EBS 12.2 Monitoring

Points to note before monitoring EBS 12.2

- The EBS 12.2 Oracle Applications tier is supported only when the Oracle tier is based on a 11g or a 12c non-containerized instance. EBS with an Oracle 12c containerized instance is not currently supported.
- When setting up the Oracle tier as part of setting up monitoring of a EBS environment, create the collector user directly in the Oracle database and set the user as 'Editions Enabled'. This can be done as follows:

  a. create user `<precise_col_username>` identified by enable editions
  b. grant create session to `<precise_col_username>`

  During Oracle collector installation, use this new account, and then select the **Use Existing User** option.

- The recommended order of installation of EBS tiers is:

  a. Install the Precise for Oracle collector agent.
  b. Install Web agent.
  c. Install J2EE agent.
  d. Install the Interpoint and instruct the installer to integrate with Insight.
  e. Perform the Interpoint for OA action items.

Installing EBS 12.2 monitoring for all tiers

Oracle Tier

Install the Precise for Oracle collector agent with a user account with **Editions Enabled** as mentioned in the previous section. If either the Precise PMDB Oracle instance or the EBS Oracle database instance is on Oracle 12c, switch to using the 12c tuner by modifying the registry or use the sqlnet.ora workaround as mentioned in the following IDERA solution: https://idera.secure.force.com/precise/50133000000f80n?srPos=0=501.

Web Tier

Install Precise for Web collector agent and monitor the Web Apache and Web Weblogic instances.

For Web Apache, the httpd.conf file is in a location such as:

```
/u01/install/VISION/fs1/FMW_Home/webtier/instances/EBS_web_EBSDB_OHS1/config/OHS/EBS_web_EBSDB
```

For Web Weblogic, add the following jar to the classpath of oacore and oafm servers:

```
/products/www/instances/<instance id>/lib/psww_filter.jar
```

In Web Weblogic, instance installation action items for adding filter content for oacore and oafm servers using the web.xml are in a location similar to:

```
/u01/install/VISION/fs1/FMW_Home/Oracle_EBS-app1/applications/oacore/html/WEB-INF/web.xml
```

```
/u01/install/VISION/fs1/FMW_Home/Oracle_EBS-app1/applications/oafm/webservices/WEB-INF/web.xml
```

J2EE Tier

Install Precise for J2EE collector and monitor the OA Core, FM, and Forms J2EE instances. If EBS is based on a Weblogic Server, then the update of some arguments is done via Weblogic Console as mentioned in the action items screen. Once the Forms J2EE instance is installed, you can include custom instrumentation by adding a custom invocation type `FORMS` to Custom.xml under `j2ee/config/<jvm>/` and the methods to instrument as well. Here is an example of adding the `startProcess` method of the class `oracle.forms.servlet.RunformProcess`.

```
<instrumenter-config>
     <custom-config>
          <java-classes>
               <java-class>
                    <invocation-type>
                         FORMS
                    </invocation-type>
                    <class-name>
                         oracle.forms.servlet.RunformProcess<
                    </class-name>
                    <methods>
                         <method>
                              <name>
                                   startProcess
                              </name>
                         </method>
                    </methods>
               </java-class>
          </java-classes>
     </custom-config>
</instrumenter-config>
```

You can add more methods as per your requirement. If you need all the methods under the class `oracle.forms.servlet.RunformProcess`, you can edit the custom.xml as following:

```
<instrumenter-config>
     <custom-config>
          <java-classes>
               <java-class>
                    <invocation-type>
                         FORMS
                    </invocation-type>
                    <class-name>
                         oracle.forms.servlet.RunformProcess
                    </class-name>
                    <methods>
                         <method>
                              <name>
                                   *
                              </name>
                              <apply-to-subtypes>
                                   true
                              </apply-to-subtypes>
                         </method>
                    </methods>
               </java-class>
          </java-classes>
     </custom-config>
</instrumenter-config>
```

OA Tier

Add the Oracle Applications tier instance and configure the Forms server jvm to be used by this instance. EBS 12 is based on Forms 10g and is configured to run in servlet mode by default.

Add the following tag to the registry under `instances/j2ee/<instance server>/<jvm>.xml`:

```
<oa_collection_mode>forms_servlet</oa_collection_mode>
```

Restart the Precise for J2EE agent and restart the FORMS JVM.

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 using the following steps.

- 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 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 that the file names may appear lowercase or uppercase (for example CUSTOM.pll). In the following task, all filenames are uppercase. If you copy and then 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. You need to know the Precise collector user name (i.e the Oracle user with which you installed the Oracle collector) as that will be used to prefix a function as follows -

```
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 re-generate 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 that for Oracle Applications version R12, replace the `f60gen` command with `frmcmp_batch`.
10. To verify the CUSTOM library change, run the following script: `/products/oracle//schema/pss_verify_oa_customer_pll.sh`

After configuring the interpoint refer to Precise Installation guide for Manually updating Oracle Applications system profile (Oracle applications 11.5 or higher).