

caIntegrator Bioconductor Installation Guide



The information and links on this page are no longer being updated and are provided for reference purposes only.



To Print the Guide

You can create a PDF of the guide. For instructions refer to the tip [Printing multiple pages](#). If you want to print a single page, refer to [Printing a page](#).

This document provides instructions for installing caIntegrator Bioconductor.

Topics in this document include:

- [Overview](#)
- [Prerequisites](#)
- [Install Dependent Packages](#)
 - [Globus 4.0.3](#)
 - [Apache Axis 1.4](#)
 - [ActiveMQ 4.0.2](#)
 - [R 2.9.0](#)
 - [Dependent R packages \(RCurl, SJava, RWebServices, RUnit, DNACopy\)](#)
- [Install caDNACopy](#)
 - [caDNACopy Base Code](#)
 - [caDNACopy RWebService](#)
 - [caDNACopy Grid Service](#)
- [Install caCGHcall](#)
 - [caCGHcall Base Code](#)
 - [caCGHcall RWebService](#)
 - [caCGHcall Grid Service](#)
- [Launch Bioconductor Services](#)
 - [Starting Bioconductor Services](#)
- [Optional Improvements](#)
 - [Scripts to Startup/Shutdown Bioconductor](#)
 - [Setting Up Common Logging](#)



Entering Commands

Text in this document formatted as monospace bold type indicates Unix or R commands you should type, as indicated, at the command line.

Overview

caIntegrator provides the capability to perform segmentation and copy number analysis using either caDNACopy or caCGHcall. Both of these tools are part of the Bioconductor suite and are implemented as caGrid services wrapping the R Bioconductor code. You can install one or both of these tools either on a separate server from caIntegrator or on the same server.



Warning

The R processing associated with caDNACopy and/or caCGHcall can be very CPU and memory intensive.

The current installation instructions only support installation on Linux platforms. Recommended configurations are given at the bottom.

Prerequisites

The following software packages must be installed prior to beginning the instructions:

- [Java Development Kit \(JDK\) 1.6.x](#)
 - Ensure that JAVA_HOME is defined, preferably in the user or system profile
 - Ensure that \$JAVA_HOME/bin is in PATH
- [Ant 1.7.x](#)
 - Ensure that ANT_HOME is defined, preferably in the user or system .profile
 - Ensure that \$ANT_HOME/bin is in PATH
- [Tomcat 5.5.x](#)
 - Instructions assume Tomcat is configured to run on port 8080. If you are using another port, substitute it into later instructions
 - Ensure that CATALINA_HOME is set
 - Tomcat may be autostarted or started as part of the Bioconductor launch script (provided later in these instructions)

Install Dependent Packages

The caIntegrator Bioconductor installation also depends on several other packages that may not be installed on your system. Summaries of these packages and directions for installing them follow.

Globus 4.0.3

Installs at /usr/local/ws-core-4.0.3

1. [Download](#)
2. Unpack tar file into /usr/local/ws-core.4.0.3
3. Ensure that GLOBUS_LOCATION is set to /usr/local/ws-core-4.0.3

Apache Axis 1.4

Installs files into \$CATALINA_HOME

1. [Download](#)
2. Unpack tar file in home directory.
3. Move axis-1_4/webapps/axis directory to \$CATALINA_HOME/webapps.
4. Stop and restart Tomcat (\$CATALINA_HOME/bin/shutdown.sh; \$CATALINA_HOME/bin/startup.sh).
5. Test your Axis installation by visiting <http://YOUR-HOSTNAME:8080/axis/happyaxis.jsp> (be sure to substitute the name of your machine in the URL).

