

ORASentry

User Guide

Table of Contents

Overview 2
Prerequisites 3
Installation 4
Usage..... 6
Distribution 13
Annex 14

Overview

OraSentry is a "**Charityware**" that automates the **monitoring of Oracle databases' Alert log files**.

That monitoring is critical to ensure optimal operation of Oracle server. The "Alert.log" file contains all serious errors encountered by an Oracle instance (Lack of Space, block corruptions, deadlocks, internal errors ...). It is a plain text file that has to be manually edited in order to find potential errors. Usually a DBA checks the content of every instance's "Alert.log" file at least once daily. It can become very tedious if the number of instances is important.

With OraSentry, analysis of logged errors is greatly facilitated as information is raised **immediately, automatically and in a central place**. Furthermore OraSentry presents several keypoints that alleviate its deployment and its usage:

- OraSentry does not require any agent or installation on database server.
- OraSentry can be used on any operating systems supported by Sun JRE.
- OraSentry supports any versions of Oracle higher than 8.1.5
- OraSentry can monitor simultaneously more than hundred databases

OraSentry connects through a **regular JDBC connection** to each instance for which you want to monitor the Alert.log file. Then, it collects information that it displays on the central dashboard. The only requirement is a Java Runtime Environment (JRE) 1.4 or higher (available on Sun website).

Prerequisites

A Java Runtime Environment (JRE) version 1.4 or higher is required. You can download it freely from Sun website (<http://java.sun.com>). Follow instructions given by Sun for the install.

If you don't know if your system has already a JRE installed or if you are unsure of its version, you can follow instructions given in annex (1): Determining your JRE. Note that you can have more than one JRE per system.

Sun JRE is strongly recommended but you may use a JRE from another provider. Depending on the compliance of the JRE to java standards, OraSentry may or may not work properly.

For example, OraSentry doesn't run properly on GIJ (GNU libgcj) provided with some Linux distributions such as Fedora core 3.

Furthermore, be aware that OraSentry uses SunJCE (Java Cryptography Extension) library for password encryption. As it is not automatically bundled with other JRE you may have to perform an additional step for OraSentry to work fine (See annex (2): Installing SunJCE with "non-Sun" JRE).

Installation

Create a directory (for example "C:\OraSentry" on Windows or "/u01/app/OraSentry" on Linux/Unix) where you will copy OraSentry.jar file downloaded from OraSentry website.

- **Windows**

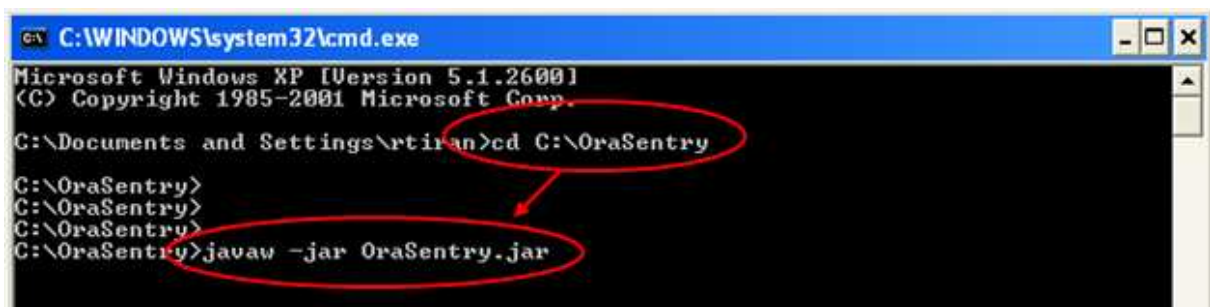
On Windows platform, JAR extension should be associated with Java executable (java.exe or javaw.exe) after successful installation of the JRE. Thereby, double-clicking OraSentry.jar file should execute OraSentry on your computer.

If it does not work, refer to annex (3): Associating JAR on Windows.

- **Windows / Command Line**

If you prefer to run OraSentry from a Command Prompt, set your working directory to the folder where OraSentry.jar is located (it's important to do so for OraSentry to retrieve its configuration files).

Then, invoke javaw.exe (or java.exe) with "-jar" option as shown below:



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\rtiran>cd C:\OraSentry
C:\OraSentry>
C:\OraSentry>
C:\OraSentry>javaw -jar OraSentry.jar
```

The screenshot shows a Windows Command Prompt window with a blue title bar. The text inside the window shows the user navigating to the directory C:\OraSentry and then running the command 'javaw -jar OraSentry.jar'. Two red circles are drawn around the directory path 'C:\OraSentry' and the command 'javaw -jar OraSentry.jar' to highlight them.

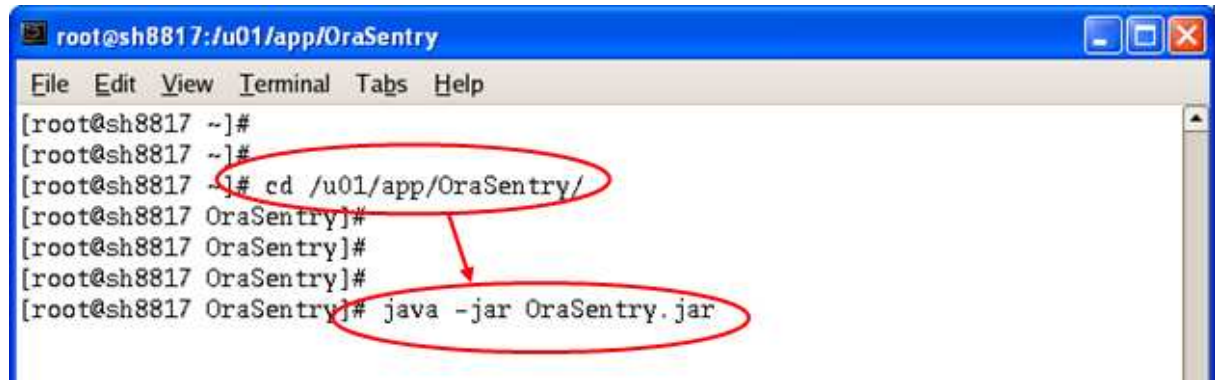
In that example, javaw.exe (or java.exe) location needs to be listed in the environment path (%PATH%). If not, you have to specify the full path to java executable on the command line.

- Unix / Linux

Procedure is very similar to the one for Windows Command line.

Set your working directory to the folder where OraSentry.jar is located (it's important to do so for OraSentry to retrieve its configuration files).

Then, invoke java executable with "-jar" option as shown below:

A terminal window titled "root@sh8817:/u01/app/OraSentry" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the following commands and prompts:

