Opsware™ Process Automation System

Version 7.0, Administering Opsware Process Automation System

For Process Automation System Administrators

Table of Contents

Updating documentation

Documentation enhancements are a continual project at Opsware. You can update the documentation set at any time using the following procedure (which is also available in the PAS readme file).

To obtain PAS documentation

1. On The Opsware Network Web site (https://support1.opsware.com/support/index.php), log in with the Opsware Network account name and password that you received when you purchased PAS.
2. On the **Support** tab, click the **Product Docs** subtab.
3. Under **Quick Jump**, click **Process Automation System**.
4. Under **Process Automation System**, click **ZIP** beside **PAS 7.0 Full Documentation Set**.
5. Extract the files in the .zip file to the appropriate locations on your system:
   - For the tutorials to run, you must store the .swf file and the .html file in the same directory.
   - To obtain the repository that reflects the state of the flow at the start of the tutorial, unzip the file Exportof<preceding_tutorial_name>.zip.
   - To obtain the scriptlet for the tutorial that includes using scriptlets, click the scriptlet .txt file name.
   - To update your Central or Studio Help:
     a. Under **Help Files**, and then click **Studio Help File Bundle** or **Central Help File Bundle**.
     b. In the **File Download** box appears, click either **Open** or **Save**.
     c. Extract the files to the Opsware\PAS home directory, in either the **\Central\docs\help\Central** or **\Studio\docs\help\Studio** subdirectory, overwriting the existing file.

Documentation of administrative tasks

Some administrative tasks are performed from within PAS Central and some are performed outside of Central. For instance, you configure and enable external authentication providers on the Administration tab in PAS Central, but making other configuration changes to Central can require work with files outside of the Central Web application. In addition to providing some conceptual and reference material related to PAS administration, this administrator's guide provides procedures for completing the following administrative tasks:

- Configuring Active Directory or LDAP over SSL
- Configuring PAS 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 PAS
- Supporting a Central server cluster

- Administering a Central server cluster

Overview

Administering PAS includes:

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

  You can map PAS user roles either to external or to internal group.

- Enabling PAS to run Ops flows against remote machines and integrate them with other applications. For more information, see Configuring PAS 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 PAS
- Supporting a Central server cluster

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

Useful information

The following information is useful for planning your installation and for various uses, such as installing or configuring PAS components and creating URLs that launch PAS flows.

Default ports used by PAS components

By default, PAS components use the following ports:

- Central: 8443

  If Central servers are clustered, port 45566 is used for SSL communications between JGroups nodes

- Between PAS components, such as Central, Scheduler, JRAS, and NRAS: 18443
- JRAS: 9004
- NRAS: 9005
- Scheduler: 19443

Security: Users, Groups, Capabilities, and Permissions

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

- Security of communications between PAS 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 PAS communications.

- User authentication, or logging in.

  You can configure PAS 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 PAS users and groups and controlling the operations and flows that they can run.

  In PAS, 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.

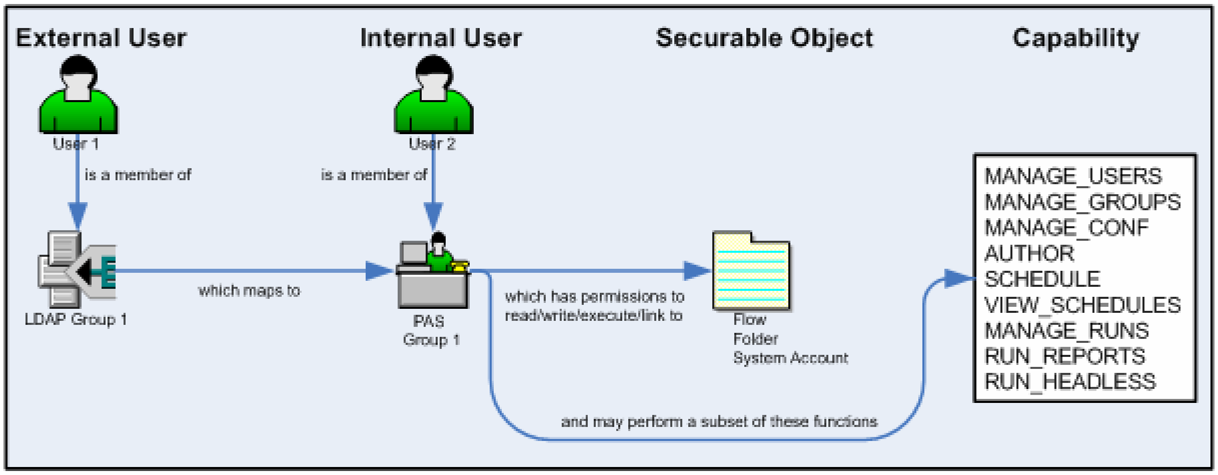

Figure 1 - Components of access control in PAS

- Protecting the confidentiality of sensitive credentials.