ActiveMQ 4.0.2

Installs at /usr/local/incubator-activemq-4.0.2

1. [Download](#)
2. Unpack tar file into /usr/local/incubator-activemq-4.0.2
3. Ensure that JMS_HOME is set to /usr/local/incubator-activemq-4.0.2
4. Edit file \$JMS_HOME/conf/activemq.xml
 - a. Change line that reads "<broker useJmx="true">" to "<broker useJmx="true" persistent="false">"
 - b. Replace line that reads "<transportConnector name="default" uri="tcp://localhost:61616" discoveryUri="multicast://default"/>" with "<transportConnector name="default" uri="tcp://localhost:61616"/>".
 - c. Comment out or remove line that reads "<networkConnector name="default" uri="multicast://default"/>".
5. Make main binary executable.
 - a. `chmod 755 $JMS_HOME/bin/activemq`

ActiveMQ can be autostarted or started as part of the Bioconductor launch script (provided later in these instructions)

R 2.9.0

Installs at /usr/local/R-2.9.0

1. [Download](#)
2. Unpack tar file in home directory
3. Configure, build, and install
 - a. `cd R-2.9.0`
 - b. `./configure --enable-R-shlib --with-readline=no --with-x=no`
 - c. Fix src/modules/Makefile as follows (fix is from <http://tolstoy.newcastle.edu.au/R/e6/devel/09/04/1434.html>)
 - Change lines that read:

```
for d in "$(R_MODULES)"; do \  
    (cd $$d && $(MAKE) $@) || exit 1; \  
done
```

- To read:

```
@if test "$(R_MODULES)" != ""; then \  
    for d in "$(R_MODULES)"; do \  
        (cd $$d && $(MAKE) $@) || exit 1; \  
    done; \  
fi
```



Warning

Be very careful of tabs in the lines, one tab to start each line, indent beyond that with spaces. Be careful to have no spaces at the end of the lines too. Use rest of Makefile as an example.

- d. `make`
- e. `make install prefix=/usr/local/R-2.9.0`

Dependent R packages (RCurl, SJava, RWebServices, RUnit, DNACopy)

Installs files into \$R_HOME

1. `$R_HOME/bin/R`
2. At the R prompts:
 - a. `source("http://www.bioconductor.org/biocLite.R")`
 - b. `biocLite()`
 - c. `biocLite("RCurl")`
 - d. `biocLite("SJava")`
 - e. `biocLite("RWebServices")`
 - f. `biocLite("RUnit")`
 - g. `biocLite("DNACopy")`
 - h. `biocLite("CGHcall")`
 - i. `q()`
3. Build SJava links missed by install:
 - a. `cd $R_HOME/lib64/R/library/SJava/libs`
 - b. `ln -s SJava.so libRInterpreter.so`
 - c. `ln -s SJava.so libSJava.so`
4. Build and test SJava/RWebServices installation, following these steps.
 - a. `$R_HOME/bin/R`
 - b. At the R prompts:
 - i. `library(RWebServices)`
 - ii. `unpackAntScript("/tmp/rservices")`
 - iii. `q()`
 - c. `cd /tmp/rservices`
 - d. `ant recompile-sjava`
 - e. `ant basic-prop`
 - Look for any errors.
 - f. `ant rservices-test`
 - This step is optional.

Install caDNACopy

This section installs the caDNACopy grid in three pieces: base code, R web service, and caGrid service. You may skip this section if you are only using caCGHcall.

caDNACopy Base Code

Installs into R_HOME

1. `cd ~`
 - This is your home directory
2. `svn checkout https://ncisvn.nci.nih.gov/svn/bioconductor/branches/calIntegrator`
3. `cd caIntegrator/services/caDNACopy/R`
4. `$R_HOME/bin/R CMD INSTALL caDNACopy`

caDNACopy RWebService

Installs into /usr/local/bioconductor/caDNACopy

1. `cd /usr/local/bioconductor`
2. `$R_HOME/bin/R`

At the R prompts:

 - a. `library(RWebServices)`
 - b. `unpackAntScript("caDNACopy")`
 - c. `q()`
3. Optionally, fix the timeout value for all R workers to be 2 hours instead of 60 seconds.
 - a. Edit `$R_HOME/lib64/R/library/RWebServices/scripts/RWebServicesTuning.properties`. Change to: `jms.timeout=7200000` .
4. `cd caDNACopy`
5. `ant map-package -Dpkg=caDNACopy`
6. `ant unpack-package -Dpkg=caDNACopy`
7. `ant precompile`
8. Optionally, test RWebService, following these steps.

