# Precise Post-Installation Tasks by Product

User Guide

Version 10.2

IDERA

# Precise Post-Installation Tasks by Product User Guide

**Global Headquarters**
Brookhollow Central III
2950 North Loop Freeway West, Suite 700
Houston, Texas 77092
Phone: +1-877-693-1886
Fax: +1-650-898-1666

# Third-party legal notices

Third-party software may be recommended, distributed, embedded, or bundled with this Precise product. Such third-party software is licensed separately by its copyright holder. All third-party copyrights associated with this product are listed in the accompanying release notes.

# Customer Support

For technical assistance, visit our customer portal at http://www.idera.com/support/productsupport where you can find an extensive knowledge base, product updates and our online community forums. You can also contact our Customer Support Team via our customer portal, or go to http://www.idera.com/support/productsupport for a list of our support access numbers in your country.

# Contents

**1**

# Introduction

This document includes the following chapters:

# General Post-Installation Tasks

There are no general post-installation tasks.

# 3

# Post-Installation Tasks for Web Tier Collectors

When the Web Tier Collectors have been installed, post-installation action items may be required. There might be additional action items not listed below, that are required in each installation. Upon finishing your installation, verify that all action items have been completed by checking the list generated on the Action Items screen.

## Post-installation tasks for WebLogic, WebSphere, Tomcat, and Sun One (J2EE stack)

To monitor the application(s), you need to register the Precise for Web agent(s) as Web application filters. This section describes how to add Precise for Web technology to the Web application(s) Web descriptor files.

> **NOTE** If you have installed a cluster, this is the only documented post-installation action item. Verify that no additional action items are needed by checking the list Action Items screen.

- To automatically modify the `web.xml` file(s) of the monitored application(s), you are required to login as the user that has permissions to change the application files and run a script that adds the Precise for Web agents to your application.

  As an input to that script you need to provide the location of the `.EAR`, `.WAR` or `web.xml` file(s) of the application(s).

  Note that if you change a locally deployed application, then any application redeployment may override this action item, so you will need to do it again. If you do not want to do this after every redeployment, you have 2 options:

- (Best practice) To add this script to your application redeployment script, so it will be done automatically.

- Change the central application file (if the application is clustered and there is a central location as opposed to a local location). This way the Precise filters will stay in the application in each redeployment.

  > **NOTE** If you do so, you will need to redeploy the application instead of performing a mere JVM restart.

  > **NOTE** If you are installing Precise for Web on a cluster (WebLogic or WebSphere) the option to change the central application file is the only option.

- Examples of application locations:

  - The WebLogic 10.3 console shows the .EAR file location under "deployments". Then click on your application name to see the path.

  - The WebSphere 6.1 console displays the names of `.EAR` files, but not their location. The typical location of and `.EAR` file is:

    ```
    <install-dir>\profiles\[profile-name]\config\cells\[cell-name]\
    applications\PlantsByWebSphere.ear\PlantsByWebSphere.ear
    ```

  - For Tomcat 5.x & 6.x: *<tomcat-dir>*\webapps\*<app-name>*\web-inf\web.xml

  - For Sun One 6.x: *<sunone-dir>*\WebServer6.1\*<domain-name>*\webappsWeb.xml appears inside each application.

# Modifying the Web descriptor file on a Web application with multiple instances

When installing a Web application with multiple instances, running the `psww_web_desc_modify` command will create multiple EAR files, with each EAR file containing a different `instanceId` and `i3root` value, corresponding to the instances in the Web application.

To avoid creating multiple EAR files

1. Run the web desc modifier without the `-instanceId` and `-i3root` parameters.

2. Add the following parameters to the JVM command line:

    o  -DPssInstanceID=*<instance_ID>*

    o  -DPssI3Root=<precise_root>

## Adding Precise for Web filter(s) to the Web server classpath

To monitor the application, the Precise for Web agent(s) jar(s) must be loaded. Consult with your Web Administrator on where the right location is to do so. Examples:

* WebLogic 9.2: *<bea-root>*\weblogic91\samples\domains\medrec\startWebLogic.cmd WebSphere 6.1 changes the server.xmlfile located for example in

    ```
    F:\webservers_linux\poollin3\ibm\websphere61\profiles\AppSrv01\config\cells\
    poollin3Node01Cell\nodes\poollin3Node01\servers\server1\server.xml
    ```

* Tomcat 5.x & 6.x: *<tomcat-dir>* `\bin\setclasspath.bat(sh)` or click the Tomcat executable, and in the Apage Tomcat Properties dialog box go to the Java tab. There you can see the Java classpath.

    | NOTE | You will also need to add the `servlet-api.jar` file manually to the classpath. It can be found at: e.g. *<tomcat- dir>*`\common\lib\servlet-api.jar`. |
    |---|---|

* Sun One 6.x uses the management console to change the classpath or you can edit the *<sun-one-install-dir>*`\`*<server-name>*`\config\server.xml` file and follow the instruction in the action item.

## Enabling statistics collection on WebSphere 6.1 - 6.x and 7.x

1. Open WebSphere admin console (URL: `ibm/console/login.do`).

2. In the Integrated Solutions Console, click the **Monitoring and Tuning** tab.

3. From the drop-down menu, click on **Performance Monitoring Infrastructure** (PMI).

4. On the right pane, select the server upon which you want to enable statistics (for example, server1.)

5. In the next screen, click the **Runtime** tab.

6. On the Runtime tab, select either:

    o  Basic

    o  Extended

    o  All

    o  Custom

    | NOTE | If you select Custom, verify that the CommittedCount counter under Transaction Manager is enabled. |
    |---|---|

## Adding the session cookie name of the monitored application

To verify that browser-side and server-side data is properly correlated, specify the name of the session cookie that is used by the application in the `settings.xml` file. Session cookie names can be JSESSIONID, _sn, SID, etc. Contact your application administrator to find the name of the cookie.

To add the session cookie name

1. In **Registry Path** in the Registry Editor (**AdminPoint > Settings > Registry Editor**), type the following path, and then click **Go**:
   `registry/instances/www/<server_name>/<instance_name>/settings`.

2. In the Xml Editor tab, search for the <filter-session-cookie> tag.

3. Add the applicable cookie name between the tag, for example:

   <filter-session-cookie>_sn</filter-session-cookie>

4. Click **Save**.

---

**NOTE**    The cookie session name for J2EE (JSESSIONID) is the default value for the <filter-session-cookie> tag.

---

## Adding post-parameters collection for IIS6

To add post-parameters collection for IIS, see the following sections:

- Inserting wildcard mapping for Precise post-parameters collection extension
- Add the Precise post-parameters collection extension to the trusted list
- Updating the collect-post-parameters tag

### Inserting wildcard mapping for Precise post-parameters collection extension

By inserting wildcard application mapping, you can instruct IIS to allow ISAPI applications known as interceptors to intercept and examine every request before the requested page is sent to its mapped application. The effect is an application mapping that handles every file name extension.

---

**NOTE**    You must be a member of the Administrators group on the local computer to perform the following procedure or procedures. As a security best practice, log in to your computer by using an account that is not in the Administrators group, and then use the runas command to run IIS Manager as an administrator. At a command prompt, type run as `/User:Administrative_AccountName"mmc systemroot\system32\inetsrv\iis.msc"`.

---

To add a wildcard application mapping to a Web server or Web site

1. In IIS Manager, expand the local computer, expand the Web Sites folder, expand the "Default-Web-Site", right-click the virtual directory that you want to collect post parameters on, and then click **Properties**.

2. Click Add Wildcard Script Map.

3. In the Executable box, enter the full path or browse to the native handler (.dll or .exe file) that processes the request:
   `<precise_root>\products\www\instances\<instance_id>\bin\IISFilter.dll`

4. In the Name box, enter a name for the handler mapping.

5. Click **OK**.

6. Click Yes, to add the extension with an Allowed entry to the ISAPI and CGI Restrictions list. Repeat this for each virtual directory you want to collect post parameters in.

## Add the Precise post-parameters collection extension to the trusted list

1. In IIS Manager, expand the local computer; expand the Web Services Extensions, and then click **Add a new Web service extension**.

2. Put "Precise post parameters" as extension name.

3. Click **Add**. In the "Path to file" field type the path to the DLL:

   `<precise_root>\products\www\instances\<instance_id>\bin\IISFilter.dll`

4. Click **OK**.

5. Mark the "Set extension status to Allowed" check box.

6. Click **OK**.

## Updating the collect-post-parameters tag

To update the collect-post-parameters tag

1. In the "Registry Path" field in the Registry Editor (**AdminPoint > Settings > Registry Editor**), enter the following path and click **Go**:
   `registry/instances/www/<<server_name>>/<instance_name>/settings.`

2. In the Xml Editor tab, change the value of the <collect-post-parameters> tag to "true".

3. Click **Save**.

# Adding post-parameters collection for IIS7

To add post-parameters collection for IIS, see the following sections:

- [Inserting wildcard mapping for Precise post-parameters collection extension](#)

- [Updating the collect-post-parameters tag](#)

## Inserting wildcard mapping for Precise post-parameters collection extension

By inserting wildcard application mapping, you can instruct IIS to allow ISAPI applications known as interceptors to intercept and examine every request before the requested page is sent to its mapped application. The effect is an application mapping that handles every file name extension.

| NOTE | You must be a member of the Administrators group on the local computer to perform the following procedure or procedures. As a security best practice, log in to your computer by using an account that is not in the Administrators group, and then use the runas command to run IIS Manager as an administrator. At a command prompt, type run as `/User:Administrative_AccountName"mmc systemroot\system32\inetsrv\iis.msc".` |
|------|---|

| NOTE | It is highly recommended to install the Microsoft hotfix mentioned in KB 2505146 (http://support.microsoft.com/kb/2505146) as this solves a Microsoft IIS7 issue. Should you not apply the hotfix and continue to perform the following steps, your application will stop working. |
|------|---|

To add wildcard application mapping to a Web server or Web site

1. Go to the `Precise_root` directory and run the following script:

   ```
   products\www\install\psww_insert_isapi_handlers.bat <precise_root>
   <instance_id> <site_name> add-handler PreciseWebHandler <filter_path>
   ```
   where:

   | **filter_path** | The path to the instance bin folder concatenated with "IISFilter", without the `_32.dll` or `_64.dll` extension. |
   |---|---|
   | | For example: c:\precise\products\www\instances\2643\bin\IISFilter |

   For example:

   ```
   products\www\install\psww_insert_isapi_handlers.bat "E:\Precise
   Installation" 2442 Petshop add-handler PreciseWebHandler "E:\Precise
   Installation\products\www\instances\2442\bin\IISFilter"
   ```

## Updating the collect-post-parameters tag

To update the collect-post-parameters tag

1.  In the "Registry Path" field in the Registry Editor (**AdminPoint > Settings > Registry Editor**), enter the following path and click **Go**:

    ```
    registry/instances/www/<server_name>/<instance_name>/settings/server-side.
    ```

2.  In the Xml Editor tab, change the value of the <collect-post-parameters> tag to "true".

3.  Click **Save**.

# Adding post-parameters collection for all other supported Web servers

To add post-parameters collection for all other supported Web servers, see the following section:

| NOTE | This feature is not supported for Apache 1.3. |
| --- | --- |

*   Updating the collect-post-parameters tag

## Updating the collect-post-parameters tag

To update the collect-post-parameters tag

1.  In the "Registry Path" field in the Registry Editor (**AdminPoint > Settings > Registry Editor**), enter the following path and click **Go**:

    ```
    registry/instances/www/<server_name>/<instance_name>/settings/server-side.
    ```

2.  In the Xml Editor tab, change the value of the <collect-post-parameters> tag to "true".

3.  Click **Save**.

# Removing post-parameters collection for IIS7

To remove wildcard application mapping from a Web server or Web site

1.  Go to the `Precise_root` directory and run the following script:

    ```
    products\www\install\psww_insert_isapi_handlers.bat <precise_root>
    <instance_id> <site_name> remove-handler PreciseWebHandler <filter_path>
    ```
    where:

    | filter_path | The path to the instance bin folder concatenated with "IISFilter", without the `_32.dll` or `_64.dll` extension. |
    | --- | --- |
    | | For example: c:\precise\products\www\instances\2643\bin\IISFilter |

    For example:

    ```
    products\www\install\psww_insert_isapi_handlers.bat "E:\Precise
    Installation" 2442 Petshop remove-handler PreciseWebHandler "E:\Precise
    Installation\products\www\instances\2442\bin\IISFilter"
    ```

# 4

# Post-Installation Tasks for J2EE Tier Collectors

After installing Precise agents for the J2EE Tier, post-installation action items must be executed. These action items vary according to the supported application, and can usually be executed automatically from the user interface. In some circumstances, however, these action items must be executed manually.

This section describes the automatic and manual post-installation action items for J2EE Tiers, and includes the following topics:

- Automatically executing post-installation action items
- Manually executing post-installation action items

## Automatically executing post-installation action items

Most action items automatically appear as part of the installation process in the user interface.

To perform J2EE Tier post-installation tasks

1. In the J2EE Tier - Installation dialog box, click **Next**.
2. In the J2EE Tier - Post-Installation dialog box, follow the instructions to perform all post-installation action items.
3. Click **Finish**.