  System accounts are PAS 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 PAS 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:

   d. From the **File** menu, select **Add/Remove Snap-in.**

   e. In the **Add/Remove Snap-in** dialog, click **Add.**

   f. In the **Add Standalone Snap-in** dialog, select **Certificates** and click **Add.**

   g. Select **Computer Account** and then click **Next.**

   h. In the **Select Computer** dialog box, click **Finish.**

   i. 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 %PAS_HOME%\jre1.5\bin
    keytool –keystore “%PAS_HOME%\jre1.5\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 “%PAS_HOME%\jre1.5\lib\security\cacerts”
     –list -alias <some_alias>
    ```

    The default certificate store password is “changeit”.

    You should see a summary of the certificate.

16. In %PAS_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.

Configuring PAS for extended functionality

Extended functionality in PAS is the use of Ops 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 RAS operations, which are enabled, or hosted, by one of two Web services that are installed during the PAS Web application installation:

- Java Remote Action Service (JRAS)
- .NET Remote Action Service (NRAS)

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

There are two considerations that may affect how you install JRAS and NRAS.

- Where you need to install JRAS and NRAS and their content.

  The Central installation installs JRAS and NRAS 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, JRAS or NRAS 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 NRAS to integrate with MOM. The instance of NRAS must be installed on the MOM server.

  - Microsoft Exchange Server

    Operations that run against Exchange Server can only run on a machine that has the Exchange Server management tools installed and require NRAS to integrate with Exchange Server. The instance of NRAS must be installed on the Exchange server.

  - 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 NRAS to integrate with Clustering Services. The instance of NRAS 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 require JRAS to integrate with HP OpenView. The instance of JRAS must be installed on the HP OpenView machine.

  These applications require special content (IAction code), which is installed by the JRAS and NRAS content-upgrade programs JRASContentSetup.exe and NRASContentSetup.exe.

The JRAS and NRAS content-upgrade programs require the versions of JRAS and NRAS that are installed by the independent installation programs JRASSetup.exe and NRASSetup.exe. These versions are different from the versions of JRAS and NRAS that are installed by default.

Therefore, after you install the PAS Web server, you can run operations that:

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

On the other hand, you need to install either JRAS and its content-upgrade program or NRAS 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 PAS Web server.

  You only need to install JRAS or NRAS on one machine on the other domain or on the far side of the firewall in order to run a JRAS-dependent or NRAS-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 either JRAS or NRAS.	Nothing beyond the Central installation
Requires either JRAS or NRAS. Runs within the local installation of PAS. Does not run against applications that require special RAS IAction content.	Nothing beyond the Central installation, because the operation can use the NRAS or JRAS content that was installed as part of the Central installation
Runs against a machine on a different domain or across a firewall from the PAS 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: JRASSetup.exe or NRASSetup.exe as appropriate
Runs within the local installation of PAS. Runs against applications that require special RAS IAction content.	The following, run locally: JRASSetup.exe and JRASContentSetup.exe OR NRASSetup.exe and NRASContentSetup.exe as appropriate
Runs against a machine on a different domain or across a firewall from the PAS	The following, run on the machine against which you will run the operation:

Web server. Runs against applications that require special RAS IAction content.	JRASSetup.exe and JRASContentSetup.exe OR NRASSetup.exe and NRASContentSetup.exe as appropriate

The following sections describe installing JRAS and NRAS and their content, and testing the installations.

Changing Central configurations

Central configurations that you can change include:

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

  PAS 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 PAS 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 PAS home directory (which by default is C:\Program Files\Opsware\PAS).

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 PAS 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 a 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
```

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 forward-able 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 to 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 PAS home directory is represented as "PAS_HOME" in discussion and in commands.

To enable single sign-on using MIT KDC

1. 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
   ```
2. Export the the principal you just created to fitzroy.keytab:
   ```
   kadmin: ktadd –k fitzroy.keytab HTTP/fitzroy.mydomain.com
   ```
3. Copy the keytab file to the Central machine at PAS_HOME/Central/conf
4. 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
   ```
5. Open /Central/conf/jaasLogin.conf in a text editor.
6. 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;
};
```

7. Log in to Central and on the **Administration** tab, click the **System Configuration** subtab.
8. 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.
9. 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 a 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
```

10. 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
    ```

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

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

12. 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.

13. If the SSO flow invocation is from a Linux machine that is not configured to obtain Kerberos tickets automatically, obtain a forward-able 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
    ```

14. 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.

15. 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.

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.

Being able to schedule concurrent runs of flows means that 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 PAS 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".

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 %PAS_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 PAS uses, change **8080** in the following line to the desired port number.

     ```
     dharma.repaircenter.port=8080
     ```
   - By default, PAS uses the https Internet protocol. To specify that PAS use the http Internet protocol, change **https** in the following line to **http**.

     ```
     dharma.repaircenter.proto=https
     ```

Backing up PAS

Backing up PAS involves backing up your Ops flows, operations, system accounts, selection lists, and other PAS objects, and backing up the PAS database. You back up PAS 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 PAS

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 PAS Load Balancer. For high availability, you can also provide failover clustering support for the PAS 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 *PAS Installation Guide* (PAS_InstallGuide.pdf).

Administering a Central server cluster

Administering a PAS Load Balancer cluster involves:

- Adding nodes to and removing them from the cluster. For information on how to start the Load Balancer for configuriing, see the *PAS 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 *PAS 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 PAS Installation Guide [PAS_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 URI 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