HP Operations Orchestration Software

Software Version: 7.50

Administrator's Guide

Document Release Date: November 2008

Software Release Date: November 2008

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

Trademark Notices

All marks mentioned in this document are the property of their respective owners.

Finding or updating documentation on the Web

Documentation enhancements are a continual project at Hewlett-Packard Software. You can obtain or update the HP OO documentation set and tutorials at any time from the HP Software Product Manuals web site. You will need an HP Passport to log in to the web site.

To obtain HP OO documentation and tutorials

1. Go to the HP Software Product Manuals web site (*http://support.openview.hp.com/selfsolve/manuals*).
2. Log in with your HP Passport user name and password.

   OR

   If you do not have an HP Passport, click **New users – please register** to create an HP Passport, then return to this page and log in.

   If you need help getting an HP Passport, see your HP OO contact.
3. In the **Product** list box, scroll down to and select **Operations Orchestration**.
4. In the **Product Version** list, click the version of the manuals that you're interested in.
5. In the **Operating System** list, click the relevant operating system.
6. Click the **Search** button.
7. In the **Results** list, click the link for the file that you want.

Support

For support information, including patches, troubleshooting aids, support contract management, product manuals and more, visit one of the two following sites:

- *https://support1.opsware.com/support/index.php*
- *http://www.hp.com/go/hpsoftware/DCA_support*

Where administrative tasks are documented

Some administrative tasks are performed from within HP OO Central and some are performed outside of Central. For instance, you configure and enable external authentication providers on the Administration tab in Central, but making other configuration changes to Central can require work with files outside of the Central Web application.

You can complete many administrative tasks from within Central, on the **Administrative** tab. For example, on the Central **Administrative** tab, you can manage flow runs; enable and configure user authentication by AD, LDAP, and Kerberos; enable ROI reporting; and configure a Central cluster. If you can complete an administrative task entirely from within Central, you will find the procedure for performing the task in Central Help (and its PDF equivalent, the Central *User's Guide*).

This *Administrator's Guide* provides administrative concepts and procedures for completing administrative tasks that you cannot complete solely from within Central. Such administrative tasks include the following:

- Configuring Active Directory or LDAP over SSL
- Configuring HP OO for extended functionality
- Changing the maximum size of the wrapper.log file
- Enabling single sign-on for flows started with Rsflowinvoke.exe
- Enabling and disabling run-scheduling concurrency for Scheduler
- Changing Studio configurations in the Studio.properties file
- Backing up HP OO
- Supporting a Central server cluster
- Administering a Central server cluster

Overview

Administering Operations Orchestration (HP OO) includes:

- Managing security, which comprises managing *Security: Users, Groups, Capabilities, and Permissions*.

  You can map HP OO user roles either to external or to internal group.
- Enabling HP OO to run flows against remote machines and integrate them with other applications. For more information, see *Configuring HP OO for extended functionality*.
- Changing configurations for Central, including:
  - *Changing the maximum size of the Wrapper.log file*
  - *Enabling single sign-on for flows started with the Java Flow Invoke tool*
  - *Enabling and disabling run-scheduling concurrency for Scheduler*
- Changing configurations for Studio, including the Studio host server, communications port number, and protocol used.
  - The database user account and password.

    - The maximum size of the Jetty service Wrapper.log file.
- *Backing up HP OO*
- *Supporting a Central server cluster*

For information on administering flow runs, see Help for Central.

Default ports used by HP OO components

By default, HP OO components use the following ports:

- Central: 8443

  If Central servers are clustered, port 45566 is used for SSL communications between JGroups nodes
- Between HP OO components, such as Central, Scheduler, and RAS: 18443
- RAS: 9004
- Scheduler: 19443

Security: Users, Groups, Capabilities, and Permissions

Many of the HP OO security features take place in the background. From the point of view of the HP OO administrator, author, and user, HP OO security deals with:

- Security of communications between HP OO system components and between those components and the flows' target systems.

  The aspect of this that is relevant to authors is the use of the HTTPS protocol and SSH for HP OO communications.
- User authentication, or logging in.

  You can configure HP OO to use external Active Directory, LDAP, or Kerberos authentication of user logins. To accomplish this configuration, you use the Central **Administration** tab. For information on doing so, see Help for Central.

  Note: In order to make communications secure, you can configure Active Directory to run over the Secure Sockets Layer, using the LDAPS protocol. For information on doing so, see *Configuring Active Directory or LDAP over SSL (LDAPS protocol)*.
- Managing HP OO users and groups and controlling the operations and flows that they can run.

  In HP OO, groups are the basic unit for managing access to flows and controlling what they can do with the flows, but you could manage them with individual users as well. You manage groups' and users' rights by granting them:

    - Capabilities (types of actions that users can perform).

      To give your flow authors the capability to author flows, for instance, you might create a group, "Authors", which you would assign the AUTHOR capability. You manage users, groups, and capabilities from the Central Web application. For information on doing so, see Help for Central.

- Access to specific objects (such as folders, flows, operations, and system accounts within Studio).

  For example, for an author to make and test changes to a flow that has subflows, he or she needs to have the AUTHOR capability as well as the READ, WRITE, and EXECUTE permissions for the flow and the LINK permission for any subflow that is used in the flow.

  For a Central user to run a certain flow (flow X), you would add the Central user to the LEVEL_ONE, LEVEL_TWO, or LEVEL_THREE group, any of which comes with the capabilities needed to run flows, and you would assign him or her the EXECUTE permission for flow X.

  Authors assign permissions for flows and associated objects in Studio. For information on doing so, see Help for Studio.

The following graphic shows how the concepts of users, groups, capabilities, and permissions interact to let administrators and authors define how individuals can react with which objects.

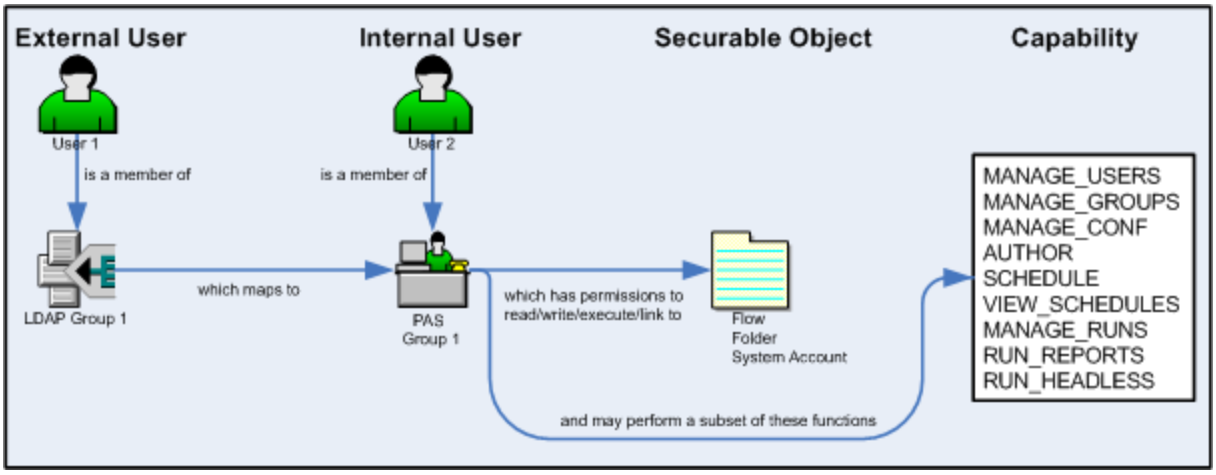

Figure 1 - Components of access control in HP OO

- Protecting the confidentiality of sensitive credentials.