## Manually executing post-installation action items

This section describes the following manual procedures:

- Manually enabling CPU capturing for IBM AIX version 5.x
- Manually enabling Precise for J2EE monitoring
- Manually executing post-installation action items when multiple clusters share a startup script
- Manually executing WebSphere post-installation action items in instance and cluster applications

## Manually enabling CPU capturing for IBM AIX version 5.x

If the application server is running on an IBM AIX version 5.x computer, and you want to enable Precise for J2EE to report CPU numbers for each method invocation, set and export the AIXTHREAD_ENRUSG parameter as follows:

```
AIXTHREAD_ENRUSG=ON
export AIXTHREAD_ENRUSG
```

## Manually enabling Precise for J2EE monitoring

If the automatic execution of the post-installation action items failed, perform the following steps to enable monitoring for the installed instance or cluster:

| NOTE | Verify that the instance (JVM) or cluster name you enter is the same as the instance (JVM) or cluster name that you defined during installation. |
|------|---|

- For Java 1.5

  - Add the following arguments to the Java command line of the JVM to be monitored:

  - For an instance

    ```
    -javaagent:<precise_root>\products\j2ee\lib\indepthj2ee-
    agent.jar=com. precise.javaperf.javaagent.agents.DefaultJavaAgent
    -Dindepth.j2ee.server=<precise_root>\products\j2ee\config\
    <INSTANCE_NAME> -DisClustered=false
    ```

  - For a cluster

    ```
    -javaagent:<precise_root>\products\j2ee\lib\indepthj2ee-
    agent.jar=com. precise.javaperf.javaagent.agents.DefaultJavaAgent
    -
    Dindepth.j2ee.server=<precise_root>\products\j2ee\config\<CLUSTER_NA
    ME> -DisClustered=true
    ```

  | NOTE | Be sure to use the `<JAVA_HOME>` that the application server uses, not the `<JAVA_HOME>` located in the Precise product installation folder. The apladmin program will attempt to prevent any use of a JVM located under <precise_root>. |
  |------|---|

- For Java 1.4

  a. Configure instrumentation by executing the following command from the <precise_root> folder, replacing `<JAVA_HOME>` with the path to the Java home of the application to be monitored:

  | **Windows** | products\j2ee\bin\apladmin modrt --jvm-id=<instance (jvm)_name/cluster_name> "<JAVA_HOME>\bin\java.exe" |
  |-------------|---|
  | | For example: |
  | | products\j2ee\bin\apladmin modrt --jvm-id=<instance (jvm)_name/cluster_name> "c:\j2sdk1.4.2_08\bin\java.exe" |
  | **UNIX** | products/j2ee/bin/apladmin.sh modrt --jvm-id=<instance (jvm)_name/cluster_name> "<JAVA_HOME>/bin/java" |
  | | For example: |
  | | products/j2ee/bin/apladmin.sh modrt --jvm-id=<instance (jvm)_name/cluster_name> "/opt/j2sdk1.4.2_08/bin/java" |

  | NOTE | Be sure to use the `<JAVA_HOME>` that the application server uses, not the `<JAVA_HOME>` located in the Precise product installation folder. The apladmin program will attempt to prevent any use of a JVM located under <precise_root>. |
  |------|---|

  b. Follow the instructions in the command output to modify the Java command line of the JVM to be monitored.

  c. Add the -DisClustered=false flag to the arguments from apladmin. For a clustered installation, specify 'true' for this flag.

  d. After executing the apladmin.modrt, add the parameters to the JVM startup command line.

  For example:

  ```
  -Xbootclasspath/p:C:\Precise\products\j2ee\etc\apl\jvms\java-vm-TestApp\lib\rt.jar
  -Xbootclasspath/p: C:\Precise\products\j2ee\lib\indepthj2eeboot.jar
  -Xbootclasspath/p: C:\Precise\products\j2ee\classes
  -Dindepth.j2ee.server= C:\Precise\products\j2ee\config\TestApp
  -DisClustered=false
  ```

## Manually executing post-installation action items when multiple clusters share a startup script

If two different clusters are started by the same startup script, the automatic action items will not work properly. In this case, the user must edit the startup script manually and adjust it to the relevant cluster configuration.

To manually execute the action items

1. Start the cluster with Precise instrumentation by adding the following parameters to the Java command:

   -javaagent:<precise_root>\products\j2ee\lib\indepthj2ee-agent.jar=com.precise.javaperf.javaagent.agents.DefaultJavaAgent
   -Dindepth.j2ee.server=<precise_root>\products\j2ee\config\<cluster_name>
   -DisClustered=true

2. Repeat for each cluster started by the startup script.

## Manually executing WebSphere post-installation action items in instance and cluster applications

By default, the user can automatically execute the post-installation action items for WebSphere application servers from the action items screen during installation. However, in some cases the automatic execution of the action items will not be possible or will fail. For example, if the WAS is secured, or the auto-detect phase failed (because the server was down, etc.), the automatic action items execution will fail. In such cases, the user will have to execute the action items manually.

To manually instrument a WebSphere instance/cluster

1. Configure instrumentation by executing the following command from the <precise_root> folder, replacing <JAVA_HOME> with the path to the Java home of the application to be monitored.

   | | |
   |---|---|
   | **Windows** | products\j2ee\bin\apladmin modrt --jvm-id=<instance (jvm)_name/cluster_name> "<JAVA_HOME>\bin\java.exe" |
   | **UNIX** | ./products/j2ee/bin/apladmin.sh modrt --jvm-id=<instance (jvm)_name/cluster_name> "<JAVA_HOME>\bin\java.exe" |

   | NOTE | Be sure to use the <JAVA_HOME> that the application server uses, not the <JAVA_HOME> located in the Precise product installation folder. |
   |---|---|

2. Follow the apladmin instructions.

3. Add the `-DisClustered=false` flag to the arguments from apladmin. For a cluster, specify 'true' for this flag.

4. To enable RMI transaction monitoring, add the following text:
   Dorg.omg.PortableInterceptor.ORBInitializerClass.com.precise.javaperf.lib.smartlink.PreciseSmartlinkORBInitializer For example, the user will add the following arguments to WAS java arguments:

   -Xbootclasspath/p:C:\Precise\products\j2ee\etc\apl\jvms\java-vm-TestApp\lib\rt.jar
   -Xbootclasspath/p: C:\Precise\products\j2ee\lib\indepthj2eeboot.jar
   -Xbootclasspath/p: C:\Precise\products\j2ee\classes
   -Dindepth.j2ee.server= C:\Precise\products\j2ee\config\TestApp
   -DisClustered=false

   | NOTE | Generally the <JAVA_HOME> for WebSphere installations can be found in the **Application -> WebSphere variables** section in the WebSphere Administrative Console. |
   |---|---|

5. Enable Application Server Metrics by performing the following steps:

   a. Verify that the Performance Monitoring Service is enabled by marking the "Enable Performance Monitoring Infrastructure (PMI)" setting located on the **Application Servers > <SERVER_NAME> > Performance Monitoring Infrastructure (PMI)** page.

   b. Verify that the initial specification level is "Basic" or "Custom".

   c. After Performance Monitoring Service is confirmed to be enabled at startup, perform the following steps to configure the Application Server Metrics collection:

      i. Select Servers > Application Servers > <SERVER_NAME> > Server Infrastructure > Administration > Custom Services

      ii. Click **New.**

       iii.   Mark the "Enable service at service startup" check box.

       iv.   Leave the External Configuration URL blank.

       v.   Set the "Classname" to com.precise.javaperf.extensions.websphere.PreciseMetricPluginLoader

       vi.   Set the "Display Name" to PreciseMetricPluginLoader

       vii.   Set the "Classpath" to: <precise_root>/products/j2ee/lib/indepthmetric.jar

       viii.   Save the changes.

6.   Update the security policy.

    a.   Edit the <WAS_INSTALL_ROOT>/profiles/<SERVER>/properties/server.policyfile to include the following:

```
grant codeBase "file:<precise_root>/-" {
        permission java.security.AllPermission;
};
```

> **NOTE**    In Windows, change the PATH separator from '\' to the UNIX style separator '/'.

7.   Restart the Application server.

> **NOTE**    For clustered applications, the user must verify that the `rt.jar` generated in step 1, and pointed to by the `Xbootclasspath`, is located in all the servers that are running as part of the WebSphere cluster. The user can copy the `rt.jar` to any central location, and update the `Xbootclasspath` accordingly.

# Post-Installation Tasks for Microsoft .NET Tier Collectors

This section includes the following topics:

- [Post-installation action items](#)

When the Microsoft .NET Tier Collectors have been installed, you need to perform post-installation action items. These action items vary depending on the .NET application type, and include editing configuration files and restarting Precise agents. Not all action items appear on screen as described in [Performing post-installation action items](#) on page 18. This section includes instructions regarding action items that appear as part of the installation, as well as an overview of instrumentation action items that are not included.

## Performing post-installation action items

To perform Microsoft .NET Tier post-installation tasks

1.  In the Microsoft .NET Tier - Installation dialog box, click **Next**.

2.  In the Microsoft .NET Tier - Post-Installation dialog box, follow the instructions to perform all post-installation tasks.

3.  Click **Finish**.

## Configuring DLLs, classes, and methods to instrument

You can add the DLLs you wish to monitor using the DLLs dialog box in the Instrumentation Settings menu, as explained in the Precise for Microsoft .NET User's Guide. But afterwards you must edit the file `instrumentation.xml` manually to define which classes and methods to instrument.

The Precise for Microsoft .NET activity tracking is based on instrumentation of your application DLLs. This information is configured in the `instrumentation.xml` file. The way you configure this file (this means which DLLs to instrument and which classes or methods) affect how much information will be tracked and also the overhead of the Precise for Microsoft .NET activity tracking on your application.

The `instrumentation.xml` file contains many configurable settings, which are explained in the *Precise* Administration Guide in detail.

The following file is an example section of the `instrumentation.xml` file which configures the instrumentation of the Microsoft Petshop sample Web application:

```
<instance name="AspNetIIS6">
   <dlls>
      <dll name="pet*.dll" /><!- instrument all DLLs that start with pet* -!>
   </dlls>
   <instrument>
      <classes>
         <class>
         <name>Petshop.*</name><!- instrument all classes of which the
         namespace begins with Petshop -!>
         <called-method>
         <methods>
         <method>
         <name>*</name><!- instrument all methods in each class -!>
         </method>
         </methods>
```

```
            </called-method>
          </class>
      </classes>
   </instrument>
   ….
```

# Defining permission settings for a Precise installation folder

The Precise for Microsoft .NET hooks that are injected into the monitored .NET process, are run under the same user account on which your .NET application is running. This user must have full permission rights to the

<*precise_root*> folder and all its sub-folders.

You can change the security settings of the <*precise_root*> folder with the Windows File Explorer Security tab.

| NOTE | If you have several application pools which translate into different Windows users, all these users must be granted full permission to the <*precise_root*> folder. |
|------|---|

# Changing the activity tracking threshold

The Precise for Microsoft .NET installation pre-configures the activity data tracking threshold and sets it to 50 milliseconds. This means that only methods, service requests, SQL statements, and Web services that have a response time that is equal to, or greater than 50 milliseconds will be collected.

You can change this default value in the `ActivityCollector.xml` file, located in the <*precise_root*>/products/dotnet/config folder any time after the installation:

```
<activity-collector-config>
   <!-- psdn_act_col.exe configuration, configurable per server (default)
   & per instance -->
   <aggregator>
      <topnsql>5</topnsql><!-- Top n SQL statement to monitor -->
      <sla>5000</sla><!-- Red SLA breach value for top HTTP invocations in msec.
      -->
      <nearsla>1000</nearsla><!-- Yellow SLA breach value for top http
      invocations in msec. -->
      <insane-rt>300</insane-rt><!-- Long running thread timeout in seconds -->
      <bit-vector>false</bit-vector>
      <last-level-bit-vector>false</last-level-bit-vector>
   </aggregator>
   <!-- Tracker.dll settings, configurable per server (default) & per instance -->
   <tracker>
      <threshold>50</threshold><!-- Threshold in msec for filtering out
      events before forwarding it to the Collector agent. -->
   </tracker>
</activity-collector-config>
```

| NOTE | The new setting will only take effect after your .NET application is restarted. For ASP.NET that means an IIS reset or recycling the application pool. |
|------|---|

| NOTE | This setting has a significant effect on the overall overhead of Precise for Microsoft .NET agent on your system. Lowering this value too much can cause your Web application and server to slow down considerably. If Precise for Microsoft .NET is installed on a development server, you can set this value to 0 to collect all data. On heavy loaded production systems, it is advisable not to change the default. |
|------|---|

More information about controlling the Precise for Microsoft .NET overhead on your .NET application in the *Precise Administration Guide.*

# About Precise for Microsoft .NET agent ports

The Precise for Microsoft .NET data collection is based on establishing a TCP connection between a tracking module (`tracker.dll`), the activity Collector and instrumentation agent. The default port numbers for the installation are respectively 20755 and 20756. If any of these port numbers are in use by another application on the server, you can change its value respectively in the `collector.xml` and `instrumenter.xml` files.

