Installation Guide Document

Product Name: caDSR Sentinel Tool
Version 3.2

Submitted to: NCI Center for Bioinformatics

Prepared by:

SCENPRO

Larry Hebel
101 West Renner Road, Suite 130
Richardson, TX 75082
May 29, 2006

Contact Information:

E-mail	sentinel@scenpro.com
Telephone	972/437-5001
FAX	972/437-3611

Revision History

Author(s)/Responsible Party	Rev No.	Date	Page(s)
Larry Hebel	Original	12/14/04	All
Larry Hebel	1	1/5/05	3, 4, 5
Larry Hebel	2	1/12/05	5
Larry Hebel	3	1/21/05	3, 4
Larry Hebel	4	2/9/05	3
Larry Hebel	5	2/23/05	3, 4, 5
Larry Hebel	6	3/17/05	3-11
Larry Hebel	7	9/2/05	All
Larry Hebel	8	9/16/05	4, 14
Larry Hebel	9	1/16/06	All
Larry Hebel	10	3/2/06	5
Larry Hebel	11	5/29/06	All

Table of Contents

1 Introduction

The purpose of this document is to detail the Installation Steps and Site Configuration Options for the caDSR Sentinel Tool Version 3.2.

The Sentinel Tool source and all related documentation can be accessed at http://gforge.nci.nih.gov/projects/sentinel. All caCORE companion projects can be referenced from http://gforge.nci.nih.gov.

Dependencies

1. These steps rely on the installation of the standard caCORE 3.2 Technology Stack for runtime and builds. **If the environment has not been previously used for one of the other caDSR Tools (e.g. CDE Curation, Browser, etc.), please refer to Section 5, Required Technology, for special notes and technology URL references.**

2. The caDSR database is installed and verified using Oracle SQL Plus or other appropriate database query tool.

3. The Oracle JAR should be copied from the Oracle Client installation to the JBoss 4.0.2 installation. The Sentinel Tool accesses the caDSR database from the web server using a 'thin' connection which does not require the installation of Oracle on the JBoss environment, only the Oracle JAR. The file, ojdbc14.jar, for Oracle 9 is included in the Sentinel Tool open source package.

4. The Oracle Application Server can **not** be executed on the same machine as JBoss with the Sentinel Tool deployed.

Document Definitions and Syntax

1. Download Package
   The zip file available from the NCICB GForge Sentinel project site noted above.

2. Folder Path Separator ('/')
   The folder path separator on Windows is the character '\' and on Unix is the character '/'. For simplicity all folder (directory) paths in this document use the '/' character. Change as needed for your environment.

3. Folder name 'sentinel'
   All examples represent the root for the extracted package as 'sentinel'.

4. Folder name 'JBoss'
   All examples represent the root for the installed JBoss 4.0.2 web server as 'JBoss'.

5. Use of quotes (" ")
   The appearance of quotes is restricted to examples in which the quote should be used as a typed key.

6. Use of apostrophes (‘ ‘)
   Apostrophes are restricted to clarification within this document and do not appear in examples or any data used for the installation of the software.

7. References to sentinel/conf/prod and build-prod
   For simplicity in the documentation all instructions assume a deployment to the Production environment, however the sentinel/conf folder contains several folders specific to different environments, i.e. DEV, QA, Stage and Production. If a deployment is being prepared to any environment other than Production, please substitute “sentinel/conf/prod” with the appropriate folder matching the deployment, e.g. “sentinel/conf/stage” when deploying to Stage.

8. Case sensitivity
   Unless otherwise stated all parameters, attributes, values, examples and XML elements are case sensitive.

2 Installation Steps

Should any errors occur during the completion of these steps, refer to Section 6 Troubleshooting, for assistance.

1. Extract the complete content of the **download package zip** file, preserving the folder names.

2. Optionally, modify 'sentinel/conf/prod/**DSRAlertDeploy.properties**'.

   This file contains site specific information to access the caDSR database should the JBoss configuration (oracle-ds.xml) or the cadsrsentinel.xml (see below) file be missing. To restore the properties to default values after making changes, extract it again from the download package. All documented properties must be set.

   'DSurl', this is the name of the caDSR database alias, e.g. CBDEV, if the Oracle thick client is required or the database connection description, e.g. <server>:<port>:<SID>, if the Oracle thin client is used. More details about the possible values of this property are written in Section 3.

   'DSusername', this is the default database account for access to the caDSR, it must have read and update authorizations.

