

caArray 1.6 MAGE-OM API

Installation Guide



NATIONAL[®]
CANCER
INSTITUTE

Center for Bioinformatics

Revised September 6, 2007

Table of Contents

| | |
|---|-----------|
| INTRODUCTION | 1 |
| MINIMAL SYSTEM REQUIREMENTS..... | 2 |
| Configuring the System Environment | 3 |
| MAGE-OM PRECONFIGURED CLIENT INSTALLATION | 3 |
| MAGE-OM API SERVER AND CLIENT INSTALLATION | 4 |
| MAGE-OM API Server Dependency | 4 |
| Downloading and Unzipping MAGE-OM Files..... | 5 |
| Setting up the System Environment..... | 6 |
| Deploying MAGE-OM | 6 |
| Starting and Stopping MAGE-OM..... | 11 |
| Testing MAGE-OM..... | 12 |
| TROUBLESHOOTING | 13 |
| APPENDIX A EXAMPLE BUILD.PROPERTIES FILE..... | 15 |
| APPENDIX B XML PARSER ISSUE 1.01..... | 19 |
| CONTACTING APPLICATION SUPPORT | 20 |

Introduction

Microarray Gene Expression Object Model (MAGE-OM) is a data exchange model for microarray experiments which has been modeled using the Unified Modeling Language (UML). The *caArray 1.6 MAGE-OM API Installation Guide* is intended to help system administrators to install the MAGE-OM Application Programming Interface (API). This guide describes how to install the three scenarios described in the [Overview of caArray MAGE-OM API Installation](#) on page 2.

Overview of caArray

The caArray (<http://caarray.nci.nih.gov/>) software has been developed by the NCI Center for Bioinformatics (NCICB) to create an information-sharing network modeled on the World Wide Web. caArray consists of a microarray database and microarray data visualization and analysis tools. caArray is an open source project, and the source code and APIs are available in the download site at the NCICB web site, <http://ncicb.nci.nih.gov/download/index.jsp>. caArray is designed to make microarray data publicly available, and to develop and bring together open source tools to analyze these data.

caArray MAGE-OM API

The caArray MAGE-OM API is a set of Java objects that adhere to the object model defined by [OMG's Gene Expression v1.1](#). The caArray MAGE-OM API objects provide access to data in the caArray database via Remote Method Invocation (RMI) call issued to a dedicated MAGE server at NCI or any other site with an accessible MAGE-OM server installation.

There are two primary types of objects defined in the API:

1. MAGE-OM-compliant interfaces
2. Custom MAGE-OM Impl (implementation) objects

The MAGE-compliant objects are defined as Java interfaces, which the custom MAGE-OM Impl Java classes implement. This ensures the custom MAGE-OM Impl provide a MAGE-OM compliant API.

The MGED Society website is an excellent source for supplemental material on the MAGE object model - <http://www.mged.org/>.

NOTE:



Existing caArray development documentation can be found on the caArray page of the NCICB web site: <http://caarray.nci.nih.gov/documentation>.

| | |
|---|---|
| Overview of caArray MAGE-OM API Installation | <p>This installation guide outlines three use cases or scenarios for installing MAGE-OM API:</p> <ol style="list-style-type: none"> 1. <i>MAGE-OM Preconfigured Client Installation</i> on page 3 - This use case describes a setup where the user is using the NCICB-provided client jar file to query data in the <i>NCICB installation of caArray</i>. This client jar has the necessary URL and port information included in the jar and does not need any configuration changes to talk to the NCICB server. 2. <i>MAGE-OM API Server and Client Installation</i> on page 4 - This use case describes the scenario for building and running a <i>local instance of caArray Mage-OM client as well as server</i>. It includes the steps needed to build Mage-OM server and clients for deployment and access data from a local deployment of caArray at a non-NCICB center. <p>Before installing the MAGE-OM API, make sure your system has the minimum requirements as specified in the next section, Minimal System Requirements,</p> |
|---|---|

Minimal System Requirements

Minimal System Requirements The hardware environment that has been tested and verified by NCICB includes the following:

- **Processor:** 1GHZ Dual (SPARC)
- **Memory:** 10 GB
- **Hard Drives:** 73GB (Mirrored)

The amount of memory required depends on the amount of data stored in your database (see [Troubleshooting](#) on page 13). [Appendix B](#) and [Appendix D](#) report information from the user community for installing MAGE-OM in different environment environments.