Port conflicts can cause the Precise for Microsoft .NET Collector not to work and to submit errors. The following is an error example that can appear in the `dotnet.collector.activity.trc` file:

```
======================================
ID        : CONFIG-0000000
Time      : 2006-11-23 10:03:54.750
GMT+02 i3 Time    : 2006-11-23
08:03:54.750 GMT Process :
psdn_dncol_act.exe -l (ID:2704)
Logger    : com.Symantec.apm.i4dotnet.collector.activity
Msg       : Exception in remoting : System.Net.Sockets.SocketException: Only
one usage of each socket address (protocol/network address/port) is normally
permitted
======================================
```

The following is an error example that can appear in the `dotnet.collector.dncol_isntr.trc` file:

```
================================
====== ID : CONFIG-0000000
Time      : 2006-11-23 10:09:05.842
GMT+02 i3 Time    : 2006-11-23
08:09:05.842 GMT Process :
psdn_dncol_instr.exe (ID:4972)
Logger    : com.Symantec.apm.i4dotnet.collector.dncol_instr
Msg       : Unable to setup TCP server on port 20756, check that it is not
used by another process, ex=Only one usage of each socket address
(protocol/network address/port) is normally permitted
======================================
```

# 6

# Post-Installation Tasks for Oracle Tier Collectors

This section includes the following topics:

- [Post-installation action items](#)

## Post-installation action items

When the Oracle Tier schema is installed, post-installation tasks may be required. This mainly depends on factors such as the Oracle version, the operating system, and the granted privileges. In addition, this depends on whether Precise for Storage or Interpoint is installed.

For installation details, see the appropriate section for the specific product.

### Installing the Interpoint for PeopleSoft Collector's trigger

If you have installed the Interpoint for PeopleSoft Collector to monitor PeopleSoft activities, you need to create a trigger on the table PSPRCSRQST in the PeopleSoft database. In addition, you need to enable the database monitoring parameter in the PeopleSoft configuration file.

To install the Interpoint for PeopleSoft Collector's trigger and enable the database monitoring parameter

1. Verify that you have a valid Interpoint for PeopleSoft license.

2. Open the PeopleSoft configuration file psappsrv.cfg (usually located in the `PT%version%\appserv\%domainName%` folder). If EnableDBMonitoring=1, proceed to step 7. Otherwise, proceed to step 3.

3. Disconnect all PeopleSoft users.

4. Shut down all PeopleSoft application servers.

5. Set the database monitoring parameter to TRUE: `EnableDBMonitoring=1`

6. Restart the PeopleSoft application servers.

7. Connect to the database that PeopleSoft is installed on. Use the user that owns the PeopleSoft database.

   To find out who owns this database, run the following query:

   select owner from dba_objects where object_name = 'PSPRCSRQST';

   The owner returned is the one who owns the PeopleSoft database.

8. Run the following SQL script:

   ```
   create or replace trigger
   precise_psprcsrqst_register after
   update of runstatus on psprcsrqst
   for each row
   when (new.runstatus = 7 and
   old.runstatus != 7) begin
   dbms_application_info.set_client_i
   nfo(:new.oprid
   ||',,,'|| :new.prcsinstance || ',' || :new.prcsname
   ||',');
   exception when others then null;
   end;
   /
   ```

# 7

# Post-Installation Tasks for SQL Server Tier Collectors

This section includes the following topics:

- [Installing the product Interpoint for PeopleSoft Collector trigger](#)

When the SQL Server Tier is installed, post-installation tasks may be required. This may depend on whether Precise for Storage or ERP Extension is installed.

After you install SQL Server, a user who is a member of the sysadmin fixed server role must launch Database Engine Tuning Advisor before anyone else. When the sysadmin user launches Database Engine Tuning Advisor that action initializes the application for use by others. After Database Engine Tuning Advisor is initialized, any user who is a member of the db_owner fixed database role can use it to tune tables on databases they own. For more information about initializing Database Engine Tuning Advisor, see the Initializing Database Engine Tuning Advisor on the MSDN website.

## Installing the product Interpoint for PeopleSoft Collector trigger

If you have installed the Interpoint Collector to monitor PeopleSoft activities, you need to create a trigger on the table PSPRCSRQST in the PeopleSoft database. In addition, you need to enable the database monitoring parameter in the PeopleSoft configuration file.

To install the Interpoint for PeopleSoft Collector's trigger and enable the database monitoring parameter

1. Open the PeopleSoft configuration file: `psappsrv.cfg` (usually under `PT%version%/appserv/%domainName%`). If EnableDBMonitoring=1, proceed to step 6. Otherwise, proceed to step 2.

2. Disconnect all PeopleSoft users.

3. Shut down all PeopleSoft servers.

4. Edit the DB Monitoring parameter to be TRUE: `EnableDBMonitoring=1`

5. Restart the PeopleSoft application servers.

6. Using SQL Query Analyzer, connect to the database holding the PeopleSoft tables. Verify that you are connecting with a user that is a member of the db_owner database role in the database.

7. Execute the following batch:

```
if exists (select name from sysobjects where name = 'precise_psprcsrqst_register' and type = 'TR')
drop trigger precise_psprcsrqst_register go
if exists (select name from sysobjects where name = 'VERITAS_psprcsrqst_register' and type = 'TR')
drop trigger VERITAS_psprcsrqst_register go
if exists (select name from sysobjects where name = 'Symantec_psprcsrqst_register' and type = 'TR')
drop trigger Symantec_psprcsrqst_register go
create trigger precise_psprcsrqst_register on PSPRCSRQST for update as
declare @oprid char(30), @prcsinstance int, @old_status int, @new_status int, @info binary(50)
select @old_status = RUNSTATUS from deleted select
@new_status = RUNSTATUS from inserted if
(@new_status <> 7) or (@old_status = 7) return
select @oprid = OPRID, @prcsinstance = PRCSINSTANCE from inserted
set @info = convert(binary(50), rtrim(@oprid) + ',' + convert(varchar(20), @prcsinstance))
set context_info @info go
```

8

# Post-Installation Tasks for DB2 Collectors

This section includes the following topics:

- [Post-installation action items](#)

## Post-installation action items

When the DB2 Tier is installed, post-installation tasks may be required. Follow any instructions that might appear on the Action Items screen.

**9**

# Post-Installation Tasks for Sybase Tier Collectors

This section includes the following topics:

- [Post-installation action items](#)

## Post-installation action items

When the Sybase Tier Collectors have been installed, you might need to perform post-installation tasks. Follow any instructions that appear on the screen.

# Post-Installation Tasks for Sybase Replication Server Tier Collectors

This section includes the following topics:

- [Post-installation action items](#)

## Post-installation action items

When the Sybase Tier Collectors have been installed, you might need to perform post-installation tasks. Follow any instructions that appear on the screen.

# 11

# Post-Installation Tasks for SAP Tier Collectors

This section includes the following topics:

- [Post-installation actions items](#)

## Post-installation action items

After successfully adding the SAP system to the SAP Tier, you need to define both the mapping for organizations and locales and the locales themselves.

### Defining mapping and locale settings

As part of the pre-installation tasks, you have defined your organizations and user areas. Now that you have finished configuring the SAP system, you need to define the fields that Precise for Precise for SAP will retrieve for these groupings. By default, Precise for Precise for SAP retrieves the Department field for organization and the Building field for user area.

In addition to collecting SAP activity data from application servers, Precise for Precise for SAP can monitor locations for network availability. In Precise for Precise for SAP, a location is referred to as a locale. Each locale represents a physical location that is defined by a list of one or more devices, called Locale Devices. Precise for Precise for SAP connects to these devices to check their availability.

### Mapping the organizations

The following procedure describes how to configure organization settings.

To configure organization settings

1. In Precise for Precise for SAP, from the Settings menu, select **Mapping Settings.**

2. In the Mapping Settings dialog box, select the SAP system you want to configure and click **Edit.**

3. Choose an SAP field from the Organization list to map an SAP organization to a user.

4. Determine which SAP field will be mapped to a locale. Choose a field from the User Area list. Then click **OK.**

| NOTE | If your system maintains user lists in locations other than SAP user tables, you can still import this information into Precise for SAP. Precise for SAP can import user definitions from ASCII text files, enabling user information to be extracted from practically any source. For more information on how to import user information from ASCII files, see the SAP section in the *Precise Administration Guide*. |
|------|---|

### Mapping the Locales

To identify the locales and Locale Devices that you want Precise for Precise for SAP to monitor, you need to select existing Windows machines or other devices, such as routers. Locale Devices do not have to be solely dedicated to availability monitoring. They can run the Echo services required by Precise for SAP in the background while serving other purposes.

The following hardware components can serve as Locale Devices:

- Windows workstations or servers
- UNIX computers
- Routers

| NOTE | Verify that your Locale Devices are always available during monitoring hours. Precise recommends using workstations that are always on and available, instead of computers that may be turned off, such as end-user computers. |
|------|---|

Before updating the list of locales in Precise for Precise for SAP, verify that you have:

- Mapped all SAP user locales.
- Planned which locales should be defined in the Precise for SAP system and which server machines should be assigned to them to physically check their availability.

To configure locale settings

1. In Precise for Precise for SAP, from the Settings menu, select **Mapping Settings.**

2. In the Mapping Settings dialog box, select the SAP system you want to configure and click **Edit.**

3. Determine which SAP field will be mapped to a locale. Choose a field from the User Area list. Then click **OK.**

4. From the Settings menu, select **Locale Settings.**

5. In the Locale Settings dialog box, perform one of the following procedures:

   o To add a locale
   o To edit a locale
   o To delete a locale

To add a locale

1. Click **Add.**

2. In the Locale Settings - Edit dialog box, enter the name of the new locale in the Locale field. Enter a description (optional) of the new locale in the Description field.

3. On the Locale Definition tab, select a user area from the User Areas not assigned to this locale list and click the right arrow to move it to the User Areas assigned to this locale list.

4. On the Locale Availability tab, click **Add**.

5. In the Add Device dialog box, enter the name or the IP address of the device that should be taken into consideration by Precise for SAP when monitoring locale availability.

6. Click **OK.**

To edit a locale

1. Select the Locale Settings you want to modify.

2. Click **Edit**.

3. In the Locale Settings - Edit dialog box, in the Locale text box, change the name of the existing locale.

4. In the Description text box, enter a description (optional) or modify an existing description.

5. On the Locale Definition tab, select a user area from the User Areas not assigned to this locale list and click the right arrow to move it to the User Areas assigned to this locale list.

6. On the Locale Availability tab, click **Edit.**

7. In the Edit Device dialog box, enter the name or the IP address of the device that should be taken into consideration by Precise for SAP when monitoring locale availability. Then click **OK.**

To delete a locale

1. From the Locale Settings list, select the Locale you want to delete.

2. Click **Delete;** then click **OK** to confirm this operation.

3. Click **OK.**

# Installing the SAP Java Connector (JCo)

After installing the Precise for SAP FocalPoint, you need to install the downloaded SAP Java Connector (JCo) if you want Precise for SAP to display running batch jobs in the Batches tab or use the Precise for SAP BW. In this case, perform the following post-installation task. For information on how to download the SAP JCo, see Downloading the SAP Java Connector (JCo) on page **Error! Bookmark not defined.**.

To install the SAP JCo

1. Unzip the contents of the downloaded file to a folder of your choice on the Precise for Precise for SAP FocalPoint server.

2. Open the Agents tab in Precise AdminPoint and stop the Precise for SAP FocalPoint.

3. From the folder to which you unzipped the downloaded file:

   o Copy the `.jar` file to the `<precise_root>\java\3rd_party` folder on your Precise for Precise for SAP FocalPoint server.

   o Copy the `.dll` files to the `<precise_root>\products\sap\bin` folder on your Precise for Precise for SAP FocalPoint server.

4. Verify that the gateway port between the Precise for SAP FocalPoint and the SAP Application Server is open in both directions:

   The gateway port is 33xx, where xx has to be replaced by the instance number in the SAP Application Server.

5. Maintain the SAP communication ports in the services file in:

   ```
   %windir%\system32\drivers\etc\services
   ```

6. Replace the following <SID> and <instance-number> with the values from your application:

   ```
   sapms<SID>   36<instance-number>/tcp   sapms<SID>
   33<instance-number>/tcp  sapms<SID>   32<instance-
   number>/tcp
   ```

   For example for an SAP system with the SID = PRD and Instance Number 01:

   ```
   sapmsPRD 3601/tcp # SAP Message
   Server sapmsPRD 3301/tcp # SAP
   Gateway sapmsPRD 3201/tcp # SAP
   Dispatcher
   ```

7. Go back to AdminPoint and restart the Precise for Precise for SAP FocalPoint.

# Post-Installation Tasks for Oracle Applications Tier Collectors

This section includes the following topics:

- [Post-installation action items](#)

## Post-installation action items

When the Oracle Applications Tier Collectors have been installed, post-installation tasks are required. Depending on your application settings, you may need to stop all Insight Savvies that are currently running on the respective server and restart them. In addition, you need to grant permissions to schema user as described below and activate an Interpoint for Oracle Applications, as well as configure the Interpoint for Oracle Applications for the purpose of data collection.

