

J2EE Remote Instance Monitoring

This section includes the following topics:

- [About J2EE remote instance monitoring](#)
- [Preparing J2EE remote instance monitoring](#)
- [Installing the J2EE remote instance](#)
- [Installation on WebSphere](#)
- [Additional information](#)

About J2EE remote instance monitoring

If there are security restrictions that prevent the installation of the Precise Listener on the monitored JVM server, you can monitor J2EE instances remotely without installing the Precise Listener there. In that case, the following features which are available in regular J2EE monitoring, are not available:

- It is not possible to configure the agent using the UI Settings menu.
- The patch mechanism will not work. You will need to deploy the patches manually.
- If the JVM is part of a PeopleSoft application, the PeopleSoft related data is not displayed.
- The Availability indication for remote instances will not differentiate between the `Collector is down` and `JVM is down` status. Both statuses will be reported as `JVM is not loading data`.
- It is not possible to generate a Support file for the monitored instance server.

Remote J2EE monitoring uses an SSL-based Listener as part of the Precise Listener which receives requests from the remote J2EE instances. To enable monitoring of a remote J2EE instance, you need to extract the deployment package on the instance server and modify the instance startup to use this folder.

The data is sent from the remote monitored J2EE instance to the SSL listeners, and then processed by Precise Listeners which are configured as consumers. SSL and Consumer configurations can be applied on one or more Precise Listeners. Consumers can be configured to consume data from all SSL listeners or from a specific one. This flexibility can answer Security and performance issues.

For example:

- **Performance case.** If you have a very large activity amount which is sent to one SSL listener, you can configure several Precise Listeners as consumers to load balance the data processing and the load to the PMDB.
- **Security case.** If the monitored server and the FocalPoint are located in different security zones (the monitored server zone cannot approach to the FocalPoint zone), you can configure an SSL Listener in the monitored server zone, and a consumer Listener in the FocalPoint zone, so that the FocalPoint zone will initiate the communication. The SSL and Consumer configuration can be applied to both the Precise FocalPoint and Precise Listener.

Preparing J2EE remote instance monitoring

The preparation of J2EE remote instance monitoring is described in three parts:

- Installing an SSL Listener
- Installing a consumer Listener
- Creating the deployment package

Perform the part in the order as described.

Installing an SSL Listener

Install a new Precise Listener or edit an existing Precise Listener installation and configure the server to be an SSL Listener.



The SSL Listener can also be configured on the Precise FocalPoint.

The SSL Listener mode is set by the `server-machine-ssl-<SERVER_NAME>` property. The SSL Listener port is set by the `server-machine-ssl-port-<SERVER_NAME>` property. For example, to enable the SSL Listener on server `sun5`, run the following CLI utility command on the Main Precise FocalPoint server:

Windows	infra\bin\psin_cli.bat -i3-user admin -i3-clear-password admin -action setup -parameter setup-process=server -parameter handle-server-machine-name=sun5 -parameter setup-mode=edit -parameter server-machine-ssl-sun5=true -parameter server-machine-ssl-port-sun5=1443
UNIX	infra/bin/psin_cli.sh -i3-user admin -i3-clear-password admin -action setup -parameter setup-process=server -parameter handle-server-machine-name=sun5 -parameter setup-mode=edit -parameter server-machine-ssl-sun5=true -parameter server-machine-ssl-port-sun5=1443

Installing a consumer Listener

Install a new Precise Listener or edit an existing Precise Listener installation and configure the server to be a consumer Listener. The consumer Listener mode is set by the `server-machine-consumer-mode-<SERVER_NAME>` property. For example, to enable the consumer Listener on server `sun5`, run the following CLI utility command on the Main Precise FocalPoint server:

Windows	infra\bin\psin_cli.bat -i3-user admin -i3-clear-password admin -action setup -parameter setup-process=server -parameter handle-server-machine-name=sun5 -parameter setup-mode=edit -parameter server-machine-consumer-mode-sun5=true
UNIX	infra/bin/psin_cli.sh -i3-user admin -i3-clear-password admin -action setup -parameter setup-process=server -parameter handle-server-machine-name=sun5 -parameter setup-mode=edit -parameter server-machine-consumer-mode-sun5=true

The consumer Listener processes the data from the remote J2EE instances. The consumer Listener serves all the J2EE instances. If you have installed many remote J2EE instances and the load on the consumer Listener is high, you can configure additional Precise Listeners to be consumer Listeners.