  System accounts are HP OO objects that a user can invoke in order to run a flow in a location that requires specific authentication and permissions that the flow user might not have, without the user's being able to see the credentials he or she is using to run the flow. This means that flow users can run flows wherever necessary, but without the user's having to enter the credentials necessary to get access to the flows' targets, while the credentials remain protected.

Configuring Active Directory or LDAP over SSL (LDAPS protocol)

Machines ordinarily communicate with Active Directory using Lightweight Directory Access Protocol (LDAP, a clear-text protocol). To encrypt communications, you can set HP OO to communicate with Active Directory over Secure Sockets Layer (SSL). The LDAPS protocol is the LDAP protocol encrypted with SSL.

Important: If you are configuring LDAP to run over SSL, see your LDAP administrator about exporting a certificate, then complete the following procedure, skipping steps 1 through 9.

To configure Active Directory to communicate with LDAPS

1. On the AD machine, start Microsoft Management Console (mmc.exe).
2. To add the Certificates snap-in:
   a. From the **File** menu, select **Add/Remove Snap-in.**
   b. In the **Add/Remove Snap-in** dialog, click **Add.**
   c. In the **Add Standalone Snap-in** dialog, select **Certificates** and click **Add.**
   d. Select **Computer Account** and then click **Next.**
   e. In the **Select Computer** dialog box, click **Finish.**
   f. In the **Add standalone Snap-in** dialog, click Close, and in the **Add/Remove Snap-in** dialog click **OK.**
3. In the MMC console, open **Certificates (Local Computer)** and its subfolders **Personal\Certificates.**
4. In the right panel, find the certificate for the AD.

   For example, if the AD is "ad.mycompany.com", you should see a certificate with:

   - The same name as the AD.
   - An intended purpose of **Client Authentication.**
   - A **Domain Controller** certificate template.
5. Right-click the certificate, point to **All Tasks,** then click **Export.**
6. When the Certificate Export Wizard starts, click **Next.**
7. Make sure that **No, do not export the private key** is selected, then click **Next.**
8. In the **Export File Format** page, select **DER encoded binary X.509 (CER)** and click **Next.**
9. In the **File to Export** page, select the location and name of the exported certificate.

   The certificate file has a .cer extension.
10. Copy the exported certificate file to a location on the server machine on which you have installed Central.
11. On the Central machine, stop the RSCentral service.
12. Open a command-line window and run the following two commands:

```
cd %ICONCLUDE_HOME%\jre1.6\bin
keytool –keystore "%ICONCLUDE_HOME%\jre1.6\lib\security\cacerts" –
import –file <path_to_cert_from_step9> -alias <some_alias>
```

    In this command, **alias** is used to identify the certificate. For example, it could be named something like "mycompany_ad_cert".
13. When prompted for the certificate store's password, type "changeit".

    "changeit" is the default password. For information on using "keytool" to change the password, see the keytool documentation.
14. When you are prompted to confirm that this certificate should be trusted, type **Yes**.
15. To verify that the certificate was imported, run the following command:

```
keytool –keystore "%ICONCLUDE_HOME%\jre1.6\lib\security\cacerts"
```

```
-list -alias <some_alias>
```

The default certificate store password is “changeit”.

You should see a summary of the certificate.

16. In %ICONCLUDE_HOME%\Central\conf, open the Central.properties file in a text editor.
17. Locate the line that begins with “ADAuthGroupBased.URL” and set it to specify the LDAPS protocol, by modifying it to read as follows:

    ```
    ADAuthGroupBased.URL=LDAPS://<your_AD>:<port> ;
    ```

    For example, if your AD is ad.mycompany.com and you have configured it to use the default port 636, the line should read as follows:

    ```
    ADAuthGroupBased.URL=LDAPS://ad.mycompany.com:636 ;
    ```

18. Restart the RSCentral service.

Replacing the HP OO security certificate

The high-level procedure for replacing a HP OO security certificate breaks down into several sections:

- *Replacing the HP OO security certificate*, using the following procedure.
- *Modifying the RAS keystore to share mutual SSL authentication with Central*
- *Replacing the Studio certificate*
- *Testing the certificates*

Notes:

- Because JRAS and NRAS have been merged into a single RAS, you no longer need a separate NRAS certificate.
- This section assumes that you are replacing an HP OO 7.10 security certificate. If you are replacing a security certificate created by Opsware PAS 7.0, it may be identified as an Opsware security certificate. In the following procedure, “HP OO security certificate” should be taken as referring to that certificate.

To replace the HP OO security certificate

1. To back up the key stores, use the following:

   ```
   %iconclude_home%\Central\conf\copy rc_keystore rc_keystore.original
   >copy %iconclude_home%\ras\Java\Default\webapp\conf\ras_keystore.jks
   ras\Java\Default\webapp\conf\ras_keystore.jks.original
   ```

2. Generate the new key pair in the rc_keystore, using a command like the following example:

   ```
   >keytool -genkey -dname "CN=ubs.ag.com, O=UBS, C=SW" -alias
   ubscentralssl -keypass bran507025 -keystore rc_keystore -storepass
   bran507025 -keyalg "RSA" -validity 1095
   ```

   Note: The value for CN must match the name that you will use in the URL, which is not necessarily the host name of the server.

3. Verify the keypair, look for "Certificate[1]" with the command:

```
>keytool.exe -list -keystore rc_keystore -storepass bran507025 -v -
alias ubscentralssl
```

The output of the command should look something like the following:

```
Alias name: ubscentralssl
…
Certificate chain length: 1
Certificate[1]:
…
```

4. Generate the certificate request:

```
>keytool -certreq -alias ubscentralssl -file ubscentralssl.csr -
keypass bran507025 -keystore rc_keystore -storepass bran507025 -
keyalg "RSA"
```

5. Submit the request to your certificate authority (CA), using the method determined by your organization.

   Typically, this involves a Web form in which you paste in the contents of the CSR.

   Important: Specifically for Central, you'll need a certificate that is purposed for both server authentication and client authentication. This will probably not be the default SSL/Web Server certificate issued by the CA.

6. When you get the response back from the CA, save the file as response.cer.

   OR

   Copy the key into a new file and name the new file response.cer.

7. To verify that the certificate that the CA issued was correctly purposed (for both server authentication and client authentication), open response.cer.

