

calIntegrator Analysis Server Guide

Analysis-Server Installation Guide

This page provides an introduction to Analysis Server, with installation and troubleshooting information.

Contents

- [Introduction](#)
- [Requirements](#)
- [Setting the Environment Variables:](#)
- [Connection to CVS Repository](#)
- [Downloading and Deploying the Codebase](#)
- [Additional post-deployment test case instructions](#)
- [Steps to add a Data File](#)

Introduction




The Analysis Server performs on-the-fly statistical calculations for the calIntegrator application. It can be hosted on the same machine that runs the any application (ex. Rembrandt or ISPY) or it can be run on any number of remote compute machines. In the remote configuration, each compute machine communicates with the appropriate application via the JBossMQ Java Messaging Service (JMS) provided by JBoss. It is important to note that each compute machine does not require its own JBoss instance. The Analysis Server relies on the JMS queues defined in the JBoss instance running the calIntegrator application, such as Rembrandt.


Analysis-commons is a module which defines classes for communicating with the analysis server. calIntegrator Applications like Rembrandt , I-SPY etc use the request and response classes defined in the analysis commons module for communication with the analysis server.


Requirements

The following assumptions are in place:


You have Java and Ant installed, as listed below

Required Software Name Version URL to Download	Description
JDK 1.5.X (currently using 1.5.14) http://java.sun.com/javase/downloads/index_jdk5.jsp 	The J2SE Development Kit (JDK) supports creating J2SE applications.
JBoss Application Server (4.0.X) http://labs.jboss.com/jbossas/downloads/ 	JBoss Application server is the Java application Server
Apache Ant, 1.7.0 http://ant.apache.org/bindownload.cgi 	Apache Ant is a Java-based build tool.

R and Rserve have been installed. Please see the Rserve homepage at:
<http://stats.math.uni-augsburg.de/Rserve/> 

And the Rserve installation instructions:
<http://stats.math.uni-augsburg.de/Rserve/doc.shtml> 

It is recommended that you read the Rserve installation instructions before installing R. Specifically, the Rserve instructions recommend building the R distribution from source using the (color:red)**--enable-R-shlib flag**(color).

The R source distribution can be obtained from R project site:
<http://www.r-project.org/>  (click the link under download)

Setting the Environment Variables:



The purpose of setting operating system environment variables is so that the Java SDK and Ant build tool are available to run from anywhere in the system.

Linux

To set the environment variables in Linux, follow these steps:

Step	Action
1	As the root user, enter the following in the /etc/profile file. A PATH variable should already be created in this file, so be sure to define the JAVA_HOME and ANT_HOME export before the PATH export. Replace <installation_directory> with the directory where you have installed the Java SDK and Ant. (The location example in the Apache Ant installation (p. 5) is /usr/java.) export JAVA_HOME={ }<installation_directory>(/jdk1.5.0_10 export ANT_HOME=()<installation_directory>(/apache-ant-1.7.0 export PATH=\$JAVA_HOME/bin:\$ANT_HOME/bin:\$PATH
2	Log out and log back in so that the system recognizes your changes.

Verifying the Environment Variables in Linux

To verify that environment variables have been set correctly, follow these steps:

Step	Action
1	From the command line, enter: echo \$JAVA_HOME echo \$ANT_HOME Both of these commands should return the location where you installed these tools.
2	To verify your Java SDK installation, enter java -version from a command prompt. You should see java version "1.5.0_10" .

Connection to CVS Repository

- 1) Connect to CVS repository
- 2) Here are the parameters
 - a) Host: cbio cvs2.nci.nih.gov
 - b) Repository path: /share/content/cvsroot
 - c) User: anonymous
 - d) Password: anonymous
 - e) Connection type: pserver

Downloading and Deploying the Codebase

Step 1

Check out **cainTEGRATOR-analysis commons** from HEAD or provided CVS tag. Build it using ant script. (Please follow the steps)

- a) Open a COMMAND prompt. cd into cainTEGRATOR-analysis commons folder.
- b) then type **ant build_dependency** . It will start the build and get the message **BUILD SUCCESSFUL**.
- c) Do a CVS check out of the **cainTEGRATOR-analysis-server**

Step:2

Download the latest client jar file (use the exceptions version) from:

<http://stats.math.uni-augsburg.de/Rserve/down.shtml>  after agreeing to their licensing terms and conditions.

Copy the downloaded file into the lib directory under the analysis server install directory and rename it to Rserve.jar .

Step :3

- 1) Build it using ant script from command prompt.
- a) cd into cainTEGRATOR-analysis commons folder.
- b) then type **ant build_tar** . It will start the build and you will get the message **BUILD SUCCESSFUL**.