```
[root@sh8817 ~]#  
[root@sh8817 ~]#  
[root@sh8817 ~]# cd /u01/app/OraSentry/  
[root@sh8817 OraSentry]#  
[root@sh8817 OraSentry]#  
[root@sh8817 OraSentry]#  
[root@sh8817 OraSentry]# java -jar OraSentry.jar
```

Red circles highlight the command "cd /u01/app/OraSentry/" and the command "java -jar OraSentry.jar". A red arrow points from the first circle to the second.

In that example, java location needs to be listed in the environment path (\$PATH). If not, you have to specify the full path to java executable on the command line.

From a non-root session, if you get an error similar to:

Xlib: connection to "your_host:0.0" refused by server

Xlib: Client is not authorized to connect to Server

Just issue the command "xhost +" from a root session.

Note: On Sun Solaris operating system, you should be able to launch OraSentry by double-clicking OraSentry.jar.

Usage

OraSentry main window is constituted by two tabs:

- **Dashboard Tab**

On the left part of the window, each array line represents a monitored database:

Target	Status	Test Time
Instrumentation DB (INP01)	✓	03/11/2006 11:23:28
Instrumentation DB (INP02)	✓	03/11/2006 11:24:03
Instrumentation DB (INP03)	✓	03/11/2006 11:23:33
Instrumentation DB (INP04)	✓	03/11/2006 11:23:33

The status column can display 5 distinct icons:

- Blue Clock 🕒 : It's the default icon displayed at start-up by OraSentry while waiting for the first check to complete.
- Green Checkmark ✓ : No error (ORA-, Errors) found in target Alert.log
- Red Cross ✗ : At least one error (ORA-, Errors) found in target Alert.log
- Orange Exclamation Mark ! : Alert.log file has been modified outside Oracle regular logging process or file size is more than 30 MB. That mark is also displayed when OraSentry automatically archive the Alert.log file (if automatic archiving is activated). In that case, details about archiving are also displayed in the details panel.
- Black Eye 👁️ : OraSentry is unable to read Alert.log content (database down, lack of privileges, Alert.log removed...)

When selecting a line, details about the target and potential errors found in its Alert.log are displayed on the right side of the window.

The screenshot shows the OraSentry 4.5 application window. It features a 'Dashboard' and 'Settings' tab. A table lists various database targets with their status and test times. The 'Oracle Apps11i (Instance 1)' target is highlighted with a red 'X' in the status column. To the right, a details panel for this target is displayed, showing connection information, alert log file path, and a list of error messages. A red arrow points from the 'Acknowledge' button in the details panel to the 'Oracle Apps11i (Instance 1)' row in the table.

Target	Status	Test Time
Instrumentation DB (INP01)	✓	03/11/2006 12:03:49
Instrumentation DB (INP02)	✓	03/11/2006 12:04:13
Instrumentation DB (INP03)	✓	03/11/2006 12:03:26
Instrumentation DB (INP04)	✓	03/11/2006 12:03:53
Instrumentation DB (INP05)	✓	03/11/2006 12:04:12
Instrumentation DB (INP06)	✓	03/11/2006 12:04:23
Instrumentation DB (INP07)	✗	03/11/2006 12:03:21
Instrumentation DB (INP09)	✓	03/11/2006 12:04:14
Instrumentation DB (INP08)	✓	03/11/2006 12:03:22
Instrumentation DB (INP10)	✓	03/11/2006 11:59:27
Instrumentation DB (INP11)	✓	03/11/2006 11:59:32
eBusiness Suite (ERP1)	!	03/11/2006 12:03:25
Material Management (MAP01)	✓	03/11/2006 12:03:51
Material Management (MAP02)	✓	03/11/2006 12:03:42
Doc. Management (GEDP1)	✓	03/11/2006 12:04:07
Refer 2000 (REFP1)	✓	03/11/2006 12:03:46
Accounting DB (EPCP1)	✓	03/11/2006 12:02:12
Administrative DB (ADMP1)	✓	03/11/2006 12:03:31
Financial DB (HFMP1)	✓	03/11/2006 11:59:25
Oracle Apps11i (Instance 1)	✗	03/11/2006 12:04:23
Oracle Apps11i (Instance 2)	✓	03/11/2006 12:04:20
Human Resources (HRAP1)	✓	03/11/2006 11:59:32
Mutualisation DB (MUTP1)	✓	03/11/2006 11:59:35
Mutualisation DB (MUTP2)	✓	03/11/2006 12:04:21
9iAS Infrastructure (IASDB)	✓	03/11/2006 12:04:07
RMAN Repository (SYSP1)	✓	03/11/2006 12:04:15
Workflow DB (AWRP1)	✓	03/11/2006 12:03:20

Registered to OraSentry

Checking Target :
Oracle Apps11i (Instance 1)

Connection :
system@fr101sv0136.corp.tpnet.intra:1521:ERPV1_I1

Alert.log File :
File Name : alert_ERPV1_I1.log
Directory :
/dbms11/erpv1db/9iRac/admin/ERPV1_I1_fr101sv0136/bdump
File Size : 1.1 MB

Database :
SID : ERPV1_I1
Hosted on : fr101sv0136.corp.tpnet.intra
Oracle version : 9.2.0.5.0
Startup Time : 11/09/2006 16:14

Details :
[ORA-29702](#): error occurred in Cluster Group Service operation
[ORA-01157](#): cannot identify/lock data file 35 - see DBWR trace file
[ORA-01110](#): data file 35: '/dbms12/erpv1/data/tx_data9.dbf'
[ORA-27041](#): unable to open file
[ORA-1157](#) signalled during: ALTER DATABASE OPEN...

Acknowledge

If errors are found, you can acknowledge them by clicking the "Acknowledge" button on the details panel. By doing so, the status of the target goes back to a green checkmark until a new error is raised in the database Alert.log. As long as you don't acknowledge, each new error found in Alert.log is appended to previous ones.

- Documentation

If you click on a hyperlink, OraSentry opens a browser that displays the corresponding error definition in the Oracle documentation (hosted at <http://tahiti.oracle.com>). You need an Internet access for that functionality to work properly.

- Display order

The order that targets are displayed on the dashboard is the same than the one that they are listed in the "cnx" configuration file. You can re-order entries in the "cnx" file with a text editor. However, it is recommended to backup the "cnx" file before any modification.

- Internal errors

If internal errors are found (ORA-600 or ORA-7445) and if at least one of those errors is a published one (based on the "ORA-600 lookup tool" database), an extra section is displayed to indicate the references of associated Metalink notes:

```

Details :

Mon Oct 19 09:29:01 2009
Errors in file
/siebex00/ms/ora/udump/siebex002_ora_970794.trc:
ORA-00600: internal error code, arguments:
[kolasGetLength-1], [], [], [], [], [], []

Mon Oct 19 09:35:15 2009
Errors in file
/siebex00/ms/ora/udump/siebex002_ora_970794.trc:
ORA-00600: internal error code, arguments:
[kolasGetLength-1], [], [], [], [], [], []

ORA-600/7445 Metalink References :

See note 406814.1 for ORA-00600[kolasGetLength-1]

```

- Logical standby

In case a target database is a logical standby, OraSentry may throw a warning like the following:

```

Details :
It appears that this database is a Logical Standby operating
in guard mode ALL. OraSentry does not work in that
configuration. Please refer to documentation.

```

That is due to the fact that OraSentry needs to create a directory object within the standby database and that operation is prohibited due to the guard mode being set to "ALL". In that situation, the easiest is to downgrade the guard mode to "STANDBY" (SQL> alter database guard standby;).

- Error context

Starting with Orasentry version 5, a small magnifier glass icon is displayed in front of each error found. By clicking that icon, you access a new screen from where you can retrieve the "context" of the error - that is to say, an extract of the alert.log containing few lines around the error:

```
Tue Jul 20 15:46:30 2010
Errors in file
/siebec01/ms/ora/udump/siebec011_ora_2146488.trc:
ORA-00600: internal error code, arguments:
[kolasGetLength-1], [], [], [], [], [], [], []
```

The number of lines to retrieve before and after the error can be changed to fit your need. Clicking the retrieve button extracts the relevant part of the alert.log:

Target : SIEBEP01 - Inst1
 Connection : system@SIEPRC1:1521:SIEBEP011
 Alert.log File : alert_siebec011.log

Retrieve lines before and lines after:

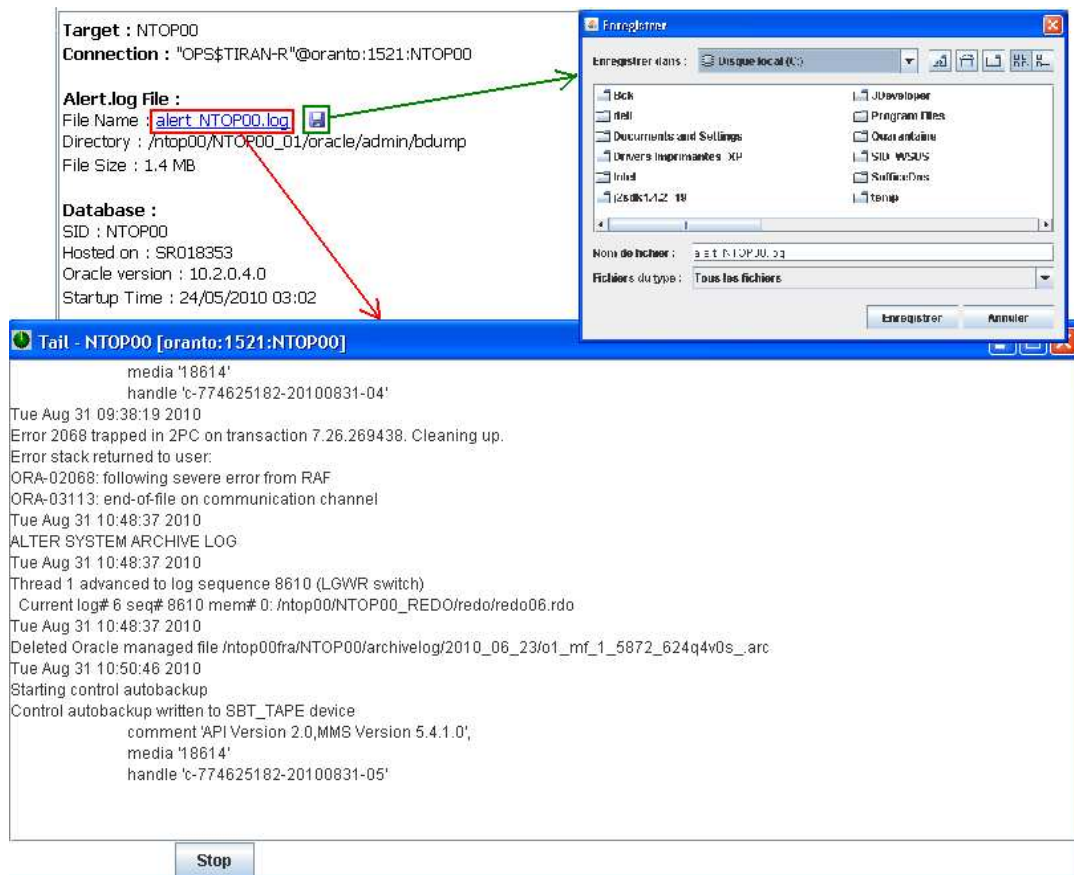
"ORA-00600: internal error code, arguments: [kolasGetLength-1], [], [], [], [], [], [], []"