8. On the **Details** tab, scroll down to the **Enhanced Key Usage** field.

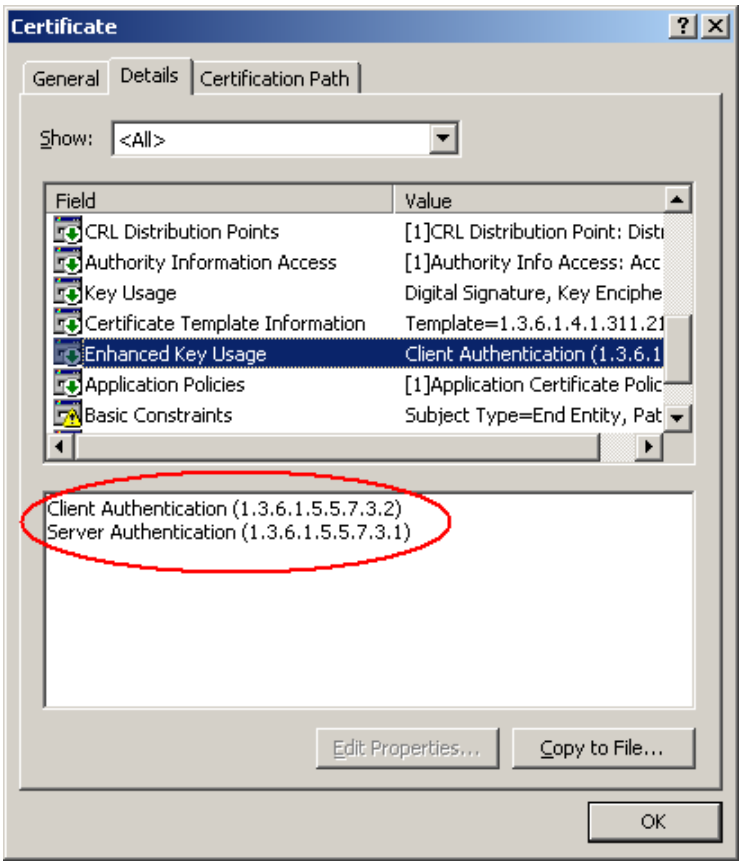

Figure 2 - response.cer editor

Before importing the response, import the certificates for each CA in the chain leading up to the CA that signed the certificate request.

9. To obtain the certificates for each CA:
   a. Open the response.cer file and click the **Certification Path** tab.
   b. If the top level CA is the same CA for which you added a certificate when configuring LDAP over SSL, you can skip the CA and go to the next.
   c. Highlight each additional CA in the chain and click **View Certificate**.
   d. On the **Details** tab for that certificate, click **Copy to File** and follow the instructions to export the cert as a DER encoded certificate.
10. If there are any intermediate CA servers, import the certificates for those, using commands like the following:

```
>keytool -import -alias ubsint1ca -file ubsint1ca.cer -keystore
rc_keystore -storepass bran507025
Certificate was added to keystore
>keytool -import -alias ubsint2ca -file ubsint1ca.cer -keystore
rc_keystore -storepass bran507025
Certificate was added to keystore
```

11. When all CA certificates in the chain are imported and trusted, use commands like the following to import the response received from the certificate authority.

    Make sure that the alias used in this command matches the alias used when you generated the key in step 2.

```
>keytool -import -trustcacerts -alias ubscentralssl -file
response.cer -keystore rc_keystore -storepass bran507025
```

The response to that command should be something like the following:

```
Certificate reply was installed in keystore
```

12. To verify that the certificate and associated certificate chain are in the store, use a command such as:

```
>keytool -list -keystore rc_keystore -storepass bran507025 -v -alias
ubscentralssl
```

The output should look something like the following example:

```
Alias name: ubscentralssl
Creation date: May 9, 2007
Entry type: keyEntry
Certificate chain length: 4
Certificate[1]:
Owner: CN=central.qa.opsware.services.ubs.com, O=UBS AG, C=SW
Issuer: CN=UBS Issuing CA 2, O="UBS AG."
…
Certificate[2]:
Owner: CN=central.qa.opsware.services.ubs.com, O=UBS AG, C=SW
Issuer: CN=UBS Issuing CA 2, O="UBS AG."
…
Certificate[3]:
Owner: CN=central.qa.opsware.services.ubs.com, O=UBS AG, C=SW
Issuer: CN=UBS Issuing CA 2, O="UBS AG."
…
Certificate[4]:
Owner: CN=central.qa.opsware.services.ubs.com, O=UBS AG, C=SW
Issuer: CN=UBS Issuing CA 2, O="UBS AG."
…
```

13. To remove the default HP OO SSL certificate from the store:

```
>keytool -delete -alias pas -keystore rc_keystore -storepass
bran507025 –v
```

14. To export the new certificate for central that you have created in the rc_keystore

```
>keytool -export -keystore rc_keystore -storepass bran507025 -alias
ubscentralssl -file ubscentralssl.cer
```

Next, you modify the RAS keystore share mutual SSL authentication with Central, using the new Central certificate that you just created and a new RAS certificate that you will request.

Modifying the RAS keystore to share mutual SSL authentication with Central

To modify the RAS keystore to share mutual SSL authentication with Central

1. Delete the old certificates:

```
>keytool -delete -alias pas -keystore
%iconclude_home%\ras\Java\Default\webapp\conf\ras_keystore.jks -
storepass bran507025 -v
>keytool -delete -alias 1 -keystore
%iconclude_home%\ras\Java\Default\webapp\conf\ras_keystore.jks -
storepass bran507025 -v
```

2. Generate the new key pair:

```
>keytool -genkey -dname "CN=jras.prd.opsware.services.ubs.com, O=UBS
AG, C=SW" -alias ubsjrascert -storepass bran507025 -keypass
bran507025 -keystore
%iconclude_home%\RAS\Java\Default\webapp\conf\ras_keystore.jks -
keyalg "RSA" -validity 1095
```

   Note: The value for CN must match the name that you will use in the URL, which is not necessarily the hostname of the server.