### Granting permissions to schema user

Connect to the Oracle instance on which the corresponding collector is installed as DBA user and run query:

```
grant execute on DBMS_PIPE to <collector_schema_user>
```

### Activating the Interpoint for Oracle Applications

To activate an Interpoint for Oracle Applications

1.  Log in to the server and run the following script to install the Interpoint:

    | Windows | `\products\oracle\`*SID*`\schema\install_oracle_apps.bat` |
    | UNIX | `./products/oracle/`*SID*`/schema/install_oracle_apps.sh` |

    where `<precise_root>` is the name of the Precise root folder.

2.  To verify that the installation was successful, run the following script:

    | Windows | `verify_install_oracle_apps.bat` |
    | UNIX | `verify install oracle apps.sh` |

3.  Configure the Oracle Applications application.

    For more information on how to configure the Oracle Applications application, see [Configuring an Interpoint for Oracle Applications for data collection](#) on page 29.

4.  Restart the Precise for Oracle Collector agent for this instance.

### Configuring an Interpoint for Oracle Applications for data collection

The Interpoint for Oracle Applications identifies activities of Oracle Applications by monitoring the events that are called by forms. Configuring an Interpoint for Oracle Applications to collect the relevant data involves these tasks:

- Manually updating the customization library
- Manually updating Oracle Applications' system profile (Oracle applications 11.5 or higher)

### Manually updating the customization library

To have the Interpoint for Oracle Applications collect forms data, you must manually add some information to the customization library of the Oracle Applications forms. This information includes user, form, and application names.

The customization library consists of the following files:

- CUSTOM.pld
- CUSTOM.pll
- CUSTOM.plx

You can only modify the CUSTOM.pld file. This file does not exist by default but can be extracted from the CUSTOM.pll file. For the changes to take a global effect in the customization library after they are implemented in the CUSTOM.pld file, you need to regenerate the CUSTOM.pll and CUSTOM.plx files from the CUSTOM.pld file.

| | |
|---|---|
| **NOTE** | The file names may appear lowercase or uppercase (for example CUSTOM.pll). In the following task, all filenames are uppercase. If you copy-paste this information, verify that the case matches the actual file names on your system. |

To update the customization library on UNIX

1. Find the CUSTOM.pll and CUSTOM.plx files, usually located in your Oracle Applications folder.

2. Back up the CUSTOM.pll and CUSTOM.plx files and, if it exists, also the CUSTOM.pld file.

3. Verify that no Oracle Applications user is currently logged on to the database or any other database using the CUSTOM.pll and CUSTOM.plx files.

4. Change to the folder that contains the CUSTOM.pll file. This folder is pointed to by the application variable $FORMS60_PATH.

5. To extract the CUSTOM.pld file, run the following command using the Oracle account of Oracle Applications (such as APPS):

   f60gen Module=CUSTOM.pll Module_Type=LIBRARY userid=*user*/*password* script=yes

   | | |
   |---|---|
   | **NOTE** | For Oracle Applications version R12 replace f60gen command with frmcmp_batch. |

6. Run the following command to edit the file:

   vi CUSTOM.pld

7. Find the EVENT procedure, locate the first BEGIN after the commented sample code, and insert the following code, starting with BEGIN.

   Do not insert the script into commented sample code. You can easily mistake sample code for real code.

   ```
   begin
   if (event_name='WHEN-NEW-FORM-INSTANCE') then declare cursor_name integer;
   rows_processed integer; del char(1) :=
   substr(get_application_property(CURRENT_FORM),1,1); begin
   if (del != '/') then del := '\';
   end if;
   cursor_name := dbms_sql.open_cursor;
   dbms_sql.parse(cursor_name,
   'begin VERITAS_APPS_LOG( "F" , :username ,
   :application , :form , NULL) ; end ;' ,1); dbms_sql.bind_variable(cursor_name,':username' ,
   FND_PROFILE.VALUE('USERNAME')); dbms_sql.bind_variable(cursor_name,':application' ,
   substr(get_application_property(CURRENT_FORM),
   instr(get_application_property(CURRENT_FORM),del ,
   -1)+1, length(substr( get_application_property(CURRENT_FORM),
   instr(get_application_property(CURRENT_FORM), del,-1)+1))-4));
   dbms_sql.bind_variable(cursor_name,':form' , Name_In('system.current_form'));
   rows_processed := dbms_sql.execute(cursor_name);
   dbms_sql.close_cursor(cursor_name);
   exception
   when others then begin dbms_sql.close_cursor(cursor_name); exception
   when others then null;
   end; end; end if;
   exception
   when others then null;
   end;
   ```

8. To regenerate the CUSTOM.pll file from the CUSTOM.pld file, run the following command using the Oracle account of Oracle Applications (such as APPS):

   a. For Oracle Application version under R12 (exclusive):

   ```
   f60gen Module=CUSTOM.pld Module_Type=LIBRARY userid=user/password parse=yes
   ```

   b. For Oracle Applications version R12:

   ```
   frmcmp_batch Module=CUSTOM Module_Type=LIBRARY userid=user/password parse=yes
   ```

9. To regenerate the CUSTOM.plx file from the CUSTOM.pll file, run the following command using the Oracle

account of Oracle Applications (such as APPS):

```
f60gen Module=CUSTOM.pll Module_Type=LIBRARY userid=user/password compile_all=yes
```

> **NOTE**    For Oracle Applications version R12 replace f60gen command with frmcmp_batch.

10. To verify the CUSTOM library change, run the following script:

    ```
    <precise_root>/products/oracle/<sid>/schema/pss_verify_oa_customer_pll.sh
    ```

To update the customization library on Windows

1. Find the CUSTOM.pll and CUSTOM.plx files, usually located in your Oracle Applications folder.

2. Back up the CUSTOM.pll and CUSTOM.plx files.

3. Verify that no Oracle Applications user is currently logged on to the database or any other database using the CUSTOM.pll and CUSTOM.plx files.

4. Run the *%ORACLE_HOME%*\bin\if60gen.exe tool (sometimes called ifbld60.exe).

   > **NOTE**    For Oracle Applications version R12 replace f60gen.exe command with frmcmp.exe.

5. In the welcome wizard, select **Open an existing form,** followed by **OK**.

6. Choose custom.pll and click **Open**.

7. Under PL/SQL Libraries>Custom>Program Units right-click on Package body and select the PL/SQL Editor.

8. Find the EVENT procedure, locate the first BEGIN after the commented sample code, and insert the following code, starting with BEGIN.

   Verify that the script was not inserted into commented sample code. You can easily mistake sample code for real code.

   ```
   begin
   if (event_name='WHEN-NEW-FORM-INSTANCE') then declare
   cursor_name integer; rows_processed integer; del char(1) :=
   substr(get_application_property(CURRENT_FORM),1,1);
   begin
   if (del != '/') then del := '\';
   end if;
   cursor_name := dbms_sql.open_cursor;
   dbms_sql.parse(cursor_name,
   'begin VERITAS_APPS_LOG( "F" , :username ,
   :application , :form , NULL) ; end ;' ,1);
   dbms_sql.bind_variable(cursor_name,':username' ,
   FND_PROFILE.VALUE('USERNAME'));
   dbms_sql.bind_variable(cursor_name,':application' ,
   substr(get_application_property(CURRENT_FORM),
   instr(get_application_property(CURRENT_FORM),del
   ,
   -1)+1, length(substr(
   get_application_property(CURRENT_FORM),
   instr(get_application_property(CURRENT_FORM), del,-1)+1))-
   4)); dbms_sql.bind_variable(cursor_name,':form' ,
   Name_In('system.current_form'));
   rows_processed := dbms_sql.execute(cursor_name);
   dbms_sql.close_cursor(cursor_name);
   exception
   when others then begin dbms_sql.close_cursor(cursor_name);
   exception
   when others then null;
   end; end; end if;
   exception
   when others then null;
   end;
   ```

9. Click File>Save.

10. Run *%ORACLE_HOME%*\bin\ifcmp60.exe.

11. Fill in the following fields:

    | | |
    |---|---|
    | **File** | CUSTOM.PLL (that we changed before) |
    | **Userid** | APPS |
    | **Password** | APPS **password** |

| | |
|---|---|
| **Database** | database name |
| **Module Type** | LIBRARY |
| **Module Access** | FILE |

12. Click **OK**.

# Manually updating Oracle Applications system profile (Oracle applications 11.5 or higher)

To have an Interpoint for Oracle Applications identify interactive forms in Oracle Applications 11.5 or higher, you must manually update the system profile of Oracle Applications with site-specific initialization code. As a result, Oracle Applications forwards the required information - the user, form, and application name for interactive work and the concurrent program and application name for batch work - to an Interpoint for Oracle Applications.

Before updating Oracle Applications' system profile, you must verify that an Interpoint for Oracle Applications was installed successfully. Otherwise, updating the profile may halt Oracle Applications.

To avoid interfering with anyone's work before changing the system profile, you can alternatively change the profile for only one user to first test the outcome.

To verify that an Interpoint for Oracle Applications was installed successfully

1. Run the following script:

   | | |
   |---|---|
   | **Windows** | `\products\oracle\`*SID*`\schema\verify_install_oracle_apps.bat` |
   | **UNIX** | `./products/oracle/`*SID*`/schema/verify install oracle apps.sh` |

To update Oracle Application's system profile

1. Log in to Oracle Applications with a user that has system administrator privileges.

2. Open the System Administrator application.

3. Open the Navigator - System Administrator dialog box.

4. On the Functions tab, select **Profile>System**; then click **Open**.

5. In the Find System Profile Values form, select the Site check box and Type the following value in the Profile text box: **Init%SQL%Custom**

6. Click **Find**.

7. In the System Profile Values form, type the following in the Site text box: **begin veritas_custom_profile; end;**

8. Close the System Profile Values form; then open any form.

   If the form opens cleanly, you can start working with the Interpoint for Oracle Applications.

   If an issue occurs, verify again that the Interpoint for Oracle Applications was installed successfully. If it was installed successfully, most likely the issue was caused by erroneous text in the system profile. If the issue persists, contact Precise Customer Support.

To update the system profile for one user only

1. Log in to Oracle Applications with a user that has system administrator privileges.

2. Open the System Administrator application.

3. Select Profile>System; then click Open.

4. In the Find System Profile Values form, select the Site and User check boxes.

5. To the right of the User check box, enter the name of the user whose profile you want to change.

6. Type the following value in the Profile text box: **Init%SQL%Custom**

7. Click **Find**.

8. In the System Profile Values form, type the following in the User text box:**begin veritas_custom_profile; end;**

9. Close the System Profile Values form and log in to Oracle Applications with the user selected in Step 8.

10. Open any form.

    If the form opens cleanly, you can update the Oracle Applications' system profile.

    For more information, see

If an issue occurs, verify again that the Interpoint for Oracle Applications was installed successfully. If it was installed successfully, most likely the issue was caused by erroneous text in the system profile. If the issue persists, contact Precise Customer Support.

## Correcting problematic profiles

When the Interpoint for Oracle Applications does not correlate some of its information with the Collector agent, the following scenario may be applicable:

The Interpoint for Oracle Applications collects some of its data from the Oracle Applications using the initialization of every form that is declared in the System Profile. The System Profile contains 4 profile levels, one for each level. Listed from the weakest to the strongest are: Site, Application, Responsibility, User. The Interpoints' code is on the weakest link here, meaning, Site level. When you customize one or more profiles at a stronger level, the code that resides at Site level is hidden.

| NOTE | In this query we expect to have Interpoints' code ("begin veritas_custom_profile; end;") in all entries, otherwise there is no correlation. |
| --- | --- |

To find problematic profiles

1.  Use the following query:

    ```
    select a.profile_option_value, DECODE(a.level_id, 10001, 'Site',
    10002, 'Application', 10003, 'Responsibility', 10004, 'User', a.level_id) as "System Profile
    Level", a.level_value, to_char (a.last_update_date, 'YYYY-MON-DD')
    LAST_UPDATE_DATE
    from FND_PROFILE_OPTION_VALUES a, FND_PROFILE_OPTIONS b,
    fnd_profile_options_tl c
    where
    a.profile_option_id = b.profile_option_id and
    b.profile_option_name = c.profile_option_name and
    c.user_profile_option_name = 'Initialization SQL Statement -
    Custom' order by a.LEVEL_ID;
    ```

2.  Add the Interpoints' code within the profile (By changing the relevant profile in Oracle Applications). I.e. verify that the profile_option_value looks like: "begin veritas_custom_profile; <customer code goes here (or empty)> end;"

    This query shows all relevant profiles affecting the code.

## Correcting typing errors in the system profile

Typing errors in the system profile may prevent Oracle Applications forms from opening, including the system profile form. To validate that there is no other definition that overrides your profile definition and to correct any typing errors, you need to log in to the Oracle database externally using SQL*Plus.