```
Fri Jul 16 08:04:12 2010
alter database backup controlfile to trace as
'/siebec01/ms/ora/udump/controlfiletoTRACE.20100716.0804.trc'
Fri Jul 16 08:04:12 2010
Completed: alter database backup controlfile to trace as
'/siebec01/ms/ora/udump/controlfiletoTRACE.20100716.0804.trc'
Fri Jul 16 16:13:19 2010
Thread 1 advanced to log sequence 23
Current log# 4 seq# 23 mem# 0:
+DG_DATA/siebec01/dbs/rdo04.siebex00
Current log# 4 seq# 23 mem# 1:
+DG_INDEX/siebec01/dbf2/rdo04.siebex00
Fri Jul 16 16:15:53 2010
Errors in file /siebec01/ms/ora/udump/siebec011_ora_1355876.trc:
ORA-00600: internal error code, arguments:
[kolasGetLength-1], [], [], [], [], [], [], []
Fri Jul 16 16:15:58 2010
Trace dumping is performing id=[cdmp_20100716161558]
Fri Jul 16 17:01:33 2010
Thread 1 advanced to log sequence 24
Current log# 5 seq# 24 mem# 0:
+DG_INDEX/siebec01/dbf2/rdo05.siebex00
Current log# 5 seq# 24 mem# 1:
+DG_DATA/siebec01/dbs/rdo05.siebex00
Sat Jul 17 00:00:03 2010
Thread 1 advanced to log sequence 25
Current log# 6 seq# 25 mem# 0:
+DG_DATA/siebec01/dbs/rdo06.siebex00
Current log# 6 seq# 25 mem# 1:
+DG_INDEX/siebec01/dbf2/rdo06.siebex00
```

- "Tail" function and local transfer

OraSentry provides also alert file visualization functionality (similar to Unix "tail -f"). Just click on the hyperlink associated with the alert log filename to open a new window where is displayed the growing alert log file in real time.

Additionally, a copy of a remote alert log file can be retrieved locally by clicking the floppy icon next to the file name:



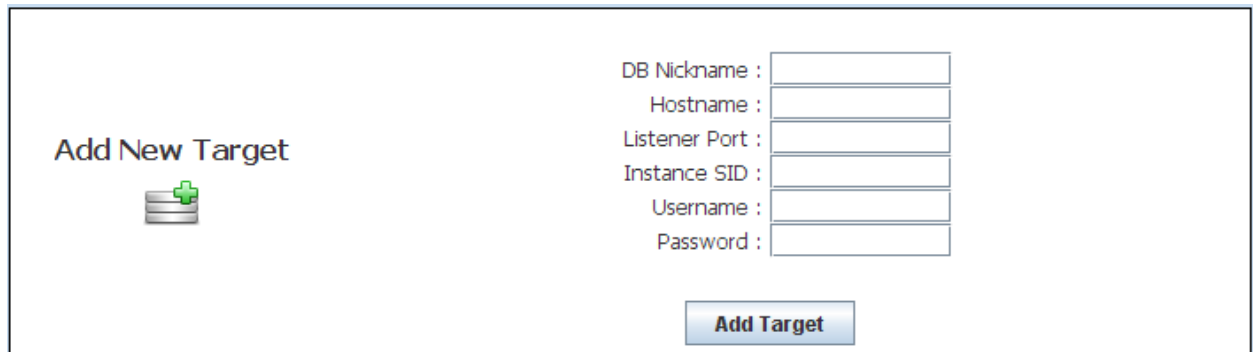
- **Settings Tab**

Settings Tab is separated in 3 distinct zones:

- Add new Target

In order to use OraSentry, you need first to specify which target databases you want to monitor.

Use the "Add New Target" zone of "Settings" tab to fill information regarding targets.



As Oracle JDBC thin client is used to establish connection between OraSentry and Target Database, you need to provide:

- Database server Hostname or IP address
- Listening port of listener (1521 by default)
- SID of the database

For better readability, you can specify a Database nickname to be displayed on the dashboard instead of JDBC thin connection string (hostname:port:sid).

Additionally, OraSentry needs to connect to target database with an account that has been granted at least the following privileges:

- Create session
- Create any directory
- Drop any directory
- Select on sys.v_\$instance
- Select on sys.v_\$database
- Select on sys.v_\$thread
- Select on sys.v_\$parameter
- Execute on sys.DBMS_UTILITY
- Execute on sys.DBMS_LOB
- Execute on sys.UTL_RAW
- Execute on sys.UTL_FILE