3. Generate the CSR:

```
>keytool -certreq -alias ubsjrascert -file ubsjrascert.csr -keypass
bran507025 -keystore
%iconclude_home%\RAS\Java\Default\webapp\conf\ras_keystore.jks -
storepass bran507025 -keyalg "RSA"
```

4. Import into the ras_keystore the certificates for the rootCA, intermediate CAs, and the server:

   If prompted to trust the certificate, type **yes** and press enter.

```
>keytool -import -keystore
%iconclude_home%\ras\Java\Default\webapp\conf\ras_keystore.jks -
storepass bran507025 -trustcacerts -file ubsrootca.cer -alias
ubsrootca
>keytool -import -keystore
%iconclude_home%\ras\Java\Default\webapp\conf\ras_keystore.jks -
storepass bran507025 -trustcacerts -file ubsint1ca.cer -alias
ubsint1ca
>keytool -import -keystore
%iconclude_home%\ras\Java\Default\webapp\conf\ras_keystore.jks -
storepass bran507025 -trustcacerts -file ubsint2ca.cer -alias
ubsint2ca
>keytool -import -keystore
%iconclude_home%\ras\Java\Default\webapp\conf\ras_keystore.jks -
storepass bran507025 -trustcacerts -file ubscentralssl.cer -alias
ubscentralssl
```

5. Verify that the certificates are in the store:

```
>keytool -list -keystore
%iconclude_home%\ras\Java\Default\webapp\conf\ras_keystore.jks -
storepass bran507025
```

The output should look like the following:

```
Your keystore contains 5 entries
ubsrootca, Apr 10, 2007, trustedCertEntry,
…
ubscentralssl, May 9, 2007, keyEntry,
…
1, Jun 23, 2006, trustedCertEntry,
…
ubsint2ca, May 9, 2007, trustedCertEntry,
…
ubsint1ca, May 9, 2007, trustedCertEntry,
…
```

6. Import the response:

```
>keytool -import -trustcacerts -alias ubsjrascert -file
jrascert_response.cer -keystore
%iconclude_home%\RAS\Java\Default\webapp\conf\ras_keystore.jks -
storepass bran507025
```

7. Export the public certificate:

```
>keytool -export -alias ubsjrascert -file ubsjrascert.cer -keystore
%iconclude_home%\RAS\Java\Default\webapp\conf\ras_keystore.jks -
storepass bran507025
```

8. Import that certificate into the central key store:

```
>keytool.exe -storepass bran507025 -keystore rc_keystore -import -
file ubsjrascert.cer -alias ubsjrascert
```

Next, you'll replace the Studio certificate, if necessary.

Studio certificate replacement`

If Studio is installed on the same machine as Central, then by default it uses the same keystore as Central (%ICONCLUDE_HOME%\Central\conf\rc_keystore). In this scenario, there is nothing further you need to do to get Studio to function in the environment.

If, however, Studio is installed on its own machine, you will need to import the certificate for the root CA into Studio's keystore. In most but not all organizations, all of the certificates that are issued to servers in the same environment share the same root CA server. Because this is not always true, the first step in replacing the Studio certificate is to import the public certificates for each root CA.

To replace the Studio certificate

- Import the public certificates for each root CA.

```
>keytool -import -file UBSRootCA.cer -alias UBSRootCA -keystore
rc_keystore -storepass bran507025
```

If you are using a self-signed certificate that did not originate from a CA in the HP OO environment, import that public, self-signed certificate as well, using a command like the preceding.

Testing the certificates

To test the certificates

- Open a web browser to *https://central.prd.iconclude.services.ubs.com:8443/PAS/*

  If you are not prompted to trust the certificate, this Central certificate import process was successful.

Configuring HP OO for extended functionality

Extended functionality in HP OO is the use of flows that can execute actions:

- On machines that are on different domains or on the other side of firewalls from the Central Web server (the machine on which you installed the Central Web application).
- That use other Web services or application programming interfaces (APIs).

Such actions are carried out by Remote Action Service (RAS) operations. RAS operations are enabled, or hosted, by the RAS Web service, which is installed during the Central Web application installation.

A RAS operation therefore requires a reference that directs it to RAS. The reference, which you configure in Studio, is made up of a name and the URL of the RAS. You also must add the reference in the RAS operation. (For information on adding a RAS reference in a RAS operation, see Help for Studio.)

There are two considerations that may affect how you install RAS.

- Where you need to install RAS and its content.

  The Central installation installs RAS on the Central server. However, to run an operation against a machine that is on a different domain or the other side of a firewall from the Web server, RAS must be installed on the machine against which you're going to run the operation.

- Which applications you will run the operation against.

  The following applications have special additional requirements:

  - Microsoft Operations Manager (MOM)

    Operations that run against MOM can only run on a MOM server and require a RAS installation on the MOM server to integrate with MOM.

  - Microsoft Exchange Server

    Operations that run against Exchange Server can only run on a machine that has the Exchange Server management tools and a RAS installed.

  - Windows Server Clustering Services

Operations that run against Clustering Services can only run on an Enterprise Edition Windows 2003 Server or a machine that has the Windows 2003 Server Administrator Pack installed. These operations require RAS to integrate with Clustering Services. The RAS must be installed on the Windows Server that is running the Clustering Services.

- HP OpenView

  Operations that run against HP OpenView can only run on a machine that is running HP OpenView and has RAS installed to integrate with HP OpenView.

These applications require special content (IAction code), which is installed by the RAS content-upgrade program RASContentSetup.exe.

The RAS content-upgrade program requires RAS that is installed by the standalone installation program RASSetup.exe.

Therefore, after you install the Central Web server, if you do not install a standalone RAS, you can run operations that:

- Run against machines on the Central Web server's domain (and are not on the other side of a firewall from the Central Web server).
- Do not require support for MOM, Exchange Server, Clustering Services, or HP OpenView.

On the other hand, you need to install RAS and its content-upgrade program in order to run an operation against:

- A machine that is on a different domain or on the other side of a firewall from the Central Web server.

  You only need to install RAS on one machine on the other domain or on the far side of the firewall in order to run a RAS-dependent operation on other machines there.

- MOM, Exchange Server, Clustering Services, or HP OpenView.

The following table summarizes this discussion.

If the operation you want to run	**Then you need to run these installation programs**
Does not require RAS.	Nothing beyond the Central installation
Requires RAS. Runs within the local installation of HP OO. Does not run against applications that require special RAS IAction content.	Nothing beyond the Central installation, because the operation can use the RAS content that was installed as part of the Central installation
Runs against a machine on a different domain or across a firewall from the Central Web server. Does not run against applications that require special RAS IAction content.	The following, run on the machine against which you will run the operation: RASSetup.exe
Runs within the local installation of HP OO.	The following, run locally: RASSetup.exe and RASContentSetup.exe

Runs against applications that require special RAS IAction content.	
Runs against a machine on a different domain or across a firewall from the Central Web server. Runs against applications that require special RAS IAction content.	The following, run on the machine against which you will run the operation: RASSetup.exe and RASContentSetup.exe

For information on installing RAS and RAS content and testing the installations, see the Installation Guide (or, on a Linux machine, the linux.README.txt file for RAS).

Changing Central configurations

Central configurations that you can change include:

- Which authentication providers are enabled and specific settings for how HP OO uses them.

  HP OO supports the following authentication providers:

  - Active Directory (AD)
  - Lightweight Directory Access Protocol (LDAP)
  - Kerberos

  For information on enabling authentication with one or more of these providers, see Help for Central. (Because topic names can change, search for "external authentication".)
- The maximum size of the Jetty service Wrapper.log file.

  If you install Central as a Windows service, then by default the maximum size for Wrapper.log is 64 megabytes (MB). When the file reaches that size, the file begins to *roll*—that is, the oldest entry is deleted as each new entry is added. For information on changing the maximum size of Wrapper.log, see the procedure, "To change the maximum size of the Jetty service Wrapper.log.
- Enabling flows started by Rsflowinvoke.exe to run without a new login

Changing the maximum size of the Wrapper.log file

To change the maximum size of the Jetty service Wrapper.log file

1. In the Jetty home directory, navigate to \extra\win32 and then open wrapper.conf.

   If you accepted the defaults in the Central installation program, the Jetty home directory is a subdirectory of the HP OO home directory.
2. Locate the property "wrapper.logfile.maxsize" and specify the maximum size in bytes that the log file should reach before it starts rolling.