Configuring the System Environment

Complete the following steps to configure your system environment for all scenarios:

| Step | Action |
|------|---|
| 1 | Download and install Java 2 SDK version 1.5.0_06+ (http://java.sun.com/javase/downloads/index.jsp) and set the JAVA_HOME environment variable to point to the SDKs installation directory. You should also put {JAVA_HOME}\bin directory on the front of your PATH variable. |
| 2 | Download and install Apache Ant version 1.6.5+ (http://ant.apache.org/) and set the ANT_HOME environment variable to point to the Ant installation directory. You should also put {ANT_HOME}\bin directory on the front of your PATH variable. |
| 3 | You must have an account from caArray including a user name and password. If you do not, go to http://caarraydb.nci.nih.gov/caarray/index.jsp to register. The user name and password are used to get the security ID. |
| 4 | Make sure your <code>java.policy</code> file is set up correctly. See <i>Troubleshooting</i> (number 1) on page 13 for more information on the <code>java.policy</code> file. |


MAGE-OM Preconfigured Client Installation

This section describes the steps necessary to download and install a preconfigured client Jar file to talk to the NCICB caArray server.

Complete these steps to download and install a preconfigured client Jar file:


| Step | Action |
|------|---|
| 1 | Go to the NCICB download web site http://ncicb.nci.nih.gov/download/index.jsp |
| 2 | Provide your email, name, and institution. Click Enter the Download Center . |
| 3 | Select caArray , agree to the caArray software license by selecting Checking this box indicates that you agree to the above terms , and click Download . |
| 4 | Select the <code>caarray-mageom-client.{version}.zip</code> file and save it to your computer. |
| 5 | Unzip the contents. |

| Step | Action |
|------|--|
| 6 | Make sure the the <code>mageom-client.jar</code> and all dependent jar files (under <code>/lib</code>) are in the classpath of the application/test cases which would be making the call to server. |

| | |
|---|---|
|  | <p>NOTE:</p> <p>If you can not access the NCICB server, see Troubleshooting on page 13 for help. If you'd like examples of how to use the API, see both the caArray 1.6 Technical Guide (http://ncicb.nci.nih.gov/download/downloadcaarray.jsp) and Downloading and Unzipping MAGE-OM Files on page 5 of this guide.</p> |
|---|---|

MAGE-OM API Server and Client Installation

This section describes the steps necessary to download and install the MAGE-OM source file which includes a client and server.

| | |
|---|--|
|  <p>BEFORE YOU BEGIN</p> | <p>There must be a running instance of caArray available in order to deploy and run MAGE-OM. The easiest configuration is to have caArray and MAGE-OM on the same machine. If they are not on the same physical machine, then caArray's file share must be visible to MAGE-OM.</p> |
|---|--|

MAGE-OM API Server Dependency

MAGE-OM API relies on caArray `SecurityManagerEJB` to provide authentication and authorization for MAGE-OM client users. `SecurityManagerFactory` is used to locate (JNDI lookup) an instance of `SecurityManager`. `SecurityManagerFactory` uses the file `security-jndi.properties` to populate the `InitialContext` object which does the JNDI lookup. If it isn't found, then default values are used.

`SecurityManager.isUserAuthenticated(String, String)` is called passing in the username and password to verify that the user is authenticated. If authenticated, then a new session ID is created.

Security is implemented as an aspect on the MAGE-OM side which intercepts the call to perform security authentication as well as filtering of objects returned as a result of a search. `CaUserAccess` method in `SecurityManager` is called to filter the objects returned from the search and return only those objects, which a user has access to view.