Of course, any account with DBA role would work fine as it has more privileges than needed. Be careful however to not use a SYS because, starting with Oracle 9i, connection to SYS should be made "as sysdba" (if O7_DICTIONARY_ACCESSIBILITY has default value of FALSE). Consequently, if using SYS, OraSentry connection would fail with "ORA-28009: Connection as SYS should be as SYSDBA or SYSOPER".

Note that the connection account cannot be OS authenticated (user identified externally), as JDBC thin client does not support external authentication.

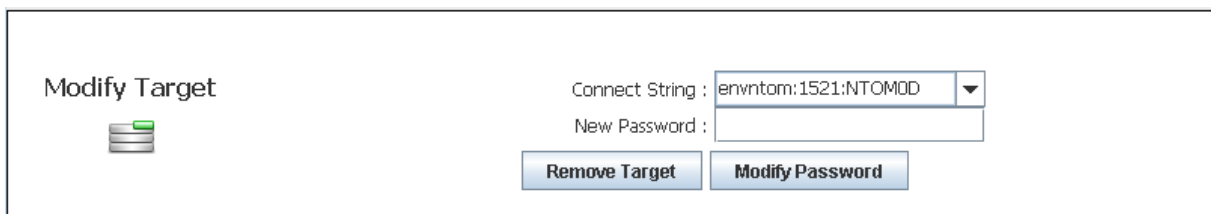
After adding one or more new targets, you need to restart OraSentry for the changes to be considered.

All connections information is stored in an encrypted configuration file named "cnx". **That file should remain in the same folder than OraSentry.jar archive.**

- Modify Target

To suppress monitoring of a target database Alert.log, use the "Modify Target" zone of "Settings" tab. Select the target from the dropdown list and click "Remove Target" button.

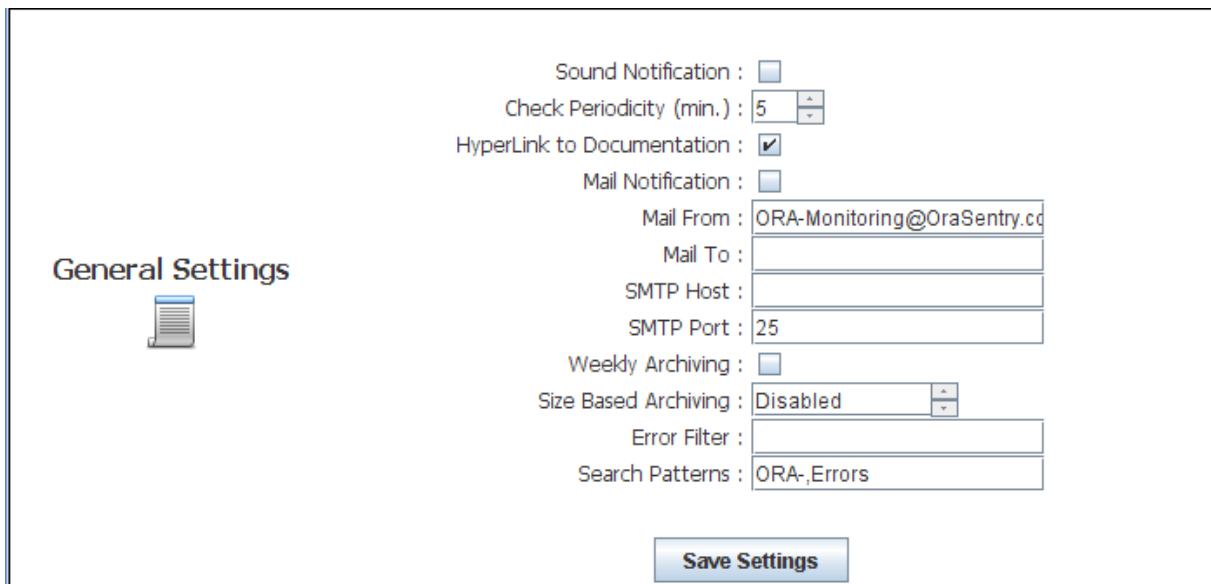
From that area, you can also modify the password that OraSentry uses to connect to a given target. Select the target in the dropdown list, fill the "New Password" field and click "Modify Password" button.



Once modifications are done, you need to restart OraSentry for the changes to be considered.

- General Settings

To modify checking parameters, use the "General Settings" zone of "Settings" tab.



You can modify the checking periodicity from 5 minutes (default) to 1 hour. You can also activate a sound notification to occur when a new error is found in a monitored Alert.log file.

Activation or Deactivation of the hyperlinks to Oracle documentation is also controlled from that zone. Note that you need an Internet connection for that functionality to work properly.

Additionally, OraSentry can send an email when a new error is detected. The email functionality is based on SMTP. In order to activate it, you need to fill in at least one recipient and the SMTP server hostname fields. You can specify several recipients by separating each address by a comma.

For debugging purposes, each time OraSentry send an email, the output from SMTP server is displayed on the console (if OraSentry is launched in console mode: DOS or Shell prompt). In case of problem, check that output to get additional information.

For Oracle databases 9iR2 (9.2) or higher, OraSentry can automatically "archive" Alert.log files. Archiving can be controlled by 2 methods:

- Size based: if Alert.log size exceeds a fixed limit (from 3 to 25 MB), it is automatically archived.