To correct typing errors

1.  Log in to your Oracle database with the APPLSYS account.

2.  Run the following command to view erroneous text: Select * from FND_PROFILE_OPTION_VALUES where PROFILE_OPTION_VALUE like '%veritas%';

    You may search for any expression by replacing `'%veritas%'` in the `where` clause with the string you are looking for. For example, to search for the profiles that were entered on July 13, 1975, execute the following command:

    ```
    Select * from FND_PROFILE_OPTION_VALUES where last_update_date=
    to_date ('13-JUL-75', 'DD-MON-YY');
    ```

3.  Run the following command to correct the errors:

    ```
    update FND_PROFILE_OPTION_VALUES set PROFILE_OPTION_VALUE =
    'begin veritas_custom_profile; end;'
    where PROFILE_OPTION_VALUE like '%veritas%';
    ```

    You may update any expression by replacing '%veritas%' in the where clause with the string you are looking for. For example, to update all profiles that were entered on July 13, 1975, execute the following command:

    ```
    update FND_PROFILE_OPTION_VALUES set
    PROFILE_OPTION_VALUE='begin veritas_custom_profile;end;' where
    last_update_date=
    to_date ('13-JUL-75', DD-MON-YY');
    ```

4.  Commit the changes.

# 13

# Post-Installation Tasks for Siebel Tier Collectors

This section includes the following topics:

- [Post-installation actions items](#)

## Post-installation actions items

When the Siebel Tier Collectors have been installed, you need to perform the following post-installation tasks:

- Perform any action items show in Post-installation dialog box. See [Possible Post-installation action items](#) on page 34.

- Modify the SRF (required for Interpoints). See [Modifying the Siebel Repository File (SQL Server)](#) on page 34. See [Modifying the Siebel Repository File (Oracle)](#) on page 35.

- Set up Siebel batch process sampling (required for Interpoints) See [Setting up Siebel batch process sampling](#) on page 35.

- Stop and restart all Insight Savvies installed on the respective server.

- See the *Precise Interpoint for Siebel User Guide* at [http://www.idera.com/support/productsupport](http://www.idera.com/support/productsupport) for additional post-installation action items.

### Possible Post-installation action items

Perform any action items show in Post-installation dialog box.

To perform Siebel Tier post-installation tasks

1. On the Finish screen, click **Next**.

2. In the Siebel Tier - Post-Installation dialog box, follow the instructions to perform all post-installation tasks.

3. If you have installed the Interpoint for an Oracle database, restart the Oracle Tier Collectors on the Siebel server.

4. Click **Finish**.

### Modifying the Siebel Repository File (SQL Server)

Once the Siebel Tier Collectors are installed, modify the Siebel Repository File (SRF), so that Interpoint for Siebel will be able to receive notifications of Siebel-related user events.

Part of the `.srf` file, is a collection of callback scripts, which are triggered by various events in a user's session. A user can view or change these scripts using the Siebel Tools.

- For each targeted Siebel application, modify the `.srf` file on the Siebel server. Define the application name by hard coding it in the **Application_PreNavigate** function.

- The modified SRFs need to be distributed to each of the Siebel servers

If you have installed Interpoint for a SQL Server database, modifying the SRF consists of the following tasks:

- Creating the Interpoint project (SQL Server)

- Creating the CX_VTSSPID table (SQL Server)

- Creating the business component (SQL Server)

- Creating the business object (SQL Server)

- Adding the Application_PreNavigate() function (SQL Server)

- Replacing the table with a custom view (SQL Server)

- Completing the modification (SQL Server)

- Debugging the script (SQL Server)

# Modifying the Siebel Repository File (Oracle)

Once the Siebel Tier Collectors are installed, modify the Siebel Repository File (SRF), so that Interpoint for Siebel will be able to receive notifications of Siebel-related user events.

Part of the `.srf` file, is a collection of callback scripts, which are triggered by various events in a user's session. A user can view or change these scripts using the Siebel Tools.

- For each targeted Siebel application, modify the `.srf` file on the Siebel server. Define the application name by hard coding it in the **Application_PreNavigate** function.

- The modified SRFs need to be distributed to each of the Siebel servers

If you have installed Interpoint for an Oracle database, modifying the SRF consists of the following tasks:

- Creating the Precise Interpoint project (Oracle)

- Creating the CX_PSS_SBL table (Oracle)

- Creating the business component (Oracle)

- Creating the business object (Oracle)

- Adding the Application_PreNavigate() function (Oracle)

- Replacing the table with a global temporary table (Oracle)

- Completing the modification (Oracle)

- Debugging the script (Oracle)

# Setting up Siebel batch process sampling

For Siebel batch processes to be sampled by the Siebel Tier Collectors, you need to create a tab-delimited text file that contains the mapping of each Siebel server to its host machine and network address. Each row in the file represents a physical machine. The first field in each row should represent the name of the Siebel server, the second field its corresponding host machine name, and the third its network address.

# Sampling Siebel batch processes (SQL Server)

If you have installed Interpoint for a SQL Server database, perform the following procedures.

To create the text file

1. Log in to the database server where your Siebel Tier Collector is installed.

2. Create the following tab-delimited text file:

   *<precise_root>*\products\sql-server\

   *instance name*\etc\siebel_machines.txt

To obtain the list of Siebel servers

1. Log in to the SrvrMgr utility.

2. Execute the following command:

   list servers

   Each returned row represents a Siebel server.

To obtain the host machine name and network address

1. Copy the getAddress.vbs script from the Utilities\Siebel folder in the Precise v9.6 download or on the DVD to the server.

2. Run the following command:

   cscript //nologo getAddress.vbs ODBC data source Administrator user name Administrator passwords where:

ODBC data source is the name of the ODBC data source used by the Siebel server to connect to SQL Server. To determine its name, check the ConnectString value in the siebel.cfg configuration file.

Administrator user name and Administrator password are the user name and password required to connect to the SQL Server instance.

3. Repeat these steps for each Siebel server.

Following is an example for the content of the siebel_machines.txt file:

sieb1 APPSRV1 00096B26E66C

sieb2 APPSRV2 00096B26E67C

For more information Siebel, see the relevant section in the *Precise Interpoint for Siebel™ Installation Guide*.

# Sampling Siebel batch processes (Oracle)

If you have installed Interpoint for an Oracle database, perform the following procedures.

To create the text file

1. Log in to the database server where your Siebel Tier Collector is installed.

2. Create the following tab-delimited text file:

| | |
|---|---|
| **Windows** | `<precise_root>\products\oracle\SID\etc\siebel_machines.txt` |
| **UNIX** | `<precise_root>/products/oracle/SID/etc/siebel_machines.txt` |

To obtain the list of Siebel servers

1. Log in to the SrvrMgr utility.

2. Execute the following command:

list servers

The following is an example of the returned information:

SBLSRVR_NAME   HOST_NAME AsDev1      POOL-Siebel1

AsDev2   POOL-Siebel2

Each returned row represents a Siebel server. The first column represents the name of the Siebel server, the second column the corresponding physical machine name (but not necessarily as it appears in Oracle).

To obtain the host machine names

1. Connect to the Siebel database.

2. Run the following command:

SELECT DISTINCT MACHINE FROM V$SESSION;

The following is an example of the returned information:

Machine

PRECISE\POOL-Siebel1

PRECISE\POOL-Siebel2

Each returned row represents a physical machine name.

If the command does not return any machine names, use the machine names provided in the HOST_NAME

column in the table of Siebel servers.

Following is an example for the content of the siebel_machines.txt file:

AsDev1 PRECISE\POOL-Siebel1

AsDev2 PRECISE\POOL-Siebel2

For more information Siebel, see the relevant section in the *Precise Interpoint for Siebel™ Installation Guide*.

**14**

# Post-Installation Tasks for Operating System Tier Collectors

This section includes the following topics:

- [Post-installation action items](#)

## Post-installation action items

When the Operating System Tier is installed, post-installation tasks may be required. Follow any instructions that might appear on the Action Items screen.

# Post-Installation Tasks for Other Tier Collectors

This section includes the following topics:

- [Post-installation action items](#)

## Post-installation action items

When the Other Tier Collectors have been installed, post-installation tasks are required if any Insight Savvies were running during the installation process.

Perform all post-installation tasks by following the instructions on the Action Items screen and click **Finish**.

**16**

# Post-Installation Tasks for
# the Precise for vCenter Server

This section includes the following topics:

- [Post-installation action items](#)

## Post-installation action items

When the vCenter Server is installed, post-installation tasks may be required. Follow any instructions that might appear on the Action Items screen.

# Post-Installation Tasks for the Tuxedo Tier Collector

This section includes the following topics:

- [Post-installation action items](#)

## Post-installation action items

When the Tuxedo Tier Collectors have been installed, post-installation tasks may be required.

- Run the following commands only if they have not been previously run for other Tuxedo instances on this server and this server is a UNIX server:

  Log in as root user and run the following commands:

  ```
  cd <precise_root>/products/insight/install
  ./tuxedo_root.sh
  ```

- A Tuxedo information system can have one or more Tuxedo servers. On each Tuxedo server, you can have one or more Tuxedo domains. A Tuxedo domain has a logical definition: it is an instance holding common business units and handling client processes. The Tuxedo Tier should be installed for each Tuxedo domain. Therefore, if your Tuxedo information system has more than one domain, you should create a Tuxedo Tier instance for each domain.

| NOTE | It is recommended to install the Insight Savvy for Tuxedo in each Tuxedo domain. To find out how many Tuxedo domains are running on your Tuxedo server machine, count how many BBL processes are running on this machine. On Windows, look for BBL processes in the Windows Task Manager. On UNIX, run the following command: `ps -ef|grep BBL`. |
|------|---|

To perform Tuxedo Tier post-installation tasks

1. In the Tuxedo Tier - Installation dialog box, click **Next**.

2. In the Tuxedo Tier - Post-Installation dialog box, follow the instructions to perform all post-installation tasks.

3. Click **Finish**.

**18**

# Post-Installation Tasks for WebSphere MQ Tier Collectors

This section includes the following topics:

- Post-installation action items

## Post-installation action items

Once the WebSphere MQ Tier Collectors have been installed, the following post-installation task may be required:

### Setting the Queue Manager attributes

To set Queue Manager attributes on Windows

1. Go to **Start>Programs>IBM WebSphere MQ** and open the MQ Services utility.

2. Right-click the specific queue manager and select **Properties**.

3. On the Exits tab, click **Add**.

4. Set the attributes as follows:

    o Name: VeritasI3Agent

    o Function: PSI_EntryPoint

    o Module: *<precise_root>*\products\insight\bin\psi_mq_exits.dll

    o [V] Data: **%ID%**

    o Sequence: **100**

    o where %*ID*% is a hexadecimal number, such as 0x 920D212F.

5. Stop and restart the queue manager.

To set Queue Manager attributes on UNIX

1. Log in as the mqm user.

2. Copy all psi_mq_exits* files from the <*precise_root*>/products/insight/bin/ folder to the *mq lib* folder.

3. Open the qm.ini file of the relevant queue manager, located in the *mq home/qmgrs* folder, and add the following lines:

    ```
    ApiExitLocal:
         Sequence=100
      Function=PSI_EntryPoint
      Module=mq lib/psi_mq_exitsName=VeritasI3Agent
      Data=%KEY%
    ```
    where *%KEY%* is a hexadecimal number, such as 0x920D212F.

4. Run the following command to stop the queue manager:

    endmqm queue manager name

5. Run the following command to restart the queue manager:

    strmqm queue manager name

# 19

# Post-Installation Tasks for Precise in a failover application

This section includes the following topics:

- [Generating Precise startup and shutdown scripts](#)
- [Configuring Precise services for Precise for SQL Server failovers](#)

## Post-installation tasks for the SQL Server Tier Collector in active/passive configurations

Installing the SQL Server Tier Collector in active/passive configurations requires post-installation tasks.

To perform post-installation tasks for SQL Server

1. Run the following command:

   `products\sql-server\install\pssq_install`

2. Press **Enter** and then type the value of the instance name as it appears in the `products\sql-server` folder.

3. If Precise for Storage is installed for an SQL Server database:

   a. Run the following command:

   `products\sql-server\install\pssq_symmetrix_install`

   b. Press **Enter** and then type **SYMMETRIX**.

4. Change the Startup type of each created service to Manual.

## Post-installation tasks for the Sybase Tier Collector in active/passive configurations

Installing the Sybase Tier Collector in active/passive configurations requires post-installation tasks.

To perform post-installation tasks for Sybase

1. Run the following command:

   `products\sybase\install\pssy_install`

2. Press **Enter** and then type the value of the instance name as it appears in the `products\sybase` folder.

3. Change the Startup type of each created service to Manual.

## Post-installation tasks for the Microsoft .NET Tier Collector in active/passive configurations

Installing the Microsoft .NET Tier Collector in active/passive configurations requires post-installation tasks.

To perform post-installation tasks for Microsoft .NET

1. Install at least one .NET instance on the active server via the Precise Installer.

2. On the passive server, perform the following actions:

   a. Run the products\dotnet\install\psdn_col_service_install.batcommand.

  b. Run the products\dotnet\install\psdn_injector.execommand.

  c. Run the regsvr32 /s products\dotnet\bin\i4np.dllcommand.

  d. Run the products\dotnet\install\psdn_gac_all.batcommand.

3. After performing these steps, you can continue to install more .NET instances on either server.

# Post-installation tasks for Insight Savvies for OS and Network in active/passive configurations

Installing the Insight Savvies for OS and Network in active/passive configurations requires post-installation tasks.