Downloading and Unzipping MAGE-OM Files

Complete the following steps to download the appropriate MAGE-OM files:

| Step | Action | | | | | | |
|----------------------------|--|-------------------------|-------------------------|-------------------------|----------------------------------|----------------------------|--------------------------------------|
| 1 | Go to the caCORE download web site: http://ncicb.nci.nih.gov/download/index.jsp . | | | | | | |
| 2 | Provide your email, name, and institution. Click Enter the Download Center . | | | | | | |
| 3 | Select caArray , agree to the caArray software license by selecting Checking this box indicates that you agree to the above terms , and click Download . | | | | | | |
| 4 | Select the appropriate MAGE-OM download files save them to your computer. <table border="1"> <thead> <tr> <th><i>File Description</i></th> <th><i>Example Filename</i></th> </tr> </thead> <tbody> <tr> <td>MAGE-OM API Source Code</td> <td>caarray-mageom-src.{version}.zip</td> </tr> <tr> <td>MAGE-OM API Java Documents</td> <td>caarray-mageom-javadoc.{version}.zip</td> </tr> </tbody> </table> | <i>File Description</i> | <i>Example Filename</i> | MAGE-OM API Source Code | caarray-mageom-src.{version}.zip | MAGE-OM API Java Documents | caarray-mageom-javadoc.{version}.zip |
| <i>File Description</i> | <i>Example Filename</i> | | | | | | |
| MAGE-OM API Source Code | caarray-mageom-src.{version}.zip | | | | | | |
| MAGE-OM API Java Documents | caarray-mageom-javadoc.{version}.zip | | | | | | |
| 5 | Create a MAGE-OM root directory (indicated by {MAGE-OM_ROOT}) | | | | | | |
| 6 | Copy caarray-mageom-src.{version}.zip to {MAGE-OM_ROOT}. | | | | | | |
| 7 | Unzip caarray-mageom-src.{version}.zip to {MAGE-OM_ROOT}. Assuming jar is in your path, type: <pre>jar -xvf caarray-mageom-src.{version}.zip</pre> OR <pre>unzip caarray-mageom-src.{version}.zip</pre> | | | | | | |

After unzipping MAGE-OM API, the directory structure should resemble Figure 1.

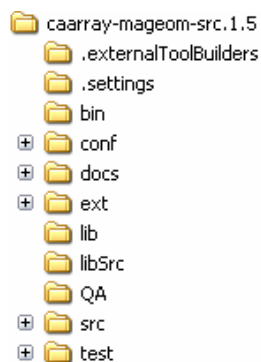



Figure 1 MAGE-OM directory structure

Setting up the System Environment

Complete the following steps to set up your system environment for this installation:

| Step | Action |
|------|---|
| 1 | Copy {MAGE-OM_ROOT}\lib\junit.jar to {ANT_HOME}\lib in order to run the JUnit tests; the MAGE-OM API tests use JUnit (http://junit.sourceforge.net). |
| 2 | Make sure that the Jars in the library are in the classpath |
| 3 | Make sure you have caArray running in JBoss for security login. |
| 4 | If you run a version other than JBoss 4.0.4, you must have the MAGE-OM server use the new jbossall-client.jar from your JBoss distribution. Just overwrite the file in {MAGE-OM_ROOT}\lib with the new one before you build the server. |

Deploying MAGE-OM

| | |
|---|--|
| NOTE:  | There must be a running instance of caArray available to complete the steps to deploy and run the MAGE-OM. |
|---|--|

Complete the following steps to deploy MAGE-OM:

| Step | Action |
|------|--|
| 1 | <p>The <code>build.xml</code> file in the root directory contains all the targets used by this project and it needs a <code>build.properties</code> file to function correctly. By default, the <code>build.xml</code> looks for a <code>local.properties</code> file which will be used to override any of the properties defined within the existing <code>build.properties</code> file in the root directory. Create a <code>local.properties</code> file and verify the settings in the <code>build.properties</code> file are correct for your environment, if they are not put the desired value into the <code>local.properties</code> file. Table 1 contains a description of each parameter and 0 contains an example <code>build.properties</code> file. \</p> <p>Note: You can create various configuration files (<code>.properties</code>) and use the specify <code>ant -Denvironment=<filename-prefix></code> to load different configuration settings for developing the MAGE-OM application.</p> <ul style="list-style-type: none"> When deploying in an external server, the appropriate values for the RMI Properties, OJB Properties and Security Properties in the <code>build.properties</code> file (see Appendix A) need to be modified to point to the correct Server/Datasource. |
| 2 | <p>From {MAGE-OM_ROOT} type the following to build the client and server:</p> <pre>ant</pre> |


