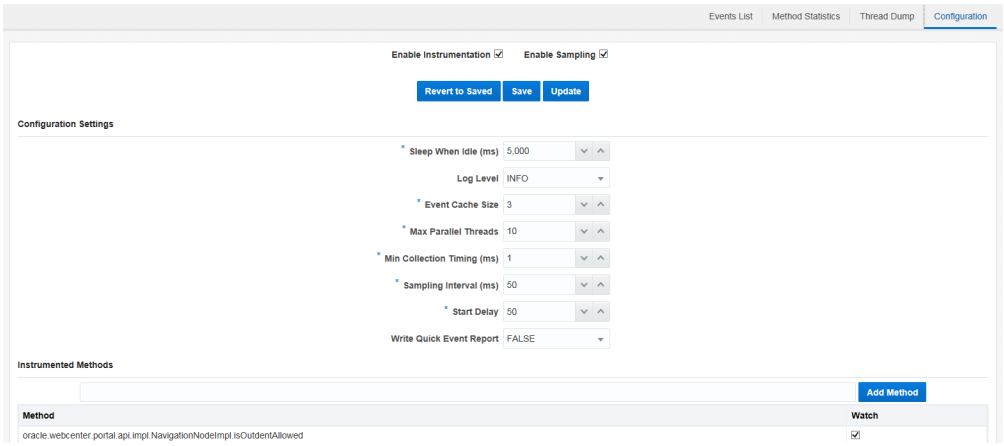
This section discusses how to make changes to the current configuration settings, and how to add methods to and remove methods from the configuration.

Topics

- How do I change configuration settings to suit my system needs?
- How do I add methods to the instrumentation configuration?

How do I change configuration settings to suit my system needs?

The Configuration page lets you enable instrumentation and sampling, as well as change certain configuration settings.



You can change the following configuration settings.

Configuration Setting	Description
Enable Instrumentation	Select to enable method instrumentation. Unselect to disable method instrumentation.
Enable Sampling	Select to enable method sampling. Unselect to disable method sampling. **Note:** You can only change this setting if Instrumentation is enabled.
Sleep When Idle	Defines how frequent to wake the internal thread to process instrumentation events. Default is *5 seconds.* Note that if you set the frequency to a greater value, you may see a delay with events appearing in the events list. A frequency less than the default time means the events list is more up-to-date but there's a cost with system overhead.

Configuration Setting	Description
Log Level	Defines the log-level. The default is *INFO*. Valid values are: • *NONE* — No logging • *ERROR* — A WebCenter Portal Performance Pack system error has occurred and must be fixed • *WARNING* — A problem has occurred but it shouldn't affect the WebCenter Portal Performance Pack system functionality • *INFO* — Provides informational messages regarding the state of the WebCenter Portal Performance Pack system • *PERF* — Provides informational messages regarding instrumentation • *DEBUG* — Provides detailed messages to assist in the diagnosis of a problem with the WebCenter Portal Performance Pack system
Event Cache Size	Defines the cache size of the monitored event. The default is *3.* The greater the cache size, the faster the reload time but with more memory required.
Max Parallel Threads	Defines the maximum number of threads to be monitored in parallel. Default value is *10.* For a heavily loaded system you may want to increase this setting to avoid missing instrumentation. However, the greater the number, the more memory required.
Collection Timing (ms)	Defines the minimum method timing to be collected for profiling. Methods that spend less than this timing are not recorded or reported. Default value is *1 millisecond.* This setting lets you reduce the amount of data that's collected so you can concentrate on slower methods.
Sampling Interval (ms)	Defines the interval between two consecutive samplings. Default is *50 milliseconds.* The smaller the sampling interval, the more precise the sampling but with greater system overload.
Start Delay (ms)	Defines the delay time before instrumentation and sampling starts. Default is *50 milliseconds.* This setting lets you ignore any small and fast events.
Write Quick Event Report	Defines whether or not to produce a separate event report using text format. Default is *False.* Valid values are *True* or *False.*

After you make changes, you can:

- **Update** — Saves the changes but does not update the configuration file.
- **Save** — Saves the changes and updates the configuration file.