   'DSpassword', this is the password required by the database account, above.

3. Optionally, modify 'sentinel/conf/prod/**cadsrsentinel.xml**'.

   This file is required by the Auto Run script (see below) and contains site specific information to access the caDSR database.

   'DSurl', this is the name of the caDSR database alias, e.g. CBDEV, if the Oracle thick client is required or the database connection description, e.g. <server>:<port>:<SID>, if the Oracle thin client is used.

   'DSusername', this is the default database account for access to the caDSR, it must have read and update authorizations.

   'DSpassword', this is the password required by the database account, above.

4. Optionally, modify 'sentinel/conf/web/**web.xml**'.

   Only the 'init-param' values need to be reviewed and changed. The default values assume the prior installation of the CDE Curation Tool. If the Curation Tool install has not occurred or if the default 'jbossDataSource' for the Curation Tool was changed then make appropriate changes in this web.xml file.

‘jbossDataSource’, this is the ‘<jndi-name>’ specified in the oracle-ds.xml for JBoss. This is the preferred and recommended specification for connectivity to the caDSR database. An example of the ‘<datasources>’ element to be placed in the oracle-ds.xml can be found in the ‘sentinel/conf/common/**oracle-ds-example.xml**’ file. The location of the oracle-ds.xml is defined by JBoss and is normally in the ‘jboss/server/default/conf’ folder.

‘DSusername’, this is the default database account for access to the caDSR, it must have read and update authorizations.

‘DSpassword’, this is the password required by the database account, above.

5. Execute ‘**ant build-prod**’ to build and run a basic automated test.
   Execute ’**ant –D”notest=true” build-prod**’ to build without a test.

   The Sentinel Tool contains a self test that may be executed as part of the build or bypassed as noted previously. The test performs basic checks on the database to ensure required tables and columns exist. It also checks the major functions of the software. These tests require the database exists and is accessible. Although the tests are not exhaustive and do not guarantee the system is bug free, they will expose any installation and basic feature errors.

   Upon completion a WAR file and two (2) JAR files are placed in the ‘sentinel/distrib/war’ folder. If the test is executed verify the content of ‘sentinel/conf/prod/ DSRAlertTest.properties’ prior to the build. Once the build is finished the test result reports are placed in ‘sentinel/test_results’.

6. Execute the sentinel/conf/prod/**load_options.sql** in your SQL IDE.

   This SQL script must be reviewed for content and edited as needed for the environment. It sets runtime configuration options within the caDSR database required by the Sentinel Tool. Please refer to Section 3 for a complete description of the script content.

   This script can be run from Eclipse, SQL Plus, Toad or any SQL IDE connected to the desired caDSR database instance. It may be modified and executed repeatedly as needed without repeating any of the previous installation steps above.

7. Optionally, execute ‘**ant build-doc**’ to create the JavaDoc.

   The resulting temporary files are placed in the ‘sentinel/tmp/cadsrsentinel/doc’ folder. If you wish to use the Sentinel Tool API you must execute this step to have the class and method documentation. The documentation is also collected and packaged in ‘sentinel/distrib/doc/sentinel_doc.zip’ by this build target.

8. Optionally, modify the **oracle-ds.xml** file

   As noted above this file contains the preferred caDSR connection configuration. An example of the ‘<datasources>’ element can be found in ‘sentinel/conf/common/oracle-ds-example.xml’. This is not a fully specified oracle-ds.xml file only the element needed to define the caDSR as a datasource. Refer to JBoss documentation for the full specification and location of the file if one does not already exist.

If the CDE Curation Tool is already installed and the default data source used, no change is needed to the oracle-ds.xml.

9. Modify the **log4j.xml** file for JBoss

   Example content can be found in 'sentinel/conf/common/**jboss_log4j.xml**'. Review all parameter values for correctness.

10. Move the **cadsrsentinel.war** from 'sentinel/distrib/war' to the 'JBoss/server/default/deploy' folder.

    Depending on the JBoss installation it may be necessary to restart the JBoss Application Server.

11. Restart **JBoss** if necessary.

    Depending on the installation JBoss may be configured to automatically deploy changes and new applications. Check the JBoss server.log to verify the deployment of the cadsrsentinel.war file. Also check the Sentinel log as specified in the log4j.xml file to verify correct specification and use of the oracle-ds.xml and web.xml settings.