Step 4

Setup JMS Configuration Files:

- 1) Copy the file "**cainTEGRATOR-jbossmq-destinations-service.xml**" from \artifacts to the directory where you installed JBoss.
/jboss-4.0.4.GA/server/default/deploy/jms folder
- 2) Copy the application specific file for example "**rembrandt-jbossmq-destinations-service.xml**" (for REMBRANDT) from \artifacts to the directory where you installed JBoss.
/jboss-4.0.4.GA/server/default/deploy/jms folder

Step 5

Deploy the Analysis Server.



The purpose of setting operating system environment variables is so that the Java SDK and Ant build tool are available to run from anywhere in the system.

Step 5.1

Copy the caintegrator-analysis-server.tar file created to the directory `/local/content/caintegrator`.

Step 5.2

Type the command: **`tar -xvf caintegrator-analysis-server.tar`**

Step 6

Go to the newly created directory `/local/content/caintegrator/bin`. Type the commands:

`chmod u+x *.sh`
`dos2unix *.sh`

Step 7

Start Rserve by typing:

`./startRserve.sh`

Rserve should now be running. The last line printed on the console should be:

"Rserv started in daemon mode."

Step 8

Go to the directory `/home/caintegrator` (you can do this by just typing `cd`).

Start the analysis server by executing the command:

`./S92Rserver start`

Additional post-deployment test case instructions

1) Open the file `/local/content/caintegrator/log/analysisServer.log` and you should see something similar to:

a) [translationalResearch:main] INFO gov.nih.nci.caintegrator.analysis.server.RThread - Done initializing Rserver connection elapsedTime=5100
b) [translationalResearch:main] INFO gov.nih.nci.caintegrator.analysis.server.RThread - RThread name=Thread-0 successfully initialized R connection.
5046

2) There may be a warning:

a) [translationalResearch:main] WARN gov.nih.nci.caintegrator.analysis.server.AnalysisServer - could not establish connection with...

But this is of no concern. It is because there will be no JMS MQs available until JMS message Queues have been established, running and listed in the JNDI lookup found at the location specified in the property:

3) After you deploy the application you should return to this log file and see the following:

a) [translationalResearch:main] INFO gov.nih.nci.caintegrator.analysis.server.AnalysisServer - Now listening for requests...

Troubleshooting :

- "could not establish connection with provider.." when starting the Analysis Server.

This error can occur when the Java Messaging Service (JMS) provider is not running or has not fully initialized. Make sure that the JMS provider that is configured in the analysis server build file is running. If it is running but has not fully initialized the analysis server will automatically connect when it is fully initialized.

- Analysis Server appears to be running but no results are returned.

Using the `jmx-console` (see the JBoss documentation) application go to the `jboss.mq.destination` section and inspect the `AnalysisResponseQueue`. Make sure that the receivers count equals 1. If it is greater than 1 then there is another listener on the queue which is stealing the results from application. Next inspect the `AnalysisRequestQueue`. Make sure that the receivers count is equal to the number of analysis server machines. For example if you have configured four analysis server machines then the receivers count should equal four.

If the receiver counts are not as expected then the problem may stem from network/firewall issues or from a conflicting instance of the application.

- If you are using a version of JBoss other than 4.0.2, you should replace the following jar files:

concurrent.jar
jbossmq-client.jar
jboss-common-client.jar
jboss-system-client.jar
jnp-client.jar

in the lib directory under the analysis server install directory, with the corresponding jar files from the JBoss distribution.

Steps to add a Data File

Step 1

The [calIntegrator-ODI GForge project files page](#) has data files for particular application

For example:

Rembrandt:

Rembrandt_mas5.dataMatrix_275_10-15-07.Rda

Rembrandt_UnifiedGene.dataMatrix_275_10-15-07.Rda

Ispy:

newDataFile.zip

Step 2

Copy the data file to the directory local/content/caintegrator/Rdatafiles/

Step 3

Restart the analysis server

Log in to the analysis machine

Stop the analysis server by executing the command:

`./S92Rserver stop`

Start the analysis server by executing the command:

`./S92Rserver start`

Using the Analysis Server for Development Purposes:

On calIntegrator project, for development, the developers share a common analysis server and have a dedicate JMS node for their local Jboss.

This allows us to independently develop and test queries against the analysis server.

You can setup something similar for development using our files

1) Copy StartStopNodes_develop to /home/calIntegrator and started and stopped similar to S92Rserver

`./StartStopNodes_develop stop`

And

`./StartStopNodes_develop start`

2) Copy runDeveloperNodes.sh to /local/content/caintegrator/bin/

And modify it by adding a new line for each developer using this file as an example

3) Copy and rename analysisServerDeveloper1.properties to local/content/caintegrator/bin/

And modify the ip address of the developer in this file plus the file name should match the setting in the runDeveloperNodes.sh file

Next use localhost:1099 on your server to connect to the analysis server. You do not need to start and stop the analysis server for each developer unless your going to upload a new R-binary file.

You can substitute actual developer name to identify each node.