- Weekly archiving: if activated, each Sunday at midnight (DB Time), OraSentry archive monitored alert.log. Note that OraSentry must be running at that time for archiving to take place.

Note that the two methods can be activated simultaneously.

Archived files are renamed by replacing the extension ".log" by an extension based on the following pattern <WW>_<YY>_<Seq>.

WW & YY are Oracle date patterns, respectively week number and two digit year. <Seq> is just a sequence increasing after each archiving during a given week. For example, if archiving is enabled alertTEST.log file would be renamed to alertTEST.08_07_1 for the first archiving that would occur during week 8 of year 2007.

The search patterns can be customized. The patterns "ORA-" and "Errors" are hardcoded therefore they will always be searched however you can define you own patterns of interest. For instance, you can decide that you want OraSentry to raise an alert whenever a "checkpoint not complete" event is logged in an Alert.log. You can specify several patterns as a comma separated list.

OraSentry can optionally apply filtering on errors found. That is useful in case a particular ORA-xxxx is raised in the alert.log and you know that it can be safely ignored, it avoids that particular error "polluting" the dashboard. A good example is bug 4343398 on Oracle 10g Standard Edition for which you get periodically "fake" errors ORA-02097 and ORA-00439 in the Alert.log. Note that you can specify a coma separated list of errors that you want to be filtered.

After modifying settings, you need to restart OraSentry for the changes to be considered.

Settings information is stored in a configuration file named "cfg". **That file should remain in the same folder than OraSentry.jar archive.**

Distribution

Orasentry is distributed as a "**CharityWare**". If after an evaluation period, you wish to continue using it, please make **a donation to Unicef**.

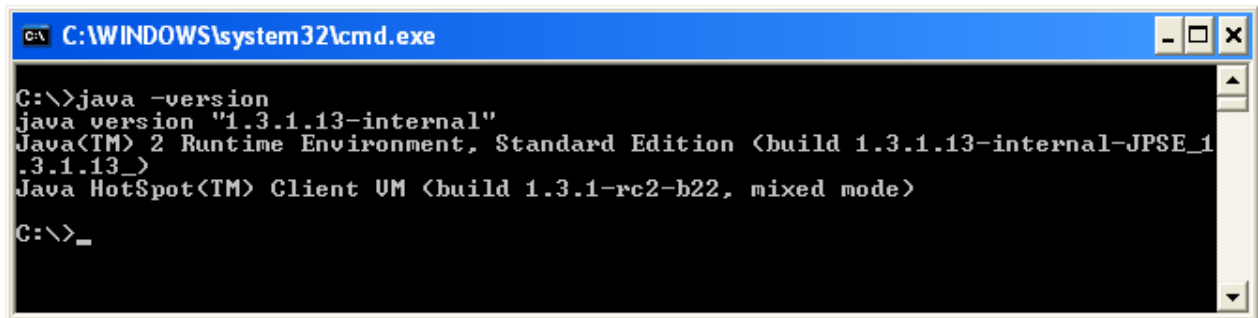
Annex

- **Determining your JRE**

From a command line prompt, enter "java -version".

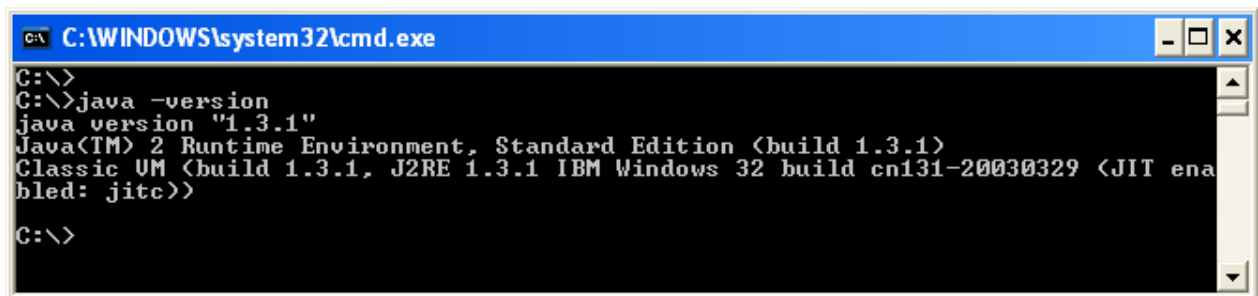
If a java executable is found in your environment PATH, you will get information about its provider and version.

For example:



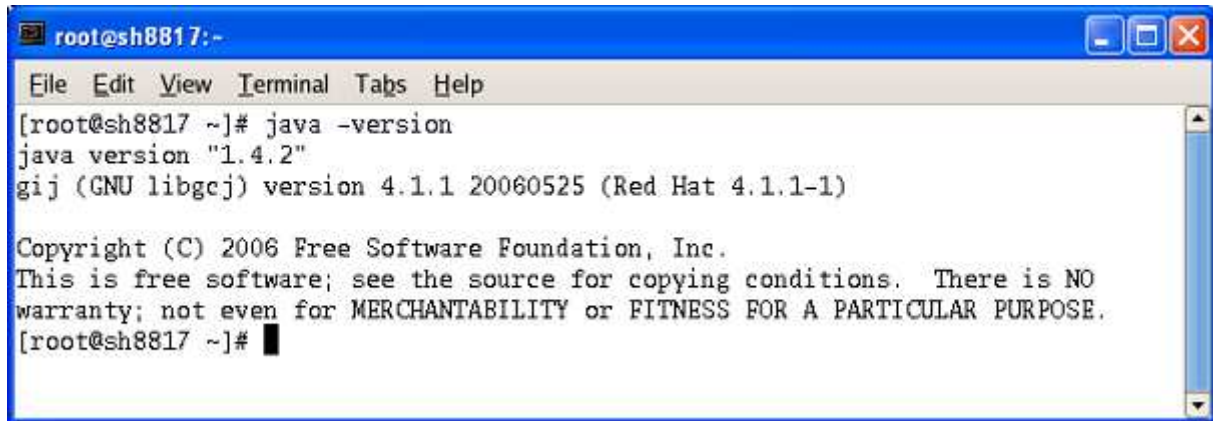
```
C:\WINDOWS\system32\cmd.exe
C:\>java -version
java version "1.3.1.13-internal"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.1.13-internal-JPSE_1
.3.1.13_)
Java HotSpot(TM) Client VM (build 1.3.1-rc2-b22, mixed mode)
C:\>_
```

On the screenshot above "Java HotSpot" indicates that we're working on a Sun JRE but its version 1.3 is not sufficiently recent for OraSentry.