By default the consumer Listener consumes data from all SSL listeners. If you want to consume data from a specific SSL Listener you can set the `server-machine-consume-target-<SERVER_NAME>` property. For example, to set the consumer Listener on server `sun5` to consume data from SSL Listener `hp2`, run the following CLI utility command on the Main Precise FocalPoint server:

Windows	infra\bin\psin_cli.bat -i3-user admin -i3-clear-password admin -action setup -parameter setup-process=server -parameter handle-server-machine-name=sun5 -parameter setup-mode=edit -parameter server-machine-consumer-mode-sun5=true -parameter server-machine-consume-target-sun5=hp2
UNIX	infra/bin/psin_cli.sh -i3-user admin -i3-clear-password admin -action setup -parameter setup-process=server -parameter handle-server-machine-name=sun5 -parameter setup-mode=edit -parameter server-machine-consumer-mode-sun5=true -parameter server-machine-consume-target-sun5=hp2

Creating the deployment package

To create the deployment package

Run the following CLI utility script to create the `deployment.zip` file that should later be extracted on the remote monitored instance server. The `deployment.zip` file can be created once and there is no need to recreate the `deployment.zip` file for each instance that you want to monitor. This `deployment.zip` file is later used in the installation stage.

Windows `infra\bin\psin_cli.bat`
 `-i3-user <user_name>`
 `{-i3-encrypted-password <encrypted_password> | -i3-clear-password <clear_password>}`
 `-action ssl-deployment`
 `{-server <SSL target server>}`

UNIX `./infra/bin/psin_cli.sh`
 `-i3-user <user_name>`
 `{-i3-encrypted-password <encrypted_password> | -i3-clear-password <clear_password>}`
 `-action ssl-deployment`
 `{-server <SSL target server>}`


Table 4-1 Elements of the deployment command


Element	Description
i3-user	See Authenticate to CLI Utility on page 8.
i3-encrypted-password	See Authenticate to CLI Utility on page 8.
action	Always ssl-deployment. Mandatory: Yes
server	SSL target server name Value: String Mandatory: No, unless you have more than one SSL Listener.

Installing the J2EE remote instance

To monitor a remote J2EE instance, perform the following steps to install the J2EE remote instance:

1. Copy the `deployment.zip` file to the J2EE instance server.
2. Extract it to a local folder.

 Verify that the JVM user has full permissions in the local folder, as the JVM must have access to the folder and files within it to initiate the installation of the J2EE instance. The files also contain the access key to the Precise installation and should only be accessible to the relevant JVM user.

 You should keep one folder per monitored instance on the server.

3. Add the following JVM parameters to the JVM startup script, where `<LOCAL_LIB>` is the extracted directory name.
 - a. For WebSphere, see [Installation on WebSphere](#).
 - b. For anything else:
`-javaagent:<LOCAL_LIB>/products/j2ee/lib/indepthj2ee-agent.jar`
 Alternatively, you can run the `displayParams` script, which will generate the JVM parameter.


 For more information on JVM arguments, see [JVM arguments](#).

4. Restart the JVM to initiate the instance installation.

JVM arguments

This section describes recommended guidelines to follow for the JVM arguments.

- All JVM arguments should appear as a single line.
- The format of the parameter generated by the `displayParam` script is
`-javaagent:<LOCAL_LIB>/products/j2ee/lib/indepth/j2eeagent.jar=com.precise.javaperf`
`.javagent.agents.DefaultJavaAgent`

 Use the correct file separator style for UNIX and Windows '\ ' or '/ '.

- To use an instance name other than the server name, add `-Djvm.name=<JVMNAME>`. For more information on what happens if the JVM name argument is not supplied, see [JVM naming](#).

- You can optionally add `-DPRECISE_SSL_APPLICATION=<APPLICATION_NAME>` to configure the instance application, and `-DPRECISE_SSL_TIER=<TIER_NAME>` to configure the instance Tier.
- To communicate with the SSL server using a proxy, you can add the following proxy parameters:
`-DPRECISE_SSL_PROXY_HOST=<PROXY_HOST>`
`-DPRECISE_SSL_PROXY_PORT=<PROXY_PORT>`
`-DPRECISE_SSL_PROXY_PROTOCOL=<PROXY_PROTOCOL>`

Supported values are `SOCKS`, `HTTP`, and `DIRECT`. The default value is `SOCKS`.

