Manually executing post-installation action items

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:

> 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_NAME> -DisClustered=true

    > 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
  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"
     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"

     > 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>.

  2. Follow the instructions in the command output to modify the Java command line of the JVM to be monitored.
  3. Add the -DisClustered=false flag to the arguments from apladmin. For a clustered installation, specify 'true' for this flag.
  4. 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"

   > 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

   > 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 by editing the <WAS_INSTALL_ROOT>/profiles/<SERVER>/properties/server.policy file to include the following:

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

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

7. Restart the Application server.

   > 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.