12. Open a browser window and enter **http://<server name>/cadsrsentinel/do/logon** in the address.

    Replace <server name> with the appropriate reference to the server and port on which JBoss is listening. This should open the Sentinel Tool Login window and confirms JBoss has deployed the application.

13. Create a command script to execute the **Alert Auto Run** process.

    First verify the following dependant JAR files have been copied to the script directory.

    activation.jar
    asm.jar
    cacore32-client.jar
    cadsrsentinel.jar
    cglib-2.1.3.jar
    commons-logging-1.1.jar
    hibernate3.jar
    log4j-1.2.13.jar
    mail.jar
    ojdbc14.jar
    spring.jar

    With the exception of cadsrsentinel.jar, built by step 5 above, these files reside in the 'sentinel/lib' directory.

Next, copy the following lines into a text editor or start with the sample files included in the 'sentinel/distrib/readme' folder.

```
#!/bin/bash

DATE=`date +%Y%m%d`
JAVA_HOME=/usr/jdk1.5.0_06
BASE_DIR=/local/content/cadsrsentinel/bin

export JAVA_HOME BASE_DIR

ORACLE_HOME=/app/oracle/product/dbhome/9.2.0
PATH=$ORACLE_HOME/bin:$PATH
LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
TNS_ADMIN=$ORACLE_HOME/network/admin
JAVA_PARMS='-Xms512m -Xmx512m -XX:PermSize=64m'

export JAVA_PARMS ORACLE_HOME TNS_ADMIN PATH LD_LIBRARY_PATH

echo "Executing Auto Run for Sentinel Tool"
echo "Executing job as `id`"
echo "Executing on `date`"

find $BASE_DIR/../reports -mtime +20 -exec rm {} \;

$JAVA_HOME/bin/java -client $JAVA_PARMS -classpath $BASE_DIR/commons-logging-1.1.jar:$BASE_DIR/log4j-
1.2.13.jar:$BASE_DIR/mail.jar:$BASE_DIR/activation.jar:$BASE_DIR/ojdbc14.jar:$BASE_DIR/cacore32-
client.jar:$BASE_DIR/hibernate3.jar:$BASE_DIR/spring.jar:$BASE_DIR/cglib-
2.1.3.jar:$BASE_DIR/asm.jar:$BASE_DIR/cadsrsentinel.jar gov.nih.nci.cadsr.sentinel.tool.AutoProcessAlerts
$BASE_DIR/log4j.xml true $BASE_DIR/cadsrsentinel.xml
```

The path references will all need to be changed to match the deploy environment. Note the mail.jar and activation.jar allow access to the SMTP email server, the ojdbc14.jar allows access to the database, the log4j-1.2.13.jar provides logging, the cacore32-client.jar provides the caCORE 3.2 API and the cadsrsentinel.jar contains the AutoProcessAlerts.class file. The cadsrsentinel.jar is created by step 3, above, and is located in 'sentinel/distrib/war'. The other JAR files support the cacore32-client.jar.

The first program argument, '$BASE_DIR/log4j.xml', contains the logging configuration provided to the Auto Run and is **not** the same log4j.xml used by JBoss. A full and complete log4j.xml for Auto Run can be found in 'sentinel/conf/common/**log4j.xml**'. Review all elements prior to executing the script.

The second program argument, 'true', directs the Auto Run to update the Last Auto Run Timestamp on the Alert Definition within the caDSR. During development and debugging this should be changed to 'false'.

The third program argument, '$BASE_DIR/cadsrsentinel.xml', specifies the caDSR database connection values for the Auto Run. The file should have been configured in step 6 and refers to the same connection found in the oracle-ds.xml. The 'DSurl' in cadsrsentinel.xml is the same as the '<connection-url>' element specification of the <server>:<port>:<SID> values.

The log file as specified in the log4j.xml contains the status of the Auto Run execution.

Additional logging information is emailed to the Alert administrator(s) at the completion of the run.

14. Create a **scheduled task** to execute the command file created in step 8.

    This script should run once every 24 hours. For a server located in the Eastern Standard Time Zone the recommended schedule time is between 3:00 and 5:00 AM or immediately after completion of any off hour backups.

15. Optionally, copy the **cadsrsentinelapi.jar** to an application project.

    The Sentinel Tool API JAR file is created by step 7, above, and is located in 'sentinel/distrib/war'. Please note the distinction in the names of the JAR files in this folder. The content of the JAR is documented by executing step 7, above. Please refer to the Javadoc for package '**gov.nih.nci.cadsr.sentinel.util**' for features and use of the API.