To perform post-installation tasks for Insight Savvies for OS and Network

1. Log in to the passive server.

2. Change to the *<precise_root>*/products/insightfolder.

3. On Windows only, change to the `<precise_root>/products/insight/host` folder and run the following command:

```
echo "<precise_root>"|products\insight\install\i3_general.exe -install
```
Do not type blanks between `<precise_root>` and the | character.

4. Only if Insight Savvy for Network is installed:

  a. Install the network driver on the passive physical server, as follows:

   Windows

    i. Log in to the passive server.

    ii. Change to the following folder:

     `\products\insight\host\install`

    iii. Run the reinstall_network_cards.batscript.

   UNIX

    i. Log in to the passive server as root user.

    ii. Run the `network_root.sh` script from the following folder:
     `.\products\insight\host\install`

5. Only if Insight Savvy for Network is installed:

  a. Include the IP address of the physical servers in the FocalPoint registry file on the server where Precise FocalPoint is installed, as follows:

    i. Open the following file:

     *<precise_root>*\products\i3fp\registry\products\insight\hosts\\*logical server name*.xml

    ii. Copy the existing physical server section to the <ips> section and change the IP address to the IP of the physical server.

     For example, for a cluster whose active server IP address is 10.1.1.1 and whose passive server IP address is 10.1.1.2, this section would look as follows:

```
<ips>
<child_1>
    <ip>10.1.1.1</ip>
</child_1>
<child_2>
    <ip>10.1.1.2</ip>
</child_2>
</ips>
```

    iii. Sync the registry changes by running the following command on the FocalPoint where you've updated the registry files:

   Windows

```
infra\bin\psin_cli.bat -action update-main-registry -i3-user <user>
-i3-clear-password <password> -registry-path
/registry/products/insight/hosts/<server>
```
   UNIX

```
./infra/bin/psin_cli.sh -action update-main-registry -i3-user <user>
-i3-clear-password <password> -registry-path
```

## Post-installation tasks for Tuxedo Tier agents in active/passive configurations

Installing Tuxedo Tier agents in active/passive configurations requires manual post-installation tasks.

When trying to find the Tuxedo shared memory address to attach during initialization, the Insight Savvy for Tuxedo uses the physical server name. If the logical server name is used during the installation, this may cause connection issues. To prevent connection issues, you need to configure the shared memory address in the registry file by adding the <tux_shmkey> parameter. This parameter takes on the hexadecimal value of the shared memory address as defined in the Tuxedo configuration file. For example:

```
*RESOURCE
S IPCKEY
100100
```

In this example, 100100 is the decimal string for the hexadecimal value 0x18704. The following is a complete Tuxedo configuration example from the Precise registry:

```
<simple>
        <tux_dir>/vip/Tuxedo71/sun4</tux_dir>
        <tux_version>71</tux_version>
        <tux_config>/vip/Tuxedo71/sun4/samples/atmi/simpapp/
        tuxconfig</tux_config>
        <tux_uid>7101</tux_uid>
        <tux_shmkey>0x18704</tux_shmkey>
        <network>
                <ports>
                        <child_1>
                                <port>0</por
                                        t>
                                </child_1>
                </ports>
        </network>
</simple>
```

To perform post-installation tasks for Tuxedo Tier agents

1.  Log in to the passive server and change to the `<precise_root>/products/insight` folder.

2.  Configure the shared memory address in the registry file to prevent connection issues, as follows:

    a.  To prevent connection issues, configure the shared memory address in the registry file, as follows:

        i.  Open the <precise_root>\products\i3fp\registry\instances\tuxedo\*logical server name*\\*instance name*.xml file.

        ii.  Under <instance name>, add the line: <tux_shmkey>0xXXXXXX</tux_shmkey>

        iii.  Sync the registry changes by running the following command on the FocalPoint where you've updated the registry files:

        UNIX

        ```
        ./infra/bin/psin_cli.sh -action update-main-registry -i3-user <user>
        -i3-clear-password <password> -registry-path
        /registry/instances/tuxedo/<server>/<instance>
        ```

        Windows

        ```
        infra\bin\psin_cli.bat -action update-main-registry -i3-user <user>
        -i3-clear-password <password> -registry-path
        /registry/instances/tuxedo/<server>/<instance>
        ```

    b.  On Windows only, change to the `<precise_root>/products/insight/host` folder and run the following command:

        ```
        echo "<precise_root>"|..\install\i3_general.exe -install
        ```

        Do not type blanks between `<precise_root >`and the | character.

## Installing the Oracle Tier agents in active/passive configurations

Installing Oracle Tier agents in active/passive configurations requires the following preconditions:

*   The monitored Oracle database is installed on both servers with the same instance ID.

*   Both databases share the same Oracle data via a shared file system and shared disks.

*   Oracle instances and servers are accessible using the logical name of the cluster.

*   Because only one database is active at a time, the Precise for Oracle agents monitor only the active instance at any given moment.

Pre-installation and installation tasks are not required, but post-installation tasks are mandatory if the Insight Savvy for Oracle is installed.

## Post-installation tasks for Oracle Tier agents in active/passive configurations

If you have installed the Insight Savvy for Oracle, you need to perform manual post-installation tasks.

To perform post-installation tasks for Oracle Tier agents

1. Log in to the passive server and change to the `<precise_root>`/products/insight folder.

2. On Windows only, change to the `<precise_root>`/products/insight/*host* folder and run the following command:

   ```
   echo "<precise_root>"|..\install\i3_general.exe -install
   ```

   Do not type blanks between `<precise_root>` and the | character.

## What happens in a failover situation (active/passive)

When the monitored application fails over to the standby server, all Precise agents must fail over, too. It is recommended that you add Precise shutdown and startup scripts to your automatic failover procedure.

See Generating Precise startup and shutdown scripts on page **Error! Bookmark not defined.**.

| NOTE | AdminPoint shows the status for Precise agents that run on the active server. Precise agents on the standby server are not represented. |
|---|---|

The following figure illustrates the live standby configuration before a failover.

**Figure B-1**    Live standby configuration before a failover
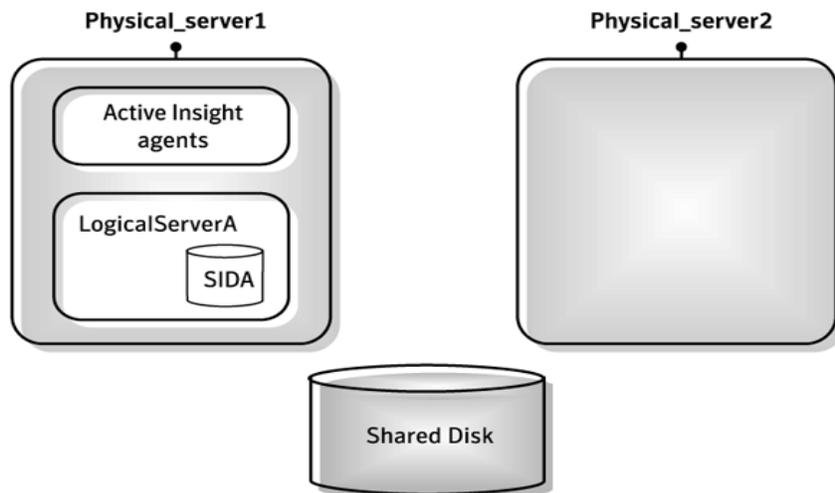


The following figure illustrates the live standby configuration after a failover.

**Figure B-2** Live standby configuration after a failover
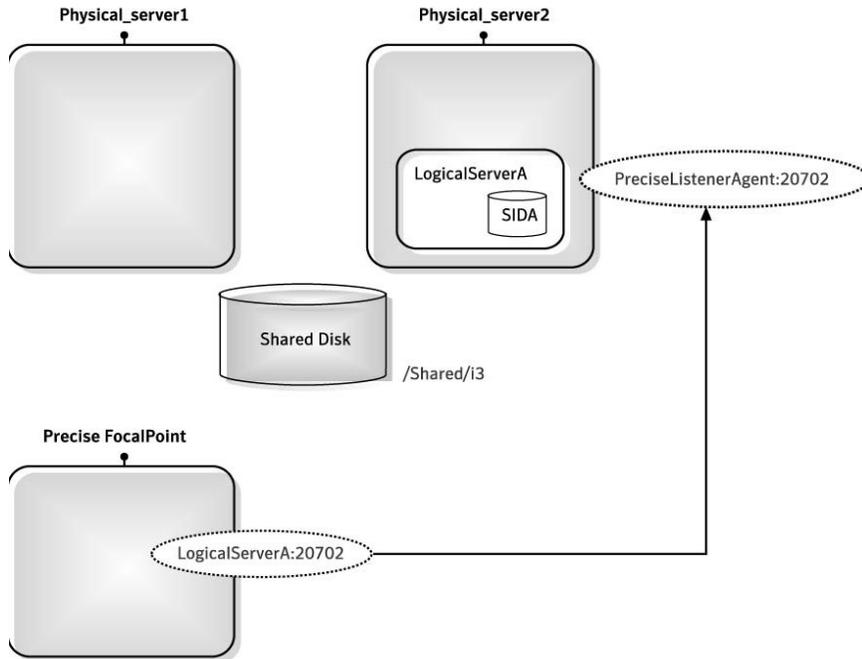
## Failover tasks in active/passive configurations

If you added the Precise shutdown and startup scripts to your automatic failover procedure, you do not need to perform any tasks when a failover occurs. Otherwise, you need to perform the following task.

To perform failover tasks in active/passive configurations

1. Log in to the servers that failed (if still alive) and stop all Precise agents.

2. Start all Precise agents on the now active server.

   | NOTE | Upon failover, you must stop all services before starting them on the active server. Verify that you stop the sentry service first. |
   |---|---|

| | |
|---|---|
| **abandonment rate** | In Precise for Web, a counter that keeps track of the percentage of users that abandon the loading of the Web page before it completes downloading. |
| **action** | An operation that Alerts FocalPoint automatically performs when detecting a warning or critical status for a specific metric. According to the defined action, Alerts FocalPoint opens a message box, sends an email or SNMP trap, or executes a program. |
| **action item (manual)** | Some installation steps cannot be executed automatically by Precise Installer. The administrator is required to execute them manually. Action Items are presented (when necessary) at end of installation or product update. |
| **AdminPoint** | The central administration console of Precise that facilitates the maintenance, configuration, and management of all installed Precise components, such as monitoring the status of all Precise agents and PMDB processes, getting license information, starting and stopping the agents, getting log data on agents and events, changing PMDB settings, and installing updates. |
| | See also PMDB. |
| | See also agent. |
| **advice** | In Precise for Oracle, an algorithm that is designed to recommend on gathering statistics, creating new indexes, change existing indexes, and add or delete hints to make Oracle's Optimizer choose a better access path and make the statement perform better. Can be activated from any DML (Data Manipulation Language) statement. |
| **agent** | A program that runs on a server machine to collect, process, or communicate performance information. The Precise installation consists of multiple agents. |
| **alert** | The state of an Alerts metric that has reached a near-critical or critical status. An alert is issued by Alerts, triggering an action and informing of an issue that has occurred or is likely to occur within the area sampled by the specific metric. |
| | See also action. |
| **Alerts** | These product provide alerts to problematic conditions before they turn into performance issues, based on predefined metrics and thresholds. Alerts can automatically perform an action, such as displaying a pop-up message, sending an email message or SNMP trap, or running a program. |
| **Alerts FocalPoint** | An agent that receives data from the InformPoint agents, stores it, and performs any action that has been user-defined for that specific alert, such as displaying a pop-up message, sending an email message or SNMP trap, or running a program. |
| | See also InformPoint. |
| **alert type** | In Alerts, the status of all metrics belonging to a metric group or a monitored instance, indicating the current performance level through colors. |
| | See also alert. |
| | See also metric. |
| **application server metric** | In Precise for J2EE, a metrics that is provided by the application server or by customer code. This can include metrics published by the Java Management Extension (JMX) APIs or vendor-specific APIs, such as IBM's Performance Management Interface (PMI). |
| **Tier** | The abbreviation for an application tier in a Precise application. ATier Tier contains one or more instances of the same technology and purpose. Application tiers do not necessarily correspond to distinct physical servers: in many cases, the tiers are logical, with one server running multiple Tiers or one Tier spans a cluster of servers. A Precise application may contain multiple Tiers on the same technology. For example, you may group J2EE instances (JVMs) into a J2EE Presentation Tier and a J2EE Business Logic Tier. Segmenting application response time into the contribution of individual application tiers is helpful in identifying the source of performance issues. Analyzing the performance and behavior of each tier separately is crucial for isolating the root causes of performance issues. |
| **cabinet** | In Precise for Oracle, Precise for SQL Server, DB2, and Precise for Sybase, the highest logical level in the SQL tab hierarchy. A cabinet contains folders and, within folders, statements. |