   You can abbreviate the size value of this property by adding k (for kilobytes) to the end of the size.

Important: Setting the value to zero (0) disables rolling, and the file will grow indefinitely.

3. Save and close the file.

Enabling single sign-on for flows started with the Java Flow Invoke tool

You can obtain security and performance benefits by configuring Central so that flows that are started from the Java version of the flow invocation tool (JRSFlowInvoke.jar) use the credentials of the person who is already logged on the machine. This is called *single sign-on (SSO).*

Note: SSO support in Central is based on the standard Kerberos 5. The procedures for enabling single sign-on for Central vary depending on whether Central is to use a Linux key distribution center (KDC) or a Windows KDC (Active Directory, which supports the Kerberos 5 specification). These procedures are documented in the following two sections, which assume that the reader is familiar with Kerberos fundamentals, that is, terms such as principal, ticket, realm, KDC and keytab.

Enabling single sign-on using Windows AD

To track an example through the following procedure, we'll assume the following:

- Central (either Windows or Linux) is located at alamo.mydomain.com
- The Windows AD domain controller is at mydomain.com
- The realm is MYDOMAIN.COM (note that for Windows AD, the realm name is usually the domain name, upper-cased).
- The account for which SSO is attempted is "jdoe".
- The HP OO home directory is represented as "PAS_HOME" in discussion and in commands.

To enable single sign-on using Windows AD

1. Add an AD account for the host (the Central server that the Java flow invocation tool will point at when running the flows). The account must have the following format:

   `HTTP/<server_name.domain_name>`

   It is advisable to configure this AD account with the settings "Password never expires" and "Use DES encryption types for this account".

   If you do not set DES encryption types for the account, AD uses the RC4-HMAC encryption type.

   Using our example, the account that you add would be:

   HTTP/alamo.mydomain.com

2. On the domain controller machine, open a command-line window and generate a keytab file, using the following command:

```
ktpass -out <server_name>.keytab -princ
<service_name>/<server_name.domain_name>@<REALM_NAME> -mapuser
```

```
<service_name>/<server_name.domain_name> -pass *** -crypto DES-CBC-
MD5 -ptype KRB5_NT_PRINCIPAL
```

where:

*** is the password that you specified when you created the above AD account.

In our example, this command would look like this:

```
ktpass -out alamo.keytab -princ HTTP/alamo.mydomain.com@MYDOMAIN.COM
-mapuser HTTP/alamo.mydomain.com -pass *** -crypto DES-CBC-MD5 -
ptype KRB5_NT_PRINCIPAL
```

Copy the keytab file (alamo.keytab in our example) to the Central server, into PAS_HOME/Central/conf directory.

3. Open PAS_HOME/Central/conf/jaasLogin.conf in a text editor.

4. Add the following "com.sun.security.jgss.accept" section after the DharmaKrb5JAAS section, replacing PAS_HOME in the highlighted section with the correct path:

```
DharmaKrb5JAAS {
   com.sun.security.auth.module.Krb5LoginModule required
               refreshKrb5Config=true;
};

com.sun.security.jgss.accept {
        com.sun.security.auth.module.Krb5LoginModule
         required
         storeKey=true
         doNotPrompt=true
         useKeyTab=true
         kdc=mydomain.com
         keyTab="PAS_HOME/Central/conf/alamo.keytab"
         realm="MYDOMAIN.COM"
         principal="HTTP/alamo.mydomain.com@MYDOMAIN.COM"
         debug=true;
};
```

5. In Central/conf, create a krb5.conf file that includes definition of the default realm and KDC (or make sure that the existing krb5.conf includes that information).

In our example, a minimal krb5.conf file would look like this:

```
[libdefaults]
        default_realm = MYDOMAIN.COM
        ticket_lifetime = 24000


[realms]
    MYDOMAIN.COM = {
        kdc = mydomain.com
        admin_server = mydomain.com
        default_domain = .mydomain.com
```

```
    }

[domain_realm]
    .mydomain.com = MYDOMAIN.COM
    mydomain.com = MYDOMAIN.COM

[pam]
    debug = true
```

6. Log in to Central and, on the **Administration** tab, click the **System Configuration** subtab.

7. Scroll down to **Kerberos Authentication Settings** and configure the location for the Kerberos 5 configuration file (krb5.conf) to point to "/Central/conf/krb5.conf".

   Notes:

   - Do not set a realm or a KDC on that page, because Central will now obtain them from the krb5.conf file.
   - You do not need to enable Kerberos authentication unless that is used for logging in.

8. Save your changes, and then restart Central.

   By default, under PAS_HOME/tools (where the java flow invocation tool JRSFlowInvoke.jar is installed) there is an sso_invoke_krb5.conf.sample file that looks like the following:

```
[libdefaults]
        default_realm = MYDOMAIN.COM
        ticket_lifetime = 24000


[realms]
    MYDOMAIN.COM = {
        kdc = mydomain.com
        admin_server = mydomain.com
        default_domain = .mydomain.com
    }

[domain_realm]
    .mydomain.com = MYDOMAIN.COM
    mydomain.com = MYDOMAIN.COM

[pam]
    debug = true
```

9. Copy sso_invoke_krb5.conf.sample to sso_invoke_krb5.conf and edit the latter to match your domain, realm, and KDC.

   By default, under PAS_HOME/Central/tools there is an sso_invoke.bat file for the Windows Central version (or sso_invoke.sh for the Linux Central version) that shows how to use the java flow invocation tool in single sign-on mode. You can run those shell scripts from that location. Or, if the invocation tool is to be used

from a different machine than the Central server, copy the JRSFlowInvoke.jar, sso_invoke.bat (or sso_invoke.sh), and sso_invoke_krb5.conf files to that machine and adjust the paths (including the path to JRE 1.6, which is required on the target machine—you can obtain JRE 1.6 from the downloads page of the Java site, *http://java.sun.com/*).

You can invoke the shell scripts with a command such as the following:

```
sso_invoke alamo.mydomain.com:8443 /Library/MyFlows/myFlow
```

Important: If the path to the flow includes spaces, you must replace the spaces with the `%20` character sequence, which is how spaces are encoded in URLs. For example, to start the flow **myflow** in the directory **\Library\My Ops Flows**, the command to launch the flow using sso_invoke.bat would be:

```
sso_invoke.bat concord.battleground.ad:8443
\Library\My%20Ops%20Flows\myflow
```

10. Log in to Central with an account that has Administrator rights.

    Next, you will need to give HEADLESS_FLOWS capability to the SSO users.