16. Optionally, delete the '**sentinel**' folder and zip.

    After deployment the entire 'sentinel' folder and zip file may be deleted. The contents of 'sentinel/build', 'sentinel/tmp', 'sentinel/ospackage' and 'sentinel/test_results' are dynamic.

3 Site Configuration

As noted in section 2, the Sentinel Tool is built using the ANT script file build.xml. Possible build task values are 'build-proto', 'build-dev', 'build-qa', 'build-stage', 'build-prod', 'build-ospackage' and 'build-doc'. Each task pulls the required deploy properties from the matching folder to create the cadsrsentinel.war and cadsrsentinel.jar files, for example **'build-proto' uses the 'sentinel/conf/proto/DSRAlertDeploy.properties'** file. The optional command line value:

–D"notest=true"

will bypass the execution of all test tasks. Changes may be required to the ANT build script for paths and other environment specifics. Please try the build.xml without modifications first and then only change it if necessary.

Operating Properties

The caDSR Sentinel Tool has several site configuration values which must be set (Site Specific Properties) and some that may be changed (General Properties) should the defaults be inadequate. All of these reside in the '**tool_options.sql**' file and will be stored in a table within the caDSR database. The options contained in this script allow for dynamic changes without the need to build or deploy a cadsrsentinel.war file. The options are retrieved as needed and not held within the user session. In rare cases it may be necessary to logout of the Sentinel Tool and login again before a configuration option is used, most of the time this is not necessary.

The options are identified by the tool name and a property. For the Sentinel Tool the tool name in the script is always 'SENTINEL'. Details of the property and values of the configuration options follow. Each is identified by the property name followed by a description of its use. In addition to the information provided here there are extensive comments in the load_options.sql file concerning valid values for each property.

1. EMAIL.ADDR
   A valid email address, e.g. bsmith@yahoo.com. This address will receive log notifications for all manual and automated runs of the Sentinel Tool system if there is a problem retrieving emails for the Sentinel Administrator(s).

   > A dedicated folder may be created in the email client. Rules could then be created to redirect these messages to the desired folder. If using Outlook, for example, create a rule based on the content of the Subject line via the Outlook, Tools, Rules and Alerts menu. This email address may receive a significant number of emails from the Sentinel Tool system.

2. EMAIL.ADMIN.NAME
   A name to appear in the 'From' line of the email. This may be the Alert Administrators actual name or simply a title. The default is 'caDSR Alert Administrator'.

3. EMAIL.HOST
   A DNS or IP reference to the host machine for the SMTP service. Use 'localhost' when the same machine is used to run JBoss and the SMTP service.

4. EMAIL.HOST.USER
   The user name / account to use for access to the SMTP service. When using a forwarding server do not include this property in the SQL script.

5. EMAIL.HOST.PSWD
   The password necessary for the user name / account for access to the SMTP service. Only include this property when EMAIL.HOST.USER is included.

6. LINK.HTTP
   The prefix to add to the report file name so the recipients may access the content via a link in the email. When a complete folder name is used it must be terminated with a path separator character, e.g. 'file://DSRAlert Reports/' indicates a folder where 'file://tmp/Alert_Reports_' is not a folder.

7. OUTPUT.DIR
   The prefix to add to the report file names when creating the reports and log files. This should be relative to the server and may or may not match the LINK.HTTP value above, e.g. this may be '/temp/DSRAlert Reports/'. When a complete folder name is used it must be terminated with a path separator character as also noted for LINK.HTTP.

8. DB.NAME
   The string to display on the Alert Reports to identify the source of the data used to generate this report. This is a user friendly alternative to showing the DSurl contained in the DSRAlertDeploy.properties and cadsrsentinel.xml value.

9. EMAIL.INTRO
   The introductory paragraph to appear at the beginning of all recipient emails. Include additional contact information for the Alert Administrator or other specific instructions as desired.

10. EMAIL.ERROR
    The additional paragraph sent to recipients when errors are encountered while attempting to generate the reports. This always appears following the introductory paragraph within the body of the email.

11. EMAIL.SUBJECT
    The subject to appear on all emails sent from the Sentinel Tools. The next section contains details concerning situations when additional information is suffixed to this value, the default is 'caDSR Alert Report'.