| | |
|---|---|
| **call path** | A subset of a stack trace including only those methods that have been instrumented. |
| **client cache** | In Precise for Web, a counter that keeps track of the percentage of requests taken from the client cache (http status 304). |
| **client-side collection** | See Web client browser-side. |
| **cluster** | A group of servers or instances that are configured to be treated as a single entity. In Oracle, all Oracle instances of the same Oracle database (RAC configuration) are treated as a cluster and presented in the Precise for Oracle UI as a Database. In J2EE, a group of JVMs that serve the same application can be defined as Cluster (even if they are not defined as a cluster in the application server configuration). All JVMs of the same cluster share the same configuration. |
| **collapsed access plan** | In Precise for SQL Server and Precise for Sybase, the access plan of a unique group of statements or batches that belong to the same collapsed statement or batch but have different access plans. This can differ due to the constants in the text of the original (not collapsed) statements. |
| | See also explained statement. |
| | See also collapsed statement. |
| **collapsed statement** | In Precise for Oracle, Precise for SQL Server, Precise for Sybase, and DB2, a statement whose constants are replaced with parameters. Each collapsed statement can have several access plans, according to the occurrences of its statements. |
| | See also collapsed access plan. |
| **Collector** | The program that runs on the monitored server to collect performance information. |
| | Some technologies allow a single Collector (agent) to serve multiple instances running on the same server. Other technologies require a dedicated Collector per monitored instance. |
| | In most cases, it must be installed on the monitored server. Collectors for SAP, SQL Server, and Sybase may reside on a remote server. |
| | See also agent. |
| **collector slice size** | In Precise for J2EE, the smallest unit of data presentation and storage. It can be configured as 30 seconds or five minutes. Regardless of the defined collector slice size, the most granular slice size presented on both the Precise for Web UI and the Precise for Microsoft .NET UI will be 15 minute intervals. |
| **completion rate** | In Precise for Web, a counter that keeps track of the percentage of users that completed the loading of a Web page. |
| **count** | The number of occurrences observed during a measurement interval. |
| **CPU time** | The average amount of time consumed by the operating system actively processing instructions on behalf of a running activity. |
| **critical status** | In Alerts, the status represented by a red bullet indicating that the value of the sampled metric has exceeded the near-critical and critical threshold values. |
| | See also metric. |
| **Cross-Tiers** | In Insight, Report Manager, and Alerts, a perspective that provides high-level information about the performance of all Tiers in an application, including operating system data. It provides a holistic view of the entire application and helps understand how the Tiers interact. |
| | See also Tier. |
| **current data** | In all databases products, a tab that presents Current Sessions and the SQL statements that were executed including statements that are still in progress. |
| **custom** | Represents executions of a type not specifically defined. |
| | See also work time. |
| **customized metric** | In Alerts, a user-defined metric that measures site-specific parameters. |
| **drilldown** | Within Precise products the filtering of analyzed data by clicking a specific entity. Then, additional information about the selected entity is presented, plus a breakdown of its activity by another entity. For example, drill down on a transaction and get an overtime graph of the selected transaction, plus a list of top users that executed the selected transaction. |
| **entry point** | Usually a top-level HTTP or EJB request. An entry point can originate when a user clicks in a browser or an E-commerce server invokes a remote EJB. Precise for J2EE makes a distinction between service request executions and other executions. The first HTTP or EJB invocation within a call path is designated as an entry point. |

| | |
|---|---|
| **application** | The highest Precise logical group. It may contain multiple Tiers of various technologies that serve an application together. For example, a Payroll Production application may contain all Web servers, application servers, transaction managers, databases, and servers that serve this application. Alternatively, it may contain any set of instances that form together an administrative group. Since Precise version 8.5, an instance can belong to more than one application. Let's say a single Oracle database serves two different applications plus the DBA wants to associate this database with a group of other databases under his responsibility regardless of served applications. In this case the Oracle database will be associated to three different Precise applications.

See also Tier.

See also instance. |
| **ERP Extension** | In Precise for Oracle, DB2, and Precise for SQL Server, an extension to the Collector that provides detailed information on the activities and resource consumption of packaged application components. It correlates the database information, and the packaged application information and lets you see users, transactions, reports, and other elements of ERP applications, such as Oracle Applications, SAP, PeopleSoft, and Siebel. |
| **event** | In Alerts, the occurrence of a sampling or progress. A sampling occurrence occurs every time a metric samples. A progress occurrence occurs when a metric's progress status is changed or when the investigated status reaches the end of the given investigation time. In AdminPoint, all occurrences reported by Precise agents, including informational events, warnings, and errors related to one of the agents. All events are shown in the Events view.

See also metric.

See also progress. |
| **executions** | In Precise for Oracle, Precise for SQL Server, DB2, and Precise for Sybase, the number of times a SQL statement was executed during the selected time frame. In SAP, the number of times a transaction was executed during the selected time frame. |
| **explained statement** | In Precise for Oracle, Precise for SQL Server, DB2, and Precise for Sybase, a statement whose access path (chosen by the RDBMS Optimizer) is clarified and translated into a visual display. Explained results include information on the objects referenced by the statement and the operations performed on these objects. |
| **extended collection** | In Precise for Oracle, a function that proactively specifies a future period during which Oracle activity data is collected and organized for subsequent analysis. Extended collections are an easy means to view collected information, assess application resource consumption, and identify bottlenecks that are inhibiting application performance and end-user productivity. |
| **Federated Precise** | Federated Precise is a version of Precise (starting with version 8.5) that can manage multiple Precise installations within unified AdminPoint screens, displaying and managing all applications, instances, and installations. |
| **findings** | A ranked list of top performance issues in the selected context. They are presented in Oracle, SQL Server, .NET, and J2EE. Each finding provides: a short explanation of the issue, detailed background information, and most important: one or more links to further investigate the issue and possibly get more granular findings on the selected context. |
| **first byte time** | In Precise for Web, a counter that keeps track of the time that it takes from the moment a new Web page is called until the first byte arrives back from the Web server. |
| **FocalPoint** | An agent that communicates with the Listeners installed on the monitored servers, receives data from the Collectors, periodically processes and stores this data in the PMDB, and serves UI requests. |
| **folder** | In Precise for SQL Server, Oracle, DB2, and Precise for Sybase, the intermediate logical level in the SQL tab hierarchy. Folders are grouped into cabinets and contain SQL statements.

See also cabinet. |
| **Framework Installer** | The application that facilitates the installation of Precise framework components. It can be invoked from the installation DVD to install a new Precise deployment. It can also be invoked from an existing Precise deployment to install an additional framework node and attach it to the originating Precise deployment.

See also Framework node. |
| **Framework node** | A set of FocalPoint agents that are installed together and manages a set of monitored instances. The performance data of these instances will be loaded into a dedicated PMDB. A single Precise deployment may contain multiple framework nodes (using a separate PMDB for each node). An application cannot span over multiple framework nodes. |

| | |
|---|---|
| **garbage collection** | An automatic process in the Java runtime application that periodically reclaims memory used by objects that are no longer referenced. The process can impact an application's performance while memory is being reclaimed. Java programmers may initiate garbage collection explicitly. |
| **grouper** | In Precise for Web, the identifier that is used to group other identifiers, such as sites or URLs. |
| **hint** | In Precise for Oracle, an instruction directed at the Oracle Optimizer that includes considerations for an execution plan. The Oracle Optimizer will build an execution plan based on the hint, ignoring its own set of considerations. |
| **hour group** | A unit that reflects the type and level of activity within the system at different times. By defining the times of the day that are peak and off-peak, or day and night, the performance analysis can be focused on those particular times of the day. If, for example, most performance issues occur within nighttime and weekend batches, it can be useful to focus only on them. |
| **InformPoint** | In Alerts, an agent that retrieves performance data from all installed Precise products, analyzes it, and sends an alert to the Alerts FocalPoint when the predefined thresholds are exceeded. |
| | See also agent. |
| **Insight** | The Precise product family that facilitates the process of monitoring and correlating system performance. It consists of Insight. |
| | See also Report Manager. |
| | See also FocalPoint. |
| **Insight FocalPoint** | An agent that receives performance information from Insight Savvies, which monitor the application. Insight FocalPoint then correlates, processes, and stores this information in a centralized location. The Insight performance history is stored in the PMDB. |
| | See also Savvy. |
| | See also PMDB. |
| **instance** | A monitored object of a specific technology. The following list specifies what constitutes an instance for the various supported technologies: J2EE - a Java Virtual Machine (JVM - a logical name set by the user), Microsoft .NET - a Common Language Runtime (CLR - a logical name set by the user), Oracle Applications - an Oracle Applications Form server, Oracle - an Oracle instance, SAP - an SAP system, SQL Server - an SQL instance, Sybase - a Sybase instance, Tuxedo - a Tuxedo domain, Web - a Web server, WebSphere MQ - an IBM WebSphere Queue Manager, DB2 - a DB2 non-partitioned database or a DB2 database partition. During installation, the instance is associated with one Tier and application. An Instance can be moved to a different application or associated with multiple applications without re-installation nor losing historical data. |
| | See also application. |
| **instance statistics** | See performance counter. |
| **instance/database changes** | In Precise for SQL Server and Oracle, the part of the Collect Schema Changes process that captures instance definition changes and database option changes and saves them in the PMDB. |
| **instrumentation** | The process of inserting fault-tolerant recording hooks in Java byte code, .NET MSIL, HTML pages, or other monitored components, resulting in the capture of performance metrics. In Precise for J2EE, a mechanism that enables collecting performance information when an application is executed. The process involves inserting special fault-tolerant recording hooks into application class objects. In WEB, the insertion of recording hooks into HTML pages can either be in memory (Dynamic Instrumentation) or file-based (Static Instrumentation). |
| **instrumentation context** | See invocation context. |
| **internal invocation** | The process of invoking a request from an HTTP request (Servlet or JSP) or an EJB. Precise for J2EE displays an internal invocation like any other invocation. |
| **invocation** | An execution of a J2EE entity (a Servlet, EJB, SQL Statement, method, and so on). When mentioned in plural (Executions), means the amount of times that the entity was executed. |
| **invocation context** | The context within which a method is invoked. For example, if Method A is invoked both from Method B and from Method C, there will be two different invocation contexts for the performance metrics collected for Method A, one for when it is invoked from Method B and one for when it is invoked from Method C. |
| | See also call path and instrumentation context. |
| **JRE (Java Runtime Application)** | As the runtime part of the Java software, the combination of the components that enable the execution of a Java program: a Java virtual machine, the core class libraries, and the files that form the Java platform. |

| | |
|---|---|
| **JSP (Java Server Page)** | An HTML page with special tags for Java scripting. An application server processes the tags and generates a Servlet. |
| **JVM (Java Virtual Machine)** | An instance of a JRE that executes Java programs. A server-side Java application server is itself a Java program that runs inside a JVM. Servlets, JSPs and EJBs are Java programs (applications) that run within the application server's JVM. Precise for J2EE monitors the JVM running the application server and the server-side Java applications within the application server. |
| **JVM Heap Memory** | The amount of real computer memory that is allocated to the JVM for executing Java programs. |
| **key metric** | In Alerts, a parameter that monitors a very important performance aspect. The status of the Oracle instance (up/down), for example, is crucial for system performance. If the instance is down, this is the first issue that needs to be solved. Marking a metric as key metric ensures that a critical alert raised for this metric receives top priority by the person that is responsible to handle alerts at any time. It can also help determine which alert to handle first in case of multiple alerts. In Alerts, key metrics are always displayed at the top of a metric table when it is sorted by alerts so that they get immediate attention. |
| **Listener** | The agent that facilitates the communication between the various Precise agents across different servers must be installed on every server where Collectors or FocalPoint agents are installed. The Listener allows communication with all other agents installed on the monitored server, while only the Precise Listener port is known by other servers. |
| **login name** | In Precise for SQL Server and Precise for Sybase, the session identifier that represents the credential used to connect to the database. When an ERP extension is installed, the user name of the packaged application's client overrides the login name. For example, when SAP extension is installed, the SAP user name overrides the login name. |
| **machine** | A session identifier. A machine as sampled, for example, by the Precise for SQL Server Collector is the identification of the machine where the client process executes. Machine is also sampled by Oracle, DB2 and Sybase. In Insight terminology Machine is called Client machine. |
| **Main framework node** | A main framework node is the single point for login and also serves as the Precise FocalPoint for the entire deployment. |
| **major collection count** | Number of estimated major garbage collection events that occurred during the last J2EE collector slice. A major garbage collection can stop the application while JVM heap memory is being reclaimed.<br>See also garbage collection. |
| **major garbage collection time** | Percentage of time spent by the JVM executing major garbage collection events during the displayed interval.<br>See also garbage collection. |
| **MBean** | A Java object that represents a manageable resource. In Precise for J2EE, MBeans, or Managed Beans, are used for application server metrics. |
| **memory logger interval** | The interval at which the Precise for J2EE Collector gathers JVM heap memory data. All snapshots of the memory logger's data collected according to the memory logger interval are summarized with counters in the current aggregation interval. The memory logger interval's time span is typically a small fraction of the aggregation interval. |
| **metric** | In Alerts, a query that helps measure performance in the application. Three types of metrics are available: System metrics relate to the internal resources, operations, and objects of the monitored infrastructure; application metrics reflect the way the applications perform; user-defined metrics can be customized to specifically relate to a site. When a metric's value exceeds one of the defined thresholds, its status changes to near-critical (yellow bullet) or critical (red bullet).<br>See also key metric. |
| **metric set** | In Alerts, a unit that groups metrics that measure related performance aspects. The following metric sets exist:<br>Status: includes metrics that alert to functional issues related to the instance.<br>Performance: includes metrics that alert to performance issues related to the instance.<br>Load: includes metrics that alert to instance-related load issues that may later cause errors or crashes.<br>Service: includes metrics that alert to instance-related SLA breaches. The metrics in this set are sampled by Insight.<br>Performance Trending: includes metrics that alert to potential future performance issues. The metrics in this set are sampled by Report Manager. |