11. The easiest way to give HEADLESS_FLOWS capability to the SSO users is:

    a. In Central, on the **Administration** tab, click the **System Configuration** sub-tab.

    b. Scroll to the Kerberos section and set the default group to a group that has HEADLESS_FLOWS capability.

       This way, any headless invocation using SSO will have the capabilities of that group (flows cannot be invoked using the headless tool unless the user under whose credentials the invocation happens, has HEADLESS_FLOWS capability).

    Or, if SSO flow invocations need to be controlled on a user-by-user basis:

    a. On the **Administration** tab, create the Central user that matches the account under which the SSO flow invocation will happen ("jdoe" in our example) and specify that it is an external user.

       For information on how to create a user and specify that it is an external user, see Help for Central.

       The user must be a member of a group that has HEADLESS_FLOWS capability; without this capability, the user will not be able to start runs using SSO flow invocation.

    In addition to having the HEADLESS_FLOWS capability, the user under whose credentials the SSO flow invocation happens needs to have **read** and **execute** permissions for the flow and the operations that the flow uses. For more information on granting permissions to flows and operations see Help for Studio.

12. If the SSO java invocation is from a Linux machine that is not configured to obtain Kerberos tickets automatically, obtain a forwardable ticket from the Windows domain controller (you might have to change /etc/krb5.conf to point it to the Windows domain controller), using a command like the following:

```
kinit -f jdoe@MYDOMAIN.COM
```

13. If Central is a Windows version hosted on a Windows 2000/2003 system, add the following registry key (do the same for the machine where the java invocation tool is invoked from, if the machine is Windows 2000/2003):

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters
Value Name: allowtgtsessionkey
Value Type: REG_DWORD
Value: 0x01
```

Enabling single sign-on using MIT KDC

Note: The following procedure assumes that the system uses a Linux version of MIT KDC.

To track an example through the following procedure, we'll assume the following:

- Central (either Windows or Linux) is located at fitzroy.mydomain.com
- KDC is at kdc.mydomain.com
- The realm is MYDOMAIN.COM.
- The account for which SSO is attempted is "jdoe".
- The HP OO home directory is represented as "PAS_HOME" in discussion and in commands.

To enable single sign-on using MIT KDC

4. On the KDC machine, add a service principal for *HTTP/fitzroy.mydomain.com@MYDOMAIN.COM* using the kadmin's `addprinc` command (for information on using kadmin, see the man pages for kadmin):
   ```
   kadmin: addprinc –randkey HTTP/fitzroy.mydomain.com@MYDOMAIN.COM
   ```
5. Export the principal you just created to fitzroy.keytab:
   ```
   kadmin: ktadd –k fitzroy.keytab HTTP/fitzroy.mydomain.com
   ```
6. Copy the keytab file to the Central machine at PAS_HOME/Central/conf
7. In Central/conf, create a krb5.conf file that includes definition of the default realm and KDC (or make sure that the existing krb5.conf includes that information).

   In our example, a minimal krb5.conf file would look like this:

```
[libdefaults]
        default_realm = MYDOMAIN.COM
        ticket_lifetime = 24000
        default_tkt_enctypes =  des3-cbc-sha1

[realms]
    MYDOMAIN.COM = {
        kdc = kdc.mydomain.com
        admin_server = kdc.mydomain.com
        default_domain = mydomain.com
    }


[domain_realm]
```

```
    .mydomain.com = MYDOMAIN.COM
    mydomain.com = MYDOMAIN.COM

[pam]
    debug = true
```

8. Open /Central/conf/jaasLogin.conf in a text editor.
9. Add the following "com.sun.security.jgss.accept" section after the DharmaKrb5JAAS section, replacing PAS_HOME with the correct path:

```
DharmaKrb5JAAS {
   com.sun.security.auth.module.Krb5LoginModule required
               refreshKrb5Config=true;
};

com.sun.security.jgss.accept {
        com.sun.security.auth.module.Krb5LoginModule
         required
         storeKey=true
         doNotPrompt=true
         useKeyTab=true
         kdc=kdc.mydomain.com
         keyTab="PAS_HOME/Central/conf/fitzroy.keytab"
         realm="MYDOMAIN.COM"
         principal="HTTP/fitzroy.mydomain.com@MYDOMAIN.COM"
         debug=true;
};
```

10. Log in to Central and on the **Administration** tab, click the **System Configuration** subtab.
11. Scroll down to **Kerberos Authentication Settings** and configure the location for the Kerberos 5 configuration file (krb5.conf) to point to "/Central/conf/krb5.conf".

    Notes:

    - Do not set a realm or a KDC on that page, because Central will now obtain them from the krb5.conf file.
    - You do not need to enable Kerberos authentication unless that is used for logging in.
12. Save your changes, and then restart Central.

    By default, under PAS_HOME/tools (where the java flow invocation tool JRSFlowInvoke.jar, is by default installed) there is an sso_invoke_krb5.conf.sample file that looks like:

```
[libdefaults]
        default_realm = MYDOMAIN.COM
        ticket_lifetime = 24000
        default_tkt_enctypes =  des3-cbc-sha1

[realms]
```

```
    MYDOMAIN.COM = {
        kdc = mydomain.com
        admin_server = mydomain.com
        default_domain = .mydomain.com
    }

[domain_realm]
    .mydomain.com = MYDOMAIN.COM
    mydomain.com = MYDOMAIN.COM

[pam]
    debug = true
```

13. Copy sso_invoke_krb5.conf.sample to sso_invoke_krb5.conf and edit the latter to match your domain, realm and KDC.

    By default, under PAS_HOME/tools there is an sso_invoke.bat file for the Windows Central version (or sso_invoke.sh for the Linux Central version) that shows how to use the java flow invocation tool in single sign-on mode. You can run those shell scripts from that location. Or, if the invocation tool is to be used from a different machine than the Central server, copy the JRSFlowInvoke.jar, sso_invoke.bat (or sso_invoke.sh), and sso_invoke_krb5.conf files to that machine and adjust the paths (including the path to JRE 1.6, which is required on the target machine—you can obtain JRE 1.6 from the downloads page of the Java site, *http://java.sun.com/*).

    The shell scripts can be invoked with a command such as in the following:

    ```
    sso_invoke fitzroy.mydomain.com:8443 /Library/MyFlows/myFlow
    ```

    Important: If the path to the flow includes spaces, you must replace the spaces with the `%20` character sequence, which is how spaces are encoded in URLs. For example, to start the flow **myflow** in the directory **/Library/My Ops Flows**, the command to launch the flow using sso_invoke.bat would be:

    ```
    sso_invoke.bat concord.battleground.ad:8443
    /Library/My%20Ops%20Flows/myflow
    ```

14. Log in to Central with an account that has Administrator rights.

    Next, you will need to give HEADLESS_FLOWS capability to the SSO users.

15. The easiest way to give HEADLESS_FLOWS capability to the SSO users is:

    b. In Central, on the **Administration** tab, click the **System Configuration** sub-tab.

    c. Scroll to the Kerberos section and set the default group to a group that has HEADLESS_FLOWS capability.

       This way, any headless invocation using SSO will have the capabilities of that group (flows cannot be invoked using the headless tool unless the user under whose credentials the invocation happens, has HEADLESS_FLOWS capability).

    Or, if SSO flow invocations need to be controlled on a user-by-user basis:

    - On the **Administration** tab, create the Central user that matches the account under which the SSO flow invocation will happen ("jdoe" in our example) and specify that it is an external user.

For information on how to create a user and specify that it is an external user, see Help for Central.

The user must be a member of a group that has HEADLESS_FLOWS capability; without this capability, the user will not be able to start runs using SSO flow invocation.

a. In addition to having the HEADLESS_FLOWS capability, the user under whose credentials the SSO flow invocation happens needs to have **read** and **execute** permissions for the flow and the operations that the flow uses. For more information on granting permissions to flows and operations see Help for Studio.

16. If the SSO flow invocation is from a Linux machine that is not configured to obtain Kerberos tickets automatically, obtain a forwardable ticket from the KDC (you might have to change /etc/krb5.conf to point it to the kdc.mydomain.com in our example), using a command like the following:

```
kinit –f jdoe@MYDOMAIN.COM
```

17. If the SSO flow invocation is from a Windows machine, a forward-able ticket needs to be obtained from the Linux MIT KDC. This can be done by using kinit executable under PAS_HOME/jre1.6/bin.
18. If the SSO flow invocation is from a Windows 2000/2003 system, add the following registry :

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters
Value Name: allowtgtsessionkey
Value Type: REG_DWORD
Value: 0x01
```

Enabling SSO when a network load balancer is used

The procedure is the same as in the above sections, the only change being that the service principal and keytab files are generated for the network load-balancer (NLB) machine and not for the individual Central nodes behind the load balancer.

For example, suppose that:

- The NLB machine is nlb.mydomain.com
- There are two Central nodes behind the load balancer: central1.mydomain.com and central2.mydomain.com

In this case, the service principal would be *HTTP/nlb.mydomain.com@MYDOMAIN.COM* (if Windows AD is used, the AD user account would be HTTP/nlb.mydomain.com), and the keytab file would be nlb.keytab.

In addition, you must:

- Copy the keytab to central1.mydomain.com and central2.mydomain.com.
- Modify the respective entries in jaasLogin.conf on those machines to point to keytab=nlb.keytab and principal=*HTTP/nlb.mydomain.com@MYDOMAIN.COM*

When you call the SSO flow invocation script, make sure that it points to nlb.mydomain.com, as in the following:

```
sso_invoke nlb.mydomain.com:<port_number> /Library/MyFlows/myFlow
```

where `<port_number>` is the port on which the network load balancer is listening.

Important: If the path to the flow includes spaces, you must replace the spaces with the `%20` character sequence, which is how spaces are encoded in URLs. For example, to start the flow **myflow** in the directory **/Library/My Ops Flows**, the command to launch the flow using sso_invoke.bat would be:

```
sso_invoke nlb.mydomain.com:<port_number>
/Library/My%20Ops%20Flows/myFlow
```

Changing the network address binding of a Central server with multiple network interfaces

To bind a Central server to a network address

1. In the Central home directory, open the \conf subdirectory.
2. Locate the Central.properties file and. open it in a text editor.
3. Locate the following lines:

   ```
   dharma.jgroups.prop.udp=UDP(down_thread=false;mcast_send_buf_size=64
   0000;mcast_port=${clustering.mcast_port:45566};\
   discard_incompatible_packets=true;ucast_recv_buf_size=20000000;mcast
   _addr=${clustering.mcast_addr:228.10.10.10};\
   up_thread=false;loopback=false;mcast_recv_buf_size=25000000;max_bund
   le_size=64000;\
   max_bundle_timeout=30;use_incoming_packet_handler=true;use_outgoing_
   packet_handler=false;\
   ucast_send_buf_size=640000;tos=16;enable_bundling=true;ip_ttl=2)
   ```

4. Immediately preceding the end parenthesis, add

   ```
   ;bind_addr=<IPADDRESS>
   ```

   where `<IP_ADDRESS>` is the IP address of the network interface that identifies the Central to other machines on the network.

   For example, to bind the server to the IP address 255.255.0.225, you would change the lines to read as follows:

   ```
   dharma.jgroups.prop.udp=UDP(down_thread=false;mcast_send_buf_size=64
   0000;mcast_port=${clustering.mcast_port:45566};\
   discard_incompatible_packets=true;ucast_recv_buf_size=20000000;mcast
   _addr=${clustering.mcast_addr:228.10.10.10};\
   up_thread=false;loopback=false;mcast_recv_buf_size=25000000;max_bund
   le_size=64000;\
   max_bundle_timeout=30;use_incoming_packet_handler=true;use_outgoing_
   packet_handler=false;\
   ucast_send_buf_size=640000;tos=16;enable_bundling=true;ip_ttl=2;bind
   _addr=255.255.0.225)
   ```

5. Save your changes and close Central.properties.

Enabling and disabling run-scheduling concurrency for Scheduler

It is now possible to have multiple runs of the same flow running at the same time. This means that you can start multiple runs of the same flow and target them to different servers, scheduling them to all start at the same time or to start a second run of the flow before the first one ends.

Important: Suppose, however, that you schedule a flow such as a health check, two run twice against the same server, separating the two flows by a certain period of time. If one of the runs goes beyond the start time of the health check's next scheduled run, then the execution of the second run can interfere with the execution of the flow in the first run.

Because it is possible to schedule concurrent runs of flows, you need be aware of the possible interactions between concurrent runs of a flow.

In some situations, you may wish to disable run-scheduling concurrency (by default, this capacity is enabled). You do so in the Scheduler's schedule.properties file.

Important: When you enable or disable run-scheduling concurrency, any existing schedules are not affected. That is, any schedules that you create to run concurrently continue to be able to run concurrently without blocking each other even after you have disabled the capacity.

To enable/disable run-scheduling concurrency

1. In the HP OO home directory, find and open the Scheduler\conf\scheduler.properties file in a text editor.
2. To disable the capacity to schedule concurrent runs of the same flow, find the following line, and change "true" to "false".

   ```
   dharma.scheduler.nonBlockingFlows=true
   ```

   OR

   If the capacity is currently disabled and you want to enable it, change "false" to "true".

Configuring maximum concurrent runs

By default, the maximum number of runs that the Central server can run at the same time is 600. If you choose to increase this, you may need also to increase the maximum amount of read-only memory allotted to the HP OO server's Central process. For information on doing so, see *Configuring maximum RAM for HP OO server process*. Remember also that increasing the number of runs that Central can execute simultaneously beyond the capabilities of your system, can affect performance.

To configure the maximum number of concurrent runs

1. In the HP OO home directory, navigate to Central\conf and open Central.properties.
2. In the line "dharma.executor.masConcurrentRuns=-600", change "600" to the greatest number of runs that your system should run at one time.
3. Save and close the file.