JVM naming

It is highly recommended to specify the JVM name in the JVM arguments as described in [JVM arguments](#)! If you did not follow the recommendation, the JVM name will be resolved automatically in the following way:

- For WebLogic and WebSphere JVMs running as WebLogic/WebSphere console managed servers correspondingly, the JVM name will be set to the server name if it was not specified explicitly.
- For the rest of the JVMs the JVM name will be set to `<server_name>_<running_number>`, where `<server_name>` is the name of the server on which the JVM runs, while `<running_number>` will help to distinguish between different JVMs running on the same server, but using different extracted Precise folders.
- ***** (Requires testing and decision if we support this case!!!!) ***** If more than one JVM will be run on the same server using the same extracted Precise folder and for none of them the JVM name argument is supplied, the activity of both JVMs will be presented in the same invocations tree and the metrics values will be displayed as average values between all the JVMs.

Shared script JVMs

If several JVMs are sharing the startup script where Precise JVM arguments were added, all of them will be started up with the same JVM name, whether it was specified or not. In such case all the data of the JVMs will be presented in the same way as described in the third bullet of [JVM naming](#).

Advanced installation notes

This section contains advanced installation notes.

- If you have uninstalled a remote J2EE instance, and you want to install it again, on the Precise FocalPoint remove the `products/infra/var/<INSTANCE_NAME>_id.txt` file from the local extraction folder, and restart the JVM.
- By default, the reinstalled instance includes history data from the previously uninstalled instance. If the server name or instance name has changed and you want the newly installed instance to include the history data of the previously uninstalled instance, you can add the following property after the removal of the `products/infra/var/<INSTANCE_NAME>_id.txt` file and before restarting of the JVM:
`-DPRECISE_SSL_RECOVERED_ID=<PREVIOUS_ID>`

To find the related instance ID, use the `get-instance-id` command, see [Installing a new instance while keeping historical data from an old instance](#) on page 82.

Installation on WebSphere

This section describes the installation procedure for WebSphere and if Java 2 Security is enabled.

Installation procedure for WebSphere

For J9, add the following to the JVM argument:

```
-javaagent:<PRECISE_LIB>\products\j2ee\lib\indepth\j2ee-agent.jar=com.precise.javaperf.javaagent.agents.IBMJavaAgent
```

For example:

```
-javaagent:E:\Precise\products\j2ee\lib\indepth\j2ee-agent.jar=com.precise.javaperf.javaagent.agents.IBMJavaAgent
```

For all the others, add the following to the JVM argument:

```
-javaagent:<LOCAL_LIB>/products/j2ee/lib/indepth/j2ee-agent.jar
```

Alternatively, you can run the `displayParams` script, which will generate the JVM parameter.

Java 2 Security is enabled

Perform the following procedure.

To determine whether or not Java 2 Security is enabled

1. Connect to the WebSphere admin console with a user that has administration rights.
2. In Admin Console>Security>Global Security, check if the Java 2 Security check box is checked.
 - a. If Java 2 Security is enabled, certain permissions need to be explicitly granted in a `server.policy` file.
 - i. Determine the location of the `server.policy` file. This can be found by examining the JVM startup command and locating the `Djava.security.policy` argument. It typically looks like this:

```
-Djava.security.policy=<IBM Installation folder>\WebSphere\AppServer\profiles\<PROFILE>\properties\server.policy where <PROFILE> is the profile name.
```

b. If no such argument is found, add one using the above example.

3. Add the following to the `server.policy` file:

```
grant {  
  permission java.util.PropertyPermission "indepth.j2ee.classLoaderHandle", "read";  
};  
grant codeBase "file: <PRECISE_LIB>/" {  
  permission java.security.AllPermission;  
};
```



The `< PRECISE_LIB>` inside the `server.policy` file needs to be specified using URL format. For example:

Windows

```
grant codeBase "file:/E:/Precise/-" {  
  permission java.security.AllPermission;  
};
```

UNIX

```
grant codeBase "file:/opt/precise/-" {  
  permission java.security.AllPermission;  
};
```

Additional information

For information on uninstallation of the remote J2EE instance, see the Deleting a Tier instance or cluster and Uninstalling a J2EE remote instance sections in the *Precise Installation Guide*.

For information on monitoring settings for a remote J2EE instance, see the About monitoring settings for a remote J2EE instance section in the *Administration Guide*.

[IDERA Website](#) | [Products](#) | [Buy](#) | [Support](#) | [Community](#) | [About Us](#) | [Resources](#) | [Legal](#)