| Parameter | Description |
|--|---|
| <code>source.dir</code> | Source directory build property. The default is <code>src</code> . |
| <code>source.java.dir</code> | Source java directory build property. The default is <code>\${source.dir}/java</code> . |
| <code>generated.source.java.dir</code> | Generated Source Java directory build property. The default is <code>\${build.dir}/generated</code> |
| <code>source.resource.dir</code> | Source resource build property. The default is <code>conf</code> . |
| <code>docs.dir</code> | Documentation directory property. The default is <code>doc</code> . |
| <code>javadoc.dir</code> | Java documentation directory property. The default is <code>target/javadoc</code> |
| <code>deployment.docs.dir</code> | Deployment documentation directory property. The default is <code>\${docs.dir}/Deployment</code> |
| <code>build.dir</code> | Build directory property. The default is <code>build</code> . |
| <code>build.lib.dir</code> | Build library property. The default is <code>lib</code> . |
| <code>build.classes.dir</code> | Build classes property. The default is <code>\${build.dir}/classes/</code> . |
| <code>build.resources.dir</code> | Build resources directory property. The default is <code>\${build.dir}/resources/</code> |
| <code>ext.dir</code> | External Resources directory property. The default is <code>ext</code> . |
| <code>build.rmic</code> | RMIC task compiler attribute. The default is <code>sun</code> . |

| <i>Parameter</i> | <i>Description</i> |
|----------------------|--|
| target.dir | Target directory. The default is local. |
| target.platform | Target platform. Set this to the proper system - either windows or unix. |
| target.shell | Target shell. Valid values are sh , bash, tcsh and so forth. |
| client.jar.name | Name of the client jar file. The default is mageom-client. |
| server.jar.name | Name of the server jar file. The default is mageom-server. |
| server.java.home | Path to JAVA_HOME root directory that the server use. The default is \${env.JAVA_HOME}, which is based on running the server on the local machine. |
| mageom.version | MAGE-OM version. The default value is 1.6 |
| rmi.public.host | The host that RMI clients can connect to. It should be the public DNS name that RMI clients can connect to. This property and rmi.public.port make up the default searchLocation URL. |
| rmi.public.port | The port that RMI clients can connect to. This property and rmi.public.host make up the default searchLocation URL. |
| rmi.server.host | The host that the RMI server binds to. It becomes the value of java.rmi.server.hostname. |
| rmi.server.port | The port that the RMI server binds to. It becomes the value of java.rmi.registry.port. |
| rmi.server.data.port | It becomes the value of java.rmi.data.port. If clients are not running outside of a firewall, then set this to -1. |
| rmi.log | Name of the RMI log file. By default, rmi.log=stderr.log. |
| server.shutdown.port | The port where the server is listed for shutdown. |
| jcdAlias | For example, jcdAlias=caPathway |
| dbmsName | Name of the database. For example, dbmsName=Oracle. |
| jdbcLevel | Java Database Connectivity (JDBC) level. For example, jdbcLevel=1.0. |
| jdbcRuntimeDriver | JDBC driver for your database. The proper value for Oracle is: oracle.jdbc.driver.OracleDriver The proper value for SQL profiling of OJB is: com.p6spy.engine.spy.P6SpyDriver |
| urlProtocol | For example, urlProtocol=jdbc. |
| urlSubprotocol | For example, urlSubprotocol=oracle. |
| urlDbalias | In the format: urlDbalias=thin:@[hostname]:1521:[schema name] where thin is your driver and 1521 is the port number. Enter your hostname and schema name. |
| databaseUser | Provide caArray application database user name which can view caArray tables. |
| databasePassword | Provide caArray application database user password which can view caArray tables. |
| SqlInLimit | The SqlInLimit entry limits the number of values in IN-sql statement, -1 for no limits. The default value for Oracle is 1000. |