12. RSVD.CS.LONG_NAME
    The Classification Scheme long name used as the parent to automatically created Classification Scheme Items used to create arbitrary collections of Administered Components (AC's) for reference on an Alert Definition.

The Alert Definition and Sentinel Tool user interface do not allow for the arbitrary selection of individual AC's. The Alert Definition Criteria only allows the identification of groups or sets of AC's via their Context, Protocol, Form, Classification Scheme (CS), Classification Scheme Item (CSI), etc. To provide for an arbitrary collection of AC's each must be associated to a known CSI and the Alert Definition must refer to the CSI in the Criteria. Refer to the RSVD.CSI.FORMAT configuration option for a description of how the CSI is determined. This option is used as the parent of the CSI acting as the arbitrary collection of desired AC's.

13. RSVD.CS.CS_IDSEQ
    The Classification Scheme (CS) database unique ID corresponding to the RSVD.CS.LONG_NAME.

    Both the long name and ID are used to verify the configuration option. See the description of RSVD.CS.LONG_NAME for more details.

14. RSVD.CSI.FORMAT
    The pattern format for the Classification Scheme Item (CSI) name created and used to identify a collection of arbitrary Administered Components (AC's).

    The collection of arbitrary AC's is specific to each user account. This ensures the collection is controlled by a single account and not a group. This name is expanded using macro substitutions (see comments in the load_options.sql file) and the caDSR is searched for a matching CSI. If not found it is created. The resulting CSI is then associated to individual AC's to form a collection which can be referenced using the CSI in the Criteria of an Alert Definition.

    The feature described by the RSVD.CS.* and RSVD.CSI.* properties is implemented in the Curation Tool user interface via the Monitor and Unmonitor buttons. Please refer to Curation Tool online help for more information.

15. ALERT.NAME.FORMAT
    The pattern format for automatically generated Alert Definition names.

    Using the Sentinel Tool API, an Alert Definition may be programmatically created outside the Sentinel Tool user interface. When an Alert Definition is created using this API, the name for the Alert Definition is set by expanding this property and performing macro substitutions as needed.

16. ADMIN.##
    A declaration of a user id to be given Sentinel Tool administrator privileges.

    The "##" can be any two (2) letters or numbers to uniquely identify the property entry in the options table. The recommended numbering is "00", "01", "02", etc. Other than uniquely identifying the entry in the table it serves no other purpose.

    The value for this property indicates the level and type of administrator privileges to grant. It may contain any of the following characters in any order.

'0' – allows the user to Edit, Run and Delete any Alert Definition. Normally a user may only Edit, Run and Delete Alert Definitions which they create. There must always be at least one user with this privilege.

'1' – makes the user a recipient of the Manual and Auto Run administrator logs. Normally a user only receives an Alert Report if included on the Recipient list and never sees the log messages generated by the report. There must always be at least one user with this privilege.

'2' – makes the user a recipient of the caDSR Statistics Report. This is a special report created by the Auto Run process and contains statistics on the content of the database used for maintenance and harmonization of the caDSR.

The user id for this option must exist in the database user account table and be a normal account for the use of any caDSR tool.

17. URL
    The URL to access the Sentinel Tool Login page, e.g. NCICB production is http://cadsrsentinel.nci.nih.gov.

18. REPORT.THRESHOLD.LIMIT
    The maximum number of rows allowed in an Alert Report before splitting the report into multiple files.

19. BROADCAST.EXCLUDE.CONTEXT.##.NAME
    BROADCAST.EXCLUDE.CONTEXT.##.CONTE_IDSEQ
    A declaration of a Context that is excluded from the Report Recipients Context Curators list.

    As with ADMIN.## the "##" can be any two (2) letters or numbers to uniquely identify the property entry in the options table. The recommended numbering is "00", "01", "02", etc. Other than uniquely identifying the entry in the table it serves no other purpose.

    The name and database id are required to validate the configuration option settings.

Database Configuration Properties

As noted above, the DSRAlertDeploy.properties contains default database connectivity information when oracle-ds.xml or cadsrsentinel.xml is not found. Following is additional information about the possible settings for these properties.

1. DSurl
   The database description, it can take either of two (2) forms, (1) the <server>:<port>:<sid> or (2) <alias>.

   The first form, <server>:<port>:<sid>, indicates a thin client connection and does not require the installation of Oracle on the application server machine. This is the same information that appears in the tnsnames.ora file for the alias description. This first form is recommended. The second form is only provided for backward compatibility.