  Here's the location of the configuration file: `<domain_home>/tahoe/<server_name>/tahoe.cfg`
- **Revert to Saved** — Removes your changes and restores the last saved configuration settings.

How do I add methods to the instrumentation configuration?

You can add and remove methods to and from the instrumentation configuration using the sampling method tree, the thread dump, or the Configuration page. You can also add and remove methods from the instrumented method tree.

You don't have to stop and start WebCenter Portal Performance Pack or the target JVM when you add or remove methods. You just perform the task.

To add or remove methods using the sampling method tree:

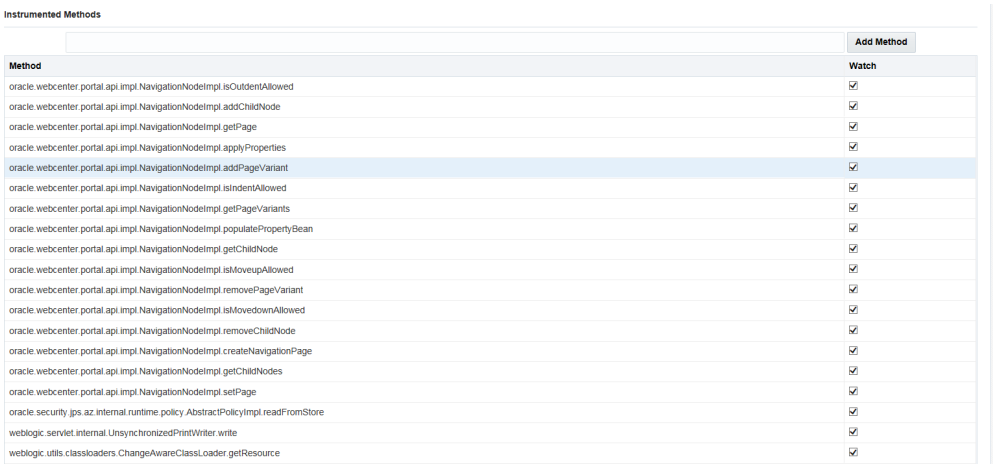
1. Go to the Configuration page and verify instrumentation and sampling have been enabled.

> **Note:**
>
> Once you've enabled the instrumentation and sampling, you must access the webapp to ensure that data is loaded in the sampling tree.

2. Go to the Sampling Method Tree section to see the key methods that may have been missed in the configuration.
3. Find the method you want to instrument and select **Watch.** To remove methods from being instrumented, just unselect **Watch** from the instrumentation and/or sampling method trees.

   A confirmation dialog is displayed for both actions.

Now let's say you've added some new code to Oracle WebCenter Portal and you want to monitor the performance. You can add and/or remove instrumented methods manually from the Configuration page.



To add new methods to the configuration

1. Go to the Configuration page and verify instrumentation and sampling have been enabled.
2. Go to the Instrumented Method section and simply add your method (`package.class.name`) and click **Add Method.**

3. Repeat the previous step until you've added all the methods you're interested in from your new code.
4. Save the configuration.
5. Sign in to Oracle WebCenter and perform functions using your new code.
6. Return toWebCenter Portal Performance Pack and monitor the performance for your new methods using the Events list or Method Statistics.

> **Note:**
>
> Methods displayed in "Grey" colored *Italics* indicate classes that are yet to be loaded by the JVM. You should wait for the class to be invoked by the program and start watching it once it is loaded. Methods displayed in "Blue" indicate changes that are not yet saved to the configuration file.

You can remove methods from the configuration by simply unselecting **Watch** for selected methods and then saving the configuration.

> **Note:**
>
> If you click **Update**, you'll change the properties in memory and the properties do take effect; but they don't persist to file until you click **Save**.