| <i>Parameter</i> | <i>Description</i> |
|---|--|
| <code>file.storage.destination</code> | The is the directory in which where caArray data and experiment design files are stored on the caArray server. Windows users should use forward slashes, for example <code>c:/myfiles/caarray/files</code> . Do not use a trailing slash. The application creates a containing directory, called "caarrayftp," and appropriate subdirectories, within the directory specified for example, <code>file.storage.destination=c:/mydocuments/caarray/files</code> . The default value is <code>/share/content/caarray</code> . |
| <code>java.naming.provider.url</code> | The default configuration points to the caArray that's running on localhost and the caArray JBoss server has an RMI registry running on the default port (1099). If this is not the case, then you must edit the <code>java.naming.provider.url</code> property value to be the port of the JBoss RMI registry for the server running caArray portal. The default value is <code>java.naming.provider.url=localhost:1099</code> . |
| <code>log4j.rootCategory</code> | The default value is <code>ERROR, A1, A2</code> . |
| <code>log4j.logger.org.apache.ojb.broker.accesslayer.sql.SqlGeneratorDefaultImpl</code> | The default value is <code>DEBUG</code> . |
| <code>log4j.logger.org.apache.ojb.broker.accesslayer.JdbcAccessImpl</code> | The default value is <code>INFO</code> . |
| <code>log4j.logger.org.apache.ojb.broker.core.PersistenceBrokerImpl</code> | The default value is <code>DEBUG</code> . |
| <code>log4j.logger.org.apache.ojb.broker.platforms.PlatformOracleImpl</code> | The default value is <code>DEBUG</code> . |
| <code>log4j.logger.org.apache.ojb.broker.query.QueryFactory</code> | The default value is <code>DEBUG</code> . |
| <code>log4j.logger.org.apache.ojb.broker.metadata.RepositoryXmlHandler</code> | The default value is <code>DEBUG</code> . |
| <code>log4j.logger.org.apache.ojb.broker.accesslayer.ConnectionManager</code> | The default value is <code>DEBUG</code> . |
| <code>log4j.logger.org.apache.ojb.broker.accesslayer.StatementManager</code> | The default value is <code>DEBUG</code> . |
| <code>log4j.logger.org.apache.ojb.odmg</code> | The default value is <code>ERROR</code> . |
| <code>log4j.logger.gov.nih.nci.common.remote.rmi.RMIBinder</code> | The default value is <code>DEBUG</code> . |
| <code>log4j.logger.gov.nih.nci.common.remote.rmi.RMISearchCriteriaHandlerProxy</code> | The default value is <code>DEBUG</code> . |

| <i>Parameter</i> | <i>Description</i> |
|---|--|
| <code>log4j.logger.gov.nih.nci.common.persistence.Query2PBQ</code> | The default value is INFO. |
| <code>log4j.logger.gov.nih.nci.common.persistence.SC2Query</code> | The default value is INFO. |
| <code>log4j.logger.gov.nih.nci.mageom.util.MAGEDiscriminatorConverter</code> | The default value is INFO. |
| <code>log4j.logger.gov.nih.nci.common.persistence.SecureSessionPersistence</code> | The default value is DEBUG. |
| <code>log4j.logger.gov.nih.nci.common.persistence.SecurityManagerFactory</code> | The default value is DEBUG. |
| <code>log4j.logger.gov.nih.nci.common.persistence.SearchInterceptor</code> | The default value is DEBUG. |
| <code>log4j.logger.gov.nih.nci.common.persistence.SecurityFilter</code> | The default value is DEBUG. |
| <code>junit.report.dir</code> | The default value is <code>target/test/junit-reports</code> |
| <code>junit.build.dir</code> | The default value is <code>target/test/build</code> |
| <code>source.junit.dir</code> | The default value is <code>test/src/java</code> |
| <code>client.test.user</code> | The default username used by tests that extend <code>gov.nih.nci.mageom.test.MageClientTestCase</code> . The default value is <code>testcaarray</code> . |
| <code>client.test.password</code> | The default password for the username value of <code>client.test.user</code> property. The default value is <code>testcaarray</code> . |
| <code>reports.dir</code> | The default value is <code>target/reports</code> |
| <code>coverage.xml.dir</code> | The default value is <code>\${reports.dir}/cobertura-xml</code> |
| <code>coverage.html.dir</code> | The default value is <code>\${reports.dir}/cobertura-html</code> |