- a. Start up Tomcat (If Tomcat is already running, you must shut it down before using the following command): `$CATALINA_HOME/bin/startup.sh`
- b. Start up ActiveMQ: `$JMS_HOME/bin/activemq &`
- c. Start up R worker for caDNAcopy: `nohup ant start-worker &`
- d. Run the test: `ant local-test`

Check any failures in test/output. Ignore failures that say "expected 79, got 80"

caDNAcopy Grid Service

Installs into `$CATALINA_HOME`

1. Download [Bioconductor-caGrid-Services.tar.gz](#).
2. Unpack tar file in home directory.
3. Build and deploy caDNAcopy grid service.
 - a. `cd caGrid/CaDNAcopy`
 - b. `ant -Dno.deployment.validation=true deployTomcat`
4. Prepare Tomcat for Grid applications.
 - a. `cd $GLOBUS_LOCATION`
 - b. `ant -f share/globus_wsrf_common/tomcat/tomcat.xml deployTomcat -Dtomcat.dir="$CATALINA_HOME"`
 - c. `cp $JMS_HOME/lib/*.jar $CATALINA_HOME/webapps/wsrif/WEB-INF/lib`
5. Optionally, fix the timeout value in `caDNAcopy.jar` to be 2 hours instead of 60 seconds.
 - a. `mkdir ~/unjar; cd ~/unjar`
 - b. `unzip $CATALINA_HOME/webapps/wsrif/WEB-INF/lib/caDNAcopy.jar`
 - c. Edit `org/bioconductor/rseviceJms/services/caDNAcopy/RWebServices4java.properties`. Change to: `jms.timeout=720000`
 - d. `zip -r caDNAcopy.jar *`
 - e. `cd $CATALINA_HOME/webapps/wsrif/WEB-INF/lib`
 - f. `cp ~/unjar/caDNAcopy.jar .`

Install caCGHcall

This section installs the caCGHcall grid service in three pieces: base code, R web service, and caGrid service. You can skip this section if you are only using caDNAcopy.

caCGHcall Base Code

Installs into `R_HOME`

1. `cd ~`
 - This is your home directory
2. `svn checkout [https://ncisvn.nci.nih.gov/svn/bioconductor/branches/caIntegrator]`
3. `cd caIntegrator/services/caCGHcall/R`
4. `$R_HOME/bin/R CMD INSTALL caCGHcall`

caCGHcall RWebService

Installs into `/usr/local/bioconductor/caCGHcall`

1. `cd /usr/local/bioconductor`
2. `$R_HOME/bin/R`
3. At the R prompts:
 - a. `library(RWebServices)`
 - b. `unpackAntScript("caCGHcall")`
 - c. `q()`
4. Optionally, fix timeout value for all R workers to be 2 hours instead of 60 seconds.
 - a. Edit `$R_HOME/lib64/R/library/RWebServices/scripts/RWebServicesTuning.properties`. Change to: `jms.timeout=720000` .
5. `cd caCGHcall`
6. Change the queue name for caCGHcall so it does not conflict with caDNAcopy, by editing `RWebServicesTuning.properties`. Change to: `jms.queue=CGHC` .
7. `ant map-package -Dpkg=caCGHcall`
8. `ant unpack-package -Dpkg=caCGHcall`
9. `ant precompile`
10. Optionally, test the RWebService, following these steps.
 - a. Start up Tomcat (If Tomcat is already running, you must shut it down before using the following command): `$CATALINA_HOME/bin/startup.sh`
 - b. Start up ActiveMQ: `$JMS_HOME/bin/activemq &`
 - c. Start up R worker for caCGHcall `nohup ant start-worker &`
 - d. Run the test: `ant local-test`

Check any failures in test/output. Ignore failures that say "expected 79, got 80"

caCGHcall Grid Service

Installs into `$CATALINA_HOME`

1. Build and deploy caCGHcall grid service.
 - a. `cd ~/caIntegrator/services/caCGHcall/caGrid/caCGHcall`
 - b. `ant -Dno.deployment.validation=true deployTomcat`
2. Prepare Tomcat for Grid applications.
 - a. `cd $GLOBUS_LOCATION`
 - b. `ant -f share/globus_wsrf_common/tomcat/tomcat.xml deployTomcat -Dtomcat.dir="$CATALINA_HOME"`
 - c. `cp $JMS_HOME/lib/*.jar to $CATALINA_HOME/webapps/wsrf/WEB-INF/lib`

Launch Bioconductor Services

This section describes how to start up all the pieces of the Bioconductor installation once they have been successfully installed. In [#Optional Improvements](#), you can find optional scripts that can automate the startup and shutdown of all processes.