```
C:\WINDOWS\system32\cmd.exe
C:\>
C:\>java -version
java version "1.3.1"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.1)
Classic VM (build 1.3.1, J2RE 1.3.1 IBM Windows 32 build cn131-20030329 <JIT ena
bled: jitc>)
C:\>
```

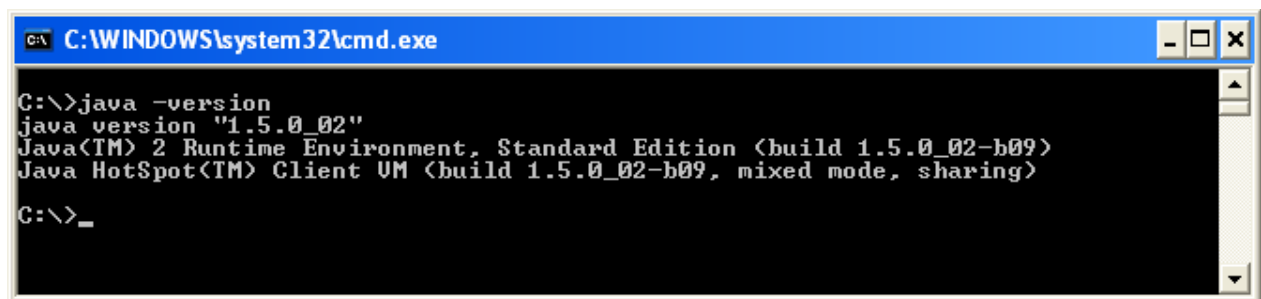
On the screenshot above "Classic VM" indicates that we're NOT working on a Sun JRE but on an IBM JRE. Furthermore, the JRE version 1.3 is not sufficiently recent for OraSentry.

A terminal window titled 'root@sh8817:-' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The command 'java -version' has been executed, resulting in the following output:

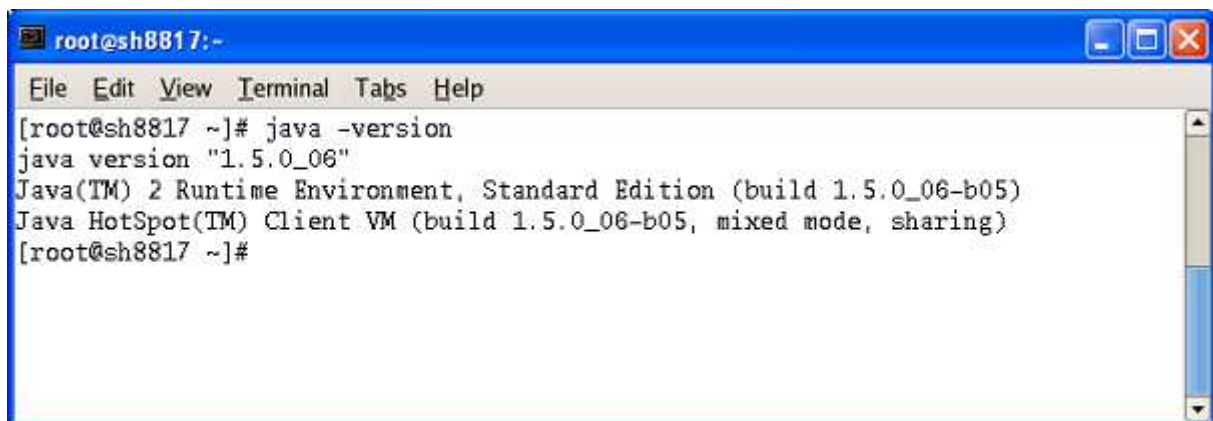
```
[root@sh8817 ~]# java -version
java version "1.4.2"
gij (GNU libgcj) version 4.1.1 20060525 (Red Hat 4.1.1-1)