| | |
|---|---|
| | Load Trending: includes metrics that alert to potential future load issues. The metrics in this set are sampled by Report Manager. |
| | Customized: includes user-defined metrics. |
| | Precise Status: includes metrics that alert to the near-critical or critical status of the installed Precise application. |
| **module** | A session identifier. In Oracle it contains the value of MODULE column in V$SESSION table. In DB2 it contains the command name of non-SQL statements or the package name for SQL statements. |
| **near-critical status** | In Alerts, a status indicating that the value of the sampled metric has exceeded the defined near-critical threshold. A near-critical status is indicated by a yellow bullet. |
| **network time** | In Precise for Web, a counter that keeps track of the time spent on network activity from the server side perspective. This includes the time to read the request from the network and the network time to send the response back to the client. |
| **node** | See Framework node. |
| | Note: when creating a new installation from DVD, it would be called "Framework." In the UI screens (columns in tables, choosing a system for a new application etc), we would call them "Nodes." For example: choose a node for the new application. |
| | When adding a new system within AdminPoint, it would be called a "Framework node". |
| **packaged application** | An application that is created and/or maintained by a third party and is not custom-built to one's specific needs. The following packaged applications have special treatment by Precise: SAP, Oracle Applications, PeopleSoft, and Siebel. They are harder to modify because the application code is either not available or hard to understand. |
| | See also ERP Extension. |
| **page size** | In Precise for Web, a counter that keeps track of the amount of data loaded from the server to display the page. |
| **page views** | In Precise for Web, a counter that keeps track of the number of Web pages viewed at a specific Web site during a selected time period. |
| **parent metric** | In Alerts, a joining of several child metrics. Each time a parent metric samples, it gathers data from a set of child metrics and presents it as a single metric query. The individual child metric values are displayed on the Thresholds tab of the Properties dialog box in Alerts. |
| | See also submetric. |
| **performance counter** | In Precise for SQL Server, a Windows performance counter as reported by the operating system. In Oracle, an instance level statistic as gathered from some Oracle V$ views. In Precise for Sybase, a sysmonitors counter collected by Sybase. In Precise for Web, an operating system or a Web server performance counter. |
| **PMDB** | The Precise data warehouse of performance and availability data. It can be hosted on an Oracle or SQL Server database. |
| **portal server** | An application server for Web-based applications that commonly provide personalization, single sign-on, and content aggregation from different sources and that host the presentation layer of Information Systems. Aggregation is the action of integrating content from different sources within a Web page. A portal may have sophisticated personalization features to provide customized content to users. Portal pages may have a different set of portlets creating content for different users. |
| **portlet** | A Java-based Web component, managed by a portlet container, that processes requests and generates dynamic content. Portlets are used by portals as pluggable user interface components that provide a presentation layer to Information Systems. |
| **Precise deployment** | An independent Precise system. It contains and manages various agent types and provides centralized monitoring and administration. A Precise deployment may contain multiple framework nodes. One of them is defined as the main framework node and it manages all other nodes of the Precise deployment. |
| **Precise FocalPoint** | See Precise FocalPoint agent. |
| **Precise FocalPoint agent** | An agent that manages all agents in a single Precise deployment. Additional product FocalPoint agents manage specific technologies and resources. |
| **Precise Proxy agent** | When multiple Framework Nodes are managed by a single Precise deployment, the Precise FocalPoint agent manages all of the agents of the main Framework node, while every other framework node has a Precise Proxy agent to manage all its agents. |

| | |
|---|---|
| **Precise_root** | This is the term used in a path for the Precise installation folder. The terms *<i3_root>* or i3 root can appear in text too. |
| **program** | A session identifier in Precise for Oracle, DB2, Precise for SQL Server, Precise for Sybase, Insight, and Alerts. A program as sampled, for example, by the Precise for SQL Server Collector, is the name of the executable that connects to the database. Applications that do not set the application name have N/A as program. When an SAP extension is installed, the SAP transaction overrides the program, and it may change during the application's lifetime. When a Siebel extension is installed, the Siebel views override the program, and it may change during the application's lifetime. In Insight, program is the name of an executable as recognized by the operating system. If an executable is invoked from a script (a batch or a shell), the script is displayed as the command entity. In Alerts, program is part of the customized metrics definition and is the name of the executable or stored procedure executed in the database that runs when the metric is sampled. In action definitions, program is the name of the executable that will run if the metric exceeds its threshold. |
| **progress** | In Alerts, the management state of a metric for which an alert has been raised. The following statuses exist: |
| | Open: An alert is raised. |
| | Investigated: The alert is taken care of. |
| | Closed: The issue has been solved. |
| **Proxy FocalPoint** | See Precise Proxy agent |
| **recommended index** | In Precise for SQL Server, a function that uses the Microsoft Index Tuning Wizard to recommend on adding indexes or statistics for a selected statement, batch, or table. Based on the results of this function, the Optimizer will choose a better access plan and make the respective statement or batch perform better. For statements and batches, recommendations are based on the content of the statement or batch. For tables, recommendations are based on all the statements that are stored in the PMDB, are executed during the selected time period, and have an average duration time greater than the value defined in the registry (where 0 is the default). |
| | See also advice. |
| **related SQL** | In Precise for Oracle, a generated statement that uses alternative syntax to access the database in different ways and returns the same output as the original statement. |
| **relative frequency** | Number of contributor executions per entry point. For example, if a service request calls three methods each time it is invoked, the Relative Frequency for the method is three executions per service request execution. Similarly, if a service request calls one method every other time it is invoked (half of the time), the Relative Frequency for the method is 0.5 executions per service request execution. |
| **rendering time** | In Precise for Web, a counter that keeps track of the time that it takes for a Web page to be loaded from the moment the first byte arrives until the Web page is fully loaded or the user interrupts or abandons the loading process. |
| **report** | A collection of queries, programming code, and layout settings that Report Manager executes to generate graphical results like tables and charts. |
| **Report Manager** | Uses historical information to identify problematic conditions, track long-term performance, volume trends and patterns, view availability issues over long periods of time and on different levels, compare the performance of similar systems, correlate between performance metrics of different products, assist in capacity planning, and generate demand-driven, user-defined reports. |
| **Report Manager FocalPoint** | An agent that examines the PMDB tables to produce scheduled performance reports. |
| | See also PMDB. |
| **report parameter** | In Report Manager, a keyword used in a report. Its value is set during the report execution. A parameter's value can be updated either permanently or for the current execution only. |
| **report property** | In Report Manager, the attributes that define a specific report, consisting of report parameters and scheduling information. |
| | See also report parameter. |
| **request** | In Precise for Web, a counter that keeps track of the number of HTTP requests sent for a viewed entity. |
| **request error** | In Precise for Web, a counter that keeps track of the percentage of requests completed with an HTTP error. |
| **sampling** | In Alerts, the process during which a metric queries a Precise product for a specific instance, retrieves values, and calculates the metric's alert level. |

| | |
|---|---|
| **sampling base** | In Alerts, the start time of a scheduled sampling process (by default Sunday, 00:00 AM). |
| **sampling period** | In specific Alerts metrics, the time frame during which statistical data is returned from other Precise products. Such a metric is for example Oracle's General Behavior metric, which returns database behavior for a certain period of time. |
| **sampling rate** | In Alerts, the frequency of a metric's regular sampling schedule. The sampling rate is measured in minutes. |
| **Savvy** | An Insight agent that collects Tier-specific performance data. |
| **scalability** | A system's ability to withstand load. For example, positive scalability means that the system continues to function properly even when it is called upon to service a larger number of users. |
| **schema changes** | In Precise for Oracle and Precise for SQL Server, a process that captures schema changes and saves them in the PMDB. In Precise for SQL Server, instance configuration parameters and database option changes are also captured. |
| **Script Installer** | A component of Precise Agent Installer that enables adding a small JavaScript script, also known as a Precise for Web browser-side agent, to the Web pages of your Web site. |
| **server** | Also called a host machine or server machine. The combination of a computer and associated operating system software that is accessed by multiple clients to perform requested functions and typically returns results to the clients. |
| **server-side collection** | In Precise for Web, the collecting of performance data from the Web server instance through the Web server agent. |
| **service + network time** | In Precise for Web, a counter that keeps track of the service and network time, including the total amount of time the request took to reach the server. |
| **Service Level Agreement** | See SLA (Service Level Agreement). |
| **response time** | In Web, the time elapsed from when a request is received by the server to the time a response is sent back to the computer that made that request. The response time is measured on the server side. In Precise for J2EE, the time it takes an invocation to complete execution. In other words, response time is the average length of time between the start time and end time of a Java method execution. For example, the SQL response time is the time it takes the JDBC method executing the SQL statement to be completed. The response time includes CPU and wait time. The response times are reported in the interval in which they complete execution. Though a contributor's average response time may be very small, the contributor may be called many times. As a result, a contributor's overall contribution to performance may be large even though its average response time is low. |
| **servlet** | A class that is loaded only once and for which the application server uses multithreading to process requests. The servlet generates an HTML page that is sent back to the Web browser. |
| **size** | In Precise for Web, a counter that keeps track of the bytes sent and received. |
| **SLA (Service Level Agreement)** | A formal definition of an information system's performance goals. Within Precise, an SLA consists of clauses corresponding to various system activities. Once a system's SLA is defined, its SLA compliance can be analyzed, and breaches can be isolated to identify their causes. |
| **socket** | An I/O abstraction layer that enables processes to communicate with each other, regardless of whether they are on the same server. Sockets are bi-directional FIFO pipes that look like regular file I/O to the developer with the abstraction layer handling all of the low-level communication details. |
| **StartPoint** | The opening page of Precise. It provides a quick overview of the application status and links to launch any of the Precise products. |
| **Statistics tab** | In Precise for Oracle, DB2, SQL Server, Sybase, J2EE, and Web this tab presents instance level statistics such as hit ratios, utilization, JMX metrics, logging, and more. |
| **submetric** | Also called child metric. In Alerts, the subquery of a parent metric. A child metric gathers its own data and combines it with the data gathered by other child metrics to form the result of the parent metric. Each child metric has its own thresholds and may be enabled or disabled individually.<br><br>See also parent metric. |
| **summary table** | A container that stores the information collected by the Precise agents and loads it into the PMDB. Summary tables store the same data at different levels of granularity: time slice, hourly, daily, weekly, and monthly. By storing data in multiple summary tables, Precise can present a detailed view and progressively higher-level views of the same data. Summary tables are particularly useful for data aging. A data purging |

policy can be implemented for each summary table so that detailed data is retained for short-term historical analyses while more summarized data is used for long-term analyses and trending.

**technology**

A technology identifies the monitored object. For example, Oracle, SQL Server, Sybase, and DB2 are different technologies, while all Web servers (such as: Apache, IIS, and WebSphere) are defined as a single Web technology.  A single monitored object can be monitored by two different technology's Collectors. For example, WebLogic server can be monitored by both a Web Collector and a J2EE Collector.

**throughput**

The number of average completions per second that are observed during an interval.

**time slice**

A unit used to break up long sessions into smaller time periods. The length of a time slice is preset and cannot be changed. It represents the maximum time that passes before the data collected can be displayed. For example, if the length of a time slice is 15 minutes, the collection is only updated at 15-minute intervals. The length of a time slice is different for each technology: In Precise for Oracle, Precise for SQL Server, Precise for Sybase, DB2, Web, .NET, MQ, Tuxedo, Network, and OS, a time slice is 15 minutes. In Precise for J2EE, a time slice is either 30 seconds or 5 minutes. In Precise for SAP, a time slice is 5 minutes.

See also collector slice size.

**URI (Uniform Resource Identifier)** The relative path to a resource after the location (network node) is found.

**URL mapping**

In Precise for Web, a function that defines rules that map URLs (Uniform Resource Locators) with dynamic parameters originating from a specific domain to a format that identifies the Web pages and prevents them from having different URLs.

**Web client browser-side**

In Precise for Web, the gathering of performance data from the browser running on the desktop of the Web application's user, through the static or dynamic instrumentation of Web pages.

**Web server**

A program that receives client requests for Web pages, retrieves the static pages and/or issues a request for dynamic page creation to an application server, and sends the pages to the client. In Precise for Web, a computer that delivers (serves up) Web pages. Every Web server has an IP address and possibly a domain name.

**work time**

The time spent in the specific entity excluding time spent by called (other) entities. In J2EE, work time of a method is calculated as the response time of that method minus the response time of the methods it invoked.

**work type**

In Precise for Web, a counter that keeps track of the type of the Web instance, such as Web, PeopleSoft, SAP, or Siebel.

**tab**

A display unit in Precise products. All tabs display data from different perspectives. For example, in Precise for Oracle, Precise for SQL Server, and Precise for Sybase, the Current tab shows information on the sessions currently active in an application, and the Objects tab displays information on Oracle or SQL Server database objects that can be used to understand relationships and associations between database schema objects.