   The second form, <alias>, indicates a thick client connection and requires the installation of the Oracle OCI on the application server machine. The database alias for the caDSR must appear in the Oracle tnsnames.ora file. Find the alias for the caDSR database then copy and paste it as the value for this attribute. **The alias name must appear exactly as entered in the tnsnames.ora file.** The tnsnames.ora file is normally located in %oracle_home%/network/admin/. The database alias appears to the left of the equal sign. For example, the alias 'cadsr.nci.nih.gov' may appear in tnsnames.ora as:

   ```
   cadsr.nci.nih.gov  =  (DESCRIPTION =
               (ADDRESS_LIST = …
               )
               (CONNECT_DATA = (SID = …)
               )
   )
   ```

   Often multiple database aliases are listed in the tnsnames.ora file, one for each Oracle Database for which access is granted. One of these must be configured for access to the caDSR database. If unsure which, please consult your Oracle DBA.

2. DSusername
   The user name / account with write access to the database named by DB_tnsname.

   It is strongly recommended this be an account which is used exclusively by the Sentinel Tool, hence the default value of 'sentinel'. By creating this dedicated account it becomes possible to identify specific database changes made by the logic independent of user interaction. In other words, when the Sentinel Tool code needs access to or makes changes in the database due to an Automated or Unattended execution, it will use this user name / account.

3. DSpassword
   The password to match the user name / account for database access.

Test Properties

Testing properties are provided in addition to the primary properties used during normal operation of the Sentinel Tool and Auto Run Process. The properties are only required if the automated tests are executed, see comments above in Section 2. The format and implementation follow the same pattern as above using the file DSRAlertTest.properties. The automated tests will create, modify and delete Sentinel Alert records. If all tests are successful the database is left in the same state as when the tests began. If any test is unsuccessful one or more Alert Definitions may remain in the database under the creator of the 'valid.userid' property detailed below. All test output, successful and unsuccessful, is stored in the 'sentinel/test_results' folder. The required properties are:

1. valid.userid
   A user id which has curate authority access to the caDSR database. As noted in the previous paragraph the test logic when successful will clean up all test data to ensure the caDSR is left in the same state as prior to the test run. It is strongly recommended that a dedicated test account be used. Should any of the tests fail it may be necessary to find the test data and manually remove it.

2. valid.pswd
   A password to match the user id.

3. invalid.userid
   A user id which does not have curate authority or which does not exist in the caDSR database. This can be a non-existent account or an account with read-only privileges to the caDSR.

4. invalid.pswd
   A password which does not exist and is known to be invalid.

4 Administration Notes

Following is some important information concerning the administration of the caDSR Sentinel Tool and the generated output. Multiple references are made to the previous sections.

Alert Reports Creation

Alert Reports are created by the automated run and manually via the Sentinel Tool user interface Run command. A manual submission causes a new execution thread to be started by the Servlet to ensure the user's browser is not locked up waiting for the Alert report creation.

Every manual run writes internal messages to the Sentinel log as defined in the JBoss log4j.xml and generates two (2) files. One is the formatted output (in HTML form with a file extension of html) containing the caDSR information matching the Alert Definition which is sent to the Alert Definition recipients via email. The other is a secondary log file (in HTML form with a file extension of html) containing messages from the run sent to the Alert Administrator via email.

The automated process will create multiple report files in HTML, write internal messages to the Sentinel log as defined in the Auto Run log4j.xml and create a secondary log file which is sent to the Sentinel Administrator via email.

Alert Report Distribution

The email sent to the Alert recipients contains links, not attachments, to the HTML file. Each recipient is sent a separate email consequently the 'CC' and 'BCC' fields are blank. To avoid potential spamming some SMTP servers are set with a limit of the number of emails that can be sent per connection. Consequently, to avoid a problem with this limit, the software establishes one (1) connection, sends one (1) email and closes the connection. If any problems occur, please contact the appropriate email support or administrator.

Additionally the configured ADMIN.* accounts receive an email with a link to the log file. The subject of this email is '<EMAIL.SUBJECT> LOG' provided there are no errors. Should errors occur the word 'ERRORS' is suffixed to the subject. This was done to facilitate the review of the emails and the creation of rules within an email reader.