Copyright (C) 2006 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
[root@sh8817 ~]#
```

On the screenshot above, the JRE version (1.4) is sufficient for OraSentry but it is not provided by SUN. It may not work properly.

A Windows command prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The command 'java -version' has been executed, resulting in the following output:

```
C:\>java -version
java version "1.5.0_02"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_02-b09)
Java HotSpot(TM) Client VM (build 1.5.0_02-b09, mixed mode, sharing)
C:\>_
```

A terminal window titled 'root@sh8817:-' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The command 'java -version' has been executed, resulting in the following output:

```
[root@sh8817 ~]# java -version
java version "1.5.0_06"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_06-b05)
Java HotSpot(TM) Client VM (build 1.5.0_06-b05, mixed mode, sharing)
[root@sh8817 ~]#
```

On the 2 screenshots above, JRE is from Sun and version is 1.5 that matches OraSentry needs.

Eventually, you may have a JRE that is not present in the environment PATH (or not in first position). To be sure you can search for JAVA executable on your machine (java.exe on Windows or java on Linux/Unix). If you find one, you can test its version as explained above (java -version). If a JRE matches requirement (1.4 or higher), you just have to setup your PATH variable to include its location or to call executable with full path.

- **Installing SunJCE with “non-Sun” JRE**

If you choose not to use Sun JRE to run OraSentry, you still have to get Sun JCE (Java Cryptography Extension) from:

<http://java.sun.com/products/jce/index.jsp>

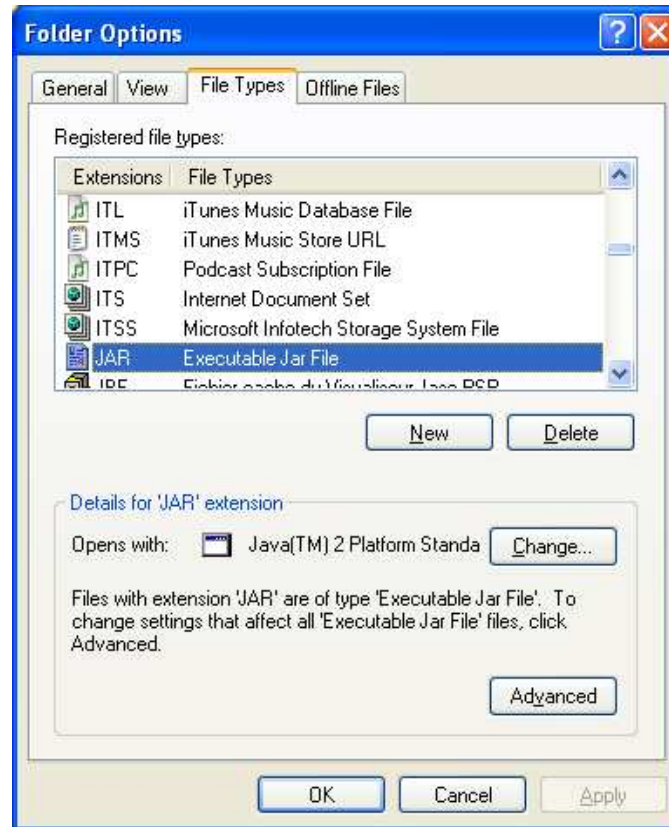
Uncompress and extract the downloaded file. The extracted “Lib” subfolder should contain:

- Jce.jar
- US_export_policy.jar
- Local_policy.jar
- Sunjce_provider.jar

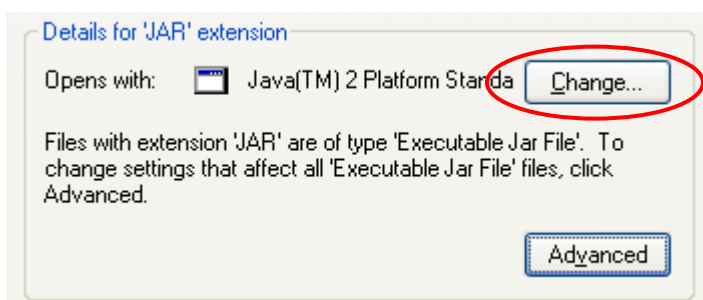
Copy these files to a location listed in your CLASSPATH or in the <JRE>/lib/ext directory (automatically part of the CLASSPATH).

- **Associating JAR extension on Windows XP/2000**

To setup association for JAR, open "Windows Explorer". From "Tools" menu, select "Folder Options" and click "File Types" tab. Scroll down to find JAR file type. On the bottom part of window, you should see the details for JAR extensions.

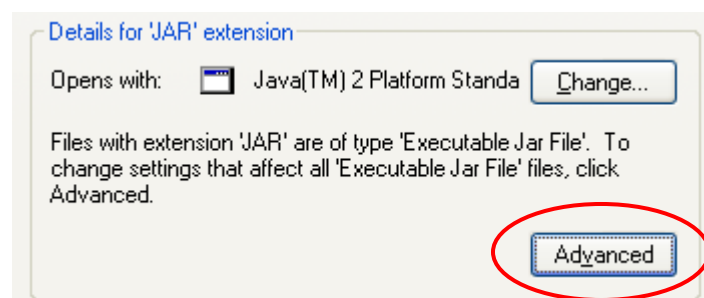


The "Opens with" part should be set to Java executable. If not, click on "Change" button and then browse to "javaw.exe" file located in JRE bin folder.

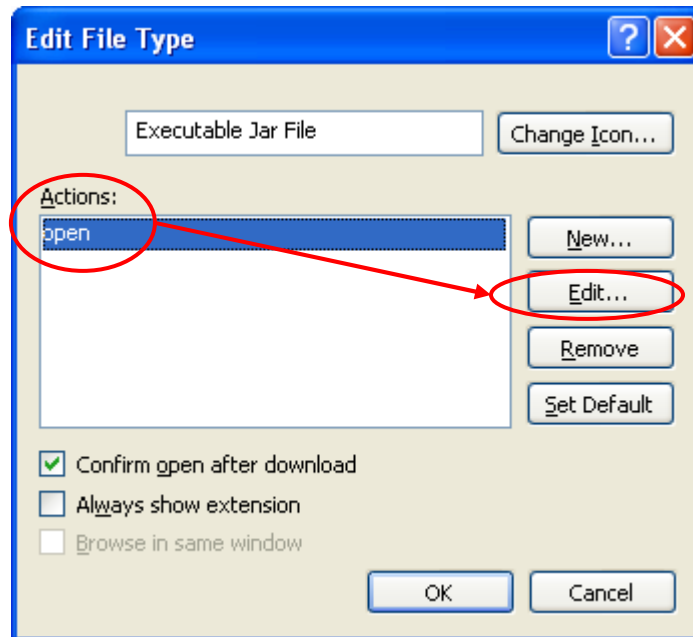


Click on change and select:
« <Path to JRE>\bin\javaw.exe »

Afterwards, click on "Advanced" button.



Under Actions, you should see "open". Select it and click on "Edit".



Edit the "Application used to perform action" field so it contains the following:

```
"C:\Program Files\Java\jre1.5.0_06\bin\javaw.exe" -jar "%1" %*
```

Change the path to match the javaw.exe path on your computer. Don't forget double quotes if the path contains spaces.