Configuring maximum RAM for HP OO server process

You can optimize performance of Central by configuring the maximum amount of random-access memory (RAM) that is allowed to the Central process. This value is not pre-allocated when the Jetty service starts.

To configure maximum RAM for the Central process

1. In the HP OO home directory, open the jetty subdirectory, and then open Start_jetty.bat for editing.
2. In the line "SET ICONCLUDE_MEM_OPTS=-Xmx1024m", change "1024" to a value that represents a desirable maximum amount of memory for the service.

   Be sure to append "m" to the value to specify that the value represents an amount in megabytes. (If you do not append the "m", the maximum memory is specified as bytes.)
3. In the HP OO home directory, navigate to Central\conf and open Wrapper.conf.
4. Change the line "wrapper.java.maxmemory=1024" to a value that represents a desirable maximum amount of memory for the service.

   In Wrapper.conf, you can leave 'm' off the end of the value, because this value always represents megabytes.

   These two values (in Start_jetty.bat and in Wrapper.conf), should be the same, but they do not have to be. Start_jetty.bat starts Central as a standalone, command-line process, and wrapper.conf controls Central when started as a Win32 service.

Changing the timeout limit for RAS operations

RAS operations are subject to a default timeout limit of 20 minutes on Central for remote RAS operations. To support RAS operations that are likely to take more than 20 minutes to complete, you can change Central's default timeout setting.

To change the timeout setting for a remote RAS installation

1. In the HP OO home directory, navigate to \Central\conf\ and open the wrapper.conf file for editing.
2. Add the following line to the file:

   `-Dras.client.timeout=<timeout in seconds>`

   where `<timeout in seconds>` is the amount of time you want the RAS to wait before timing out.

   This line overrides the default timeout value for RAS operations.

SSH operations also have their own timeout settings of 90 seconds. When you use a (copy of a) SSH operation from the Studio Library Operations folder, you can change the timeout setting by changing the value for the **Timeout** input of the operation.

Changing Studio configurations

In the \iConclude\conf\Studio.properties file, you can change the following aspects of the Studio:

- Central host server
- Default communication port used
- Choice of HTTP: or HTTPS: (secure sockets) as the Internet protocol

To change the Studio.properties file

3. Use a text editor to open %ICONCLUDE_HOME%\Studio\conf\Studio.properties
4. Edit the following lines to make the desired changes:
   - To change the name of the host server (the server on which the Web application is located), change **localhost** in the following line to the name or IP address of the host server.
     `dharma.repaircenter.host=localhost`
   - To change the port number that HP OO uses, change **8080** in the following line to the desired port number.
     `dharma.repaircenter.port=8080`
   - By default, HP OO uses the https Internet protocol. To specify that HP OO use the http Internet protocol, change **https** in the following line to **http**.
     `dharma.repaircenter.proto=https`

Configuring log file settings

HP OO records errors (ERROR), warnings (WARN), information (INFO), and debugging messages (DEBUG) in the following log files:

- For Studio: iConclude.log, in the \Studio\logs subdirectory of the HP OO home directory
- For Central: Wrapper.log, in the \Central\logs subdirectory of the HP OO home directory

Because logging activity can slow HP OO's performance and create very large log files, it is important that HP OO run with appropriate logging levels. The default logging levels have been set to provide necessary information without impacting performance. It is recommended that you use the default logging levels.

To change logging levels

5. In the jetty\resources subdirectory of the HP OO home directory, modify the log4j.properties file according to your needs.
6. Save changes.

Backing up HP OO

Backing up HP OO involves backing up your flows, operations, system accounts, selection lists, and other HP OO objects, and backing up the HP OO database. You

back up HP OO objects in Studio by backing up the repository and then placing a copy of the repository's backup in a secure location.

To back up HP OO

1. In Studio, back up each repository (**Create Backup** command, on the **Repository** menu), using the procedure given in Help for Studio.

   Each repository is backed up as a .jar file.
2. Make a copy of each repository's .jar file and store the copy in a secure location.
3. Back up the Central database and store the backup in a secure location.

   Dashboard charts are stored in the Central database, so the database backup includes Dashboard charts.

Supporting a Central server cluster

You can create failover, load-balancing, and/or run recovery support by installing Central on several servers and creating one or both of the following kinds of clusters:

- Load balancing

  You can provide this with the HP OO Load Balancer. For high availability, you can also provide failover clustering support for the HP OO Load Balancer.
- Failover and run recovery

  To provide failover and run recovery, you configure the Central.properties file in each of the Central servers.
- For Central database clustering, you can use third-party software of your choice.

For information on installing the Load Balancer and on configuring the Central.properties file, see the *HP OO Installation Guide* (InstallGuide.pdf).

Re-configuring the Central server cluster

When clustering is enabled, Central/JGroups listens on UDP port 7500. However, once you have created the cluster, it is safe to release port 7500 for other uses. To do so, perform the

To release port 7500

1. On each Central server in the cluster, in the HP OO home directory, navigate to the \Central\conf subdirectory.
2. Open the Central.properties file in a text editor.
3. Find the line that begins with `dharma.jgroups.prop.udp=UDP`
4. Insert `enable_diagnostics=false` as the first item inside the parentheses that follow that line beginning. Leave the other items as they are.

   After you have done so, the line should read as follows:

   ```
   dharma.jgroups.prop.udp=UDP(enable_diagnostics=false;…)
   ```

   where the … represents additional items.
5. Find the line that begins with `dharma.jgroups.prop.tcp=TCP`

6. Insert `enable_diagnostics=false` as the first item inside the parentheses that follow that line beginning. Leave the other items as they are.

   After you have done so, the line should read as follows:

   ```
   dharma.jgroups.prop.tcp=TCP(enable_diagnostics=false;…)
   ```

   where the … represents additional items.

7. Restart Central.
8. To verify that port 7500 is not in use, open a command window and run the following command:

   ```
   netstat -an| find "7500" port
   ```

Administering a Central server cluster

Administering an HP OO Load Balancer cluster involves:

- Adding nodes to and removing them from the cluster. For information on how to start the Load Balancer for configuring, see the *HP OO Installation Guide.*

Administering a Central failover/run recovery cluster involves:

- Adding nodes to and removing them from the cluster.

  For information on adding a node to a Central failover cluster (whether the cluster uses IP multicasting or TCP ping for internal communication), see the *HP OO Installation Guide.*

  Important: When you add a node to a cluster whose nodes use TCP ping to communicate, you must add the node in the JGroups list in each node's Central.properties file (see the HP OO Installation Guide (InstallGuide.pdf).

- Maintaining the consistency of the repository across the cluster.

  You can use the Publish Staging to Production Clusters flow to replicate a repository across a cluster. You must provide the flow with the URL for the staging server and the URL for one of the cluster nodes. With just one cluster node URL supplied, the flow discovers the rest of the nodes in the cluster and iterates through them, publishing the repository to each one.

  Although best practice is to have a staging Central server and publish the repository from there to the Central cluster in the production environment, you can run the flow from one of the nodes in the cluster.

Index