Disk Space Use

As there is no way for the software to currently know when a user is finished with a report, every file is generated with a unique name using a combination of the type of submission and a time stamp. This could cause a substantial number of files to accumulate in the folder identified by OUTPUT.DIR. To manage this, please provide a server with a large amount of free space. A scheduled task which either moves or deletes files that are too old is recommended. An example UNIX command to delete files 30 days old is:

```
find ... -mtime 30 –exec rm {} \;
```

Remember to place the folder name in the command in place of the '…' At NCI this command is part of the Auto Run script although it is not part of the example script in Section 2, above. It was also necessary to generate unique names for the files as the email only contains links to the report and not the report content. This way the user can still retrieve historical reports up to the time they become old.

Links vs. File Attachments

The use of links was chosen after careful consideration. During development of the Sentinel Tool the options considered included:

1. Attach the report file to the email.
2. Embed the report in the body of the email.
3. Zip all the reports and attach to the email.
4. Place links to the report files in the email.

Each of these approaches has advantages and disadvantages. Considerations for each approach included:

1. Potential size of a single report.
2. Potential number of recipients for any one report.
3. Total number of reports created during a single automated run.
4. Email servers stripping off attachments for possible virus protection.
5. Management of disk space.
6. Access to the reports.
7. User forwarding of a report.
8. Retrieving or accessing older reports.

The Sentinel Tool Automated Run does control the number of emails sent by creating only one per recipient. For example, if one Alert is to go to 5 recipients and a second goes to 3 of the same recipients, those 3 will receive a single email which contains two (2) links, one for each report. Further the logic organizes the output by email address to avoid problems with one person having multiple accounts with distinct names associated to the same email address.

Process Monitoring

While the report creation is running, progress can be determined by finding the most recent log file in the OUTPUT.DIR and using the UNIX 'tail' command. For Windows environments a freeware version can be obtained from Cygwin (http://www.cygwin.com/). For those not familiar with this command it displays the lines at the end of a file. The Run process (both automated and manual) flush the log file buffer with each write to facilitate tracking progress and to ensure the most up to date information is in the file should an unforeseen error occur.

5 Required Technology

The product versions which follow were used during the development and system testing of the caDSR Sentinel Tool. Although new versions of the products have been released they have not yet been certified with the Sentinel Tool version 3.0.1.

Note: deviating in any way from the products and versions listed below may cause errors. When attempting to deploy the Sentinel Tool to a different technology stack, please first deploy as documented in this Installation Guide and then begin the migration.

Product	Version	Web Site & Notes
JBoss	4.0.2	http://www.jboss.org/downloads/index
JDK	1.5.0_06	http://java.sun.com/j2se/1.5.0/
		Recommend downloading the J2EE 1.5 SDK.
Struts	1.2.2	http://struts.apache.org/
Log4j	1.2.13	http://logging.apache.org/log4j
ANT	1.6.2	http://ant.apache.org/
Oracle	9i	http://www.oracle.com
		Purchase from Oracle.
Eclipse	3.0.1	http://www.eclipse.org/platform/
		This is a development environment only and provides no runtime or build components to the Sentinel Tool.
IE	6.0+	http://www.microsoft.com/windows/ie/downloads
		This is the latest Internet Explorer from Microsoft.

6 Troubleshooting

Following is a list of known errors and resolutions that may occur during the build and deployment process. If you have a question or situation not covered in this document please contact the NCICB Help Desk via email at ncicb@pop.nci.nih.gov.

1. **java.lang.NoClassDefFoundError: oracle/jdbc/driver/OracleDriver in JBoss server.log.**
   The message indicates the oracle ojdbc14.jar is missing or not in the class path. This file is included in the open source package. If the problem persists, find the JAR in the Oracle installation folders (%oracle_home%) and copy it to the appropriate JBoss lib folder or modify the class path to include the file. If Oracle is not installed on the machine contact your database or system administrator to acquire the file.

2. **The Sentinel Tool Login screen appears and a valid user name and password is not accepted.**
   Verify the caDSR database references above, i.e. oracle-ds.xml, cadsrsentinel.xml and DSurl in DSRAlertDeploy.properties.

   Open a command prompt and enter the command "tnsping <database server>", without quotes and substituting the database server with the server reference from oracle-ds.xml.

   Verify the Oracle Application Server is **not** running.

3. **Using Firefox, Mozilla and other web browsers**
   The Sentinel Tool has only been certified with Microsoft Internet Explorer version 6.0 and higher.

If you have a question or situation not covered in this document please contact the NCICB Help Desk via email at ncicb@pop.nci.nih.gov.