Starting Bioconductor Services

See the [Bioconductor Installation Guide page 11](#).

1. Start up Tomcat (shutdown if already running)
 - a. `$CATALINA_HOME/bin/startup.sh`
2. Start up ActiveMQ
 - a. `$JMS_HOME/bin/activemq &`
3. Start up R worker for caDNAcopy (skip if only using caDNAcopy)
 - a. `cd /usr/local/bioconductor/caDNAcopy`
 - b. `nohup ant start-worker &`
4. Start up R worker for caCGHcall (skip if only using caCGHcall)
 - a. `cd /usr/local/bioconductor/caCGHcall`
 - b. `nohup ant start-worker &`

Optional Improvements

Scripts to Startup/Shutdown Bioconductor

These shell scripts can be used to automate the launch of Bioconductor. If these are used to automatically start the Bioconductor services at system startup, ensure that all environment variables (as listed in sections above) are already defined. All three scripts should be placed in the same directory and must be set to be executable (`chmod a+x`). Also note that these scripts are intended to be used with the common logging setup that is described in [#Setting Up Common Logging](#).

1. start-worker.sh

```
#!/bin/bash
#

WORKERTYPE=$1
WORKERID=$2

# Run worker
echo "Starting $WORKERTYPE worker ($WORKERID)..."
cd /usr/local/bioconductor/$WORKERTYPE
ant start-worker | sed -u "s/[java]/[$WORKERTYPE $WORKERID]/"
```

2. start-bio.sh

```
#!/bin/bash
#
LOGFILE=/usr/local/bioconductor/logs/bioconductor-combined.log

THISDIR=`dirname $0`

# Run Tomcat
echo Starting Tomcat...
$CATALINA_HOME/bin/startup.sh

# Run ActiveMQ
echo Starting ActiveMQ...
$JMS_HOME/bin/activemq start &
sleep 10

# Run caDNACopy workers
nohup $THISDIR/start-worker.sh caDNACopy 1 >> $LOGFILE &
nohup $THISDIR/start-worker.sh caDNACopy 2 >> $LOGFILE &

# Run caCGHcall workers
nohup $THISDIR/start-worker.sh caCGHcall 1 >> $LOGFILE &
nohup $THISDIR/start-worker.sh caCGHcall 2 >> $LOGFILE &
```

3. stop-bio.sh

```
#!/bin/bash
#
# Stop ActiveMQ
$JMS_HOME/bin/activemq stop &

# Stop Tomcat
$CATALINA_HOME/bin/shutdown.sh

# Stop caDNACopy/caCGHcall workers
### They stop themselves when ActiveMQ shuts down
```

Setting Up Common Logging

One disadvantage to having multiple processes for Bioconductor is that each process creates a separate log file by default. By using the following steps, the log files for Tomcat, ActiveMQ, and caDNACopy/caCGHcall are placed into a single directory, `/usr/local/bioconductor/log`. If the scripts in [#Optional Improvements](#) are used, it further improves logging by placing all messages into a single log file, `/usr/local/bioconductor/logs/bioconductor-combined.log`.

1. `mkdir /usr/local/bioconductor/logs`
2. Edit `/usr/local/activemq/conf/log4j.properties`
 - a. Change `"log4j.rootLogger=INFO, stdout"` to `"log4j.rootLogger=INFO, out"`
 - b. Change `"log4j.appender.out.file=${activemq.home}/data/activemq.log"` to `"log4j.appender.out.file=/usr/local/bioconductor/logs/activemq.log"`
3. `cd /usr/local/bioconductor/logs`
4. `ln -s /usr/local/bioconductor/caDNACopy/nohup.out ./rworker-dnacopy.log`
5. `ln -s /usr/local/bioconductor/caCGHcall/nohup.out ./rworker-cghcall.log`
6. `ln -s $CATALINA_HOME/logs/catalina.out ./tomcat.log`



Help Downloading Files

For help accessing PDF, audio, video, and compressed files on this wiki, go to [Help Downloading Files](#).