Table 1 *build.properties* parameters

| | |
|---|---|
| <p>NOTE:</p>  | <p>JBoss should be running an instance of the caArray application. The MAGE-OM server utilizes the security in caArray to filter the results that are returned from the database. The <code>java.naming.provider.url=localhost:1099</code> setting in <code>build.properties</code> points to the caArray JNDI port to look up the security EJB. This is different from the RMI server registry port. You will see messages in the JBoss log files (<code>caarray.log/server.log</code>) when security EJB is accessed.</p> <p>Make sure that the JNDI host/port are the same in MAGE-OM <code>build.properties</code> <code>java.naming.provider.url=localhost:1099</code> and also <code>jndi.properties</code> in <code>jboss/server/default/conf</code> point to the same server/port. The default in <code>jndi.properties</code> is <code>localhost:1099</code> if no entry is specified.</p> |
|---|---|

When you have completed these steps, you should see the code being compiled and packaged. When that is finished, you have the fully-configured and packaged MAGE-OM client and server under the deployment directories `{MAGE-OM_ROOT}/target/{target.dir}/client` and `{MAGE-OM_ROOT}/target/{target.dir}/server` (indicated by `{deploy_directory}`).


| | |
|---|---|
| <p>NOTE:</p>  | <p>If the build is not done on the destination server, the <code>mageom-server.zip</code> needs to be copied to the destination server. Unzip the files in any directory.</p> |
|---|---|

Starting and Stopping MAGE-OM

To start the MAGEOM server, you can follow the first set of steps or the second set of steps, described as follows:

| Step | Action |
|-----------|---|
| 1 | Make sure <code>JAVA_HOME</code> is pointing to correct Java 2 SDK. |
| 2 | Change to <code>{deploy_directory}/local/server</code> and type (if you have Cygwin installed locally or if u are on unix machine): <code>Source ./mageom_vars.sh</code> (in bash or simple execute in other shells). |
| 3 | Start the mage-om server: <code>./start_mage.sh.</code> |
| OR | |
| 3 | Double-click: <code>{deploy_directory}\local\server\bin\startmgrs.bat.</code> in Windows. |

Either choice opens a new console window where you see configuration information logged. If you have problems starting MAGE-OM, see *Troubleshooting* on page 13.

| | |
|---|--|
| <p>NOTE:</p>  | <p>See Appendix D for user experiences that describe starting the MAGE-OM server on Linux.</p> |
|---|--|

To stop the MAGEOM server, you can either:

| Step | Action |
|-----------|--|
| 1 | Change to <code>{deploy_directory}/local/server</code> and type (if you have Cygwin installed locally): <code>./stop_mage.sh</code> |
| OR | |
| 1 | (Windows only) Double-click <code>{deploy_directory}\local\server\stopmgrs.bat</code> . |

Either choice closes the console window and stops the MAGE-OM server

Testing MAGE-OM

Once the server is running, you should run some tests to verify that everything is working properly. The basic JUnit tests are defined in: `{MAGE-OM_ROOT}/test/src/java/gov/nih/nci/mageom/test`.

Complete the following steps to test MAGE-OM:

| Step | Action |
|------|--|
| 1 | Copy <code>{MAGE-OM_ROOT}\lib\junit.jar</code> to <code>{ANT_HOME}\lib</code> to run the JUnit tests; the MAGE-OM API tests use JUnit (http://junit.sourceforge.net). |
| 2 | Make sure that the Jars in the library are in the classpath |
| 3 | Edit your test to change your username/password in the JUnit test to refer to the username/password for the account created on the caArray portal. |
| 4 | To compile the tests, type the following from the <code>{MAGE-OM_ROOT}</code> directory: <code>ant compile:tests</code> |
| 5 | To run the tests, type the following from the <code>{MAGE-OM_ROOT}</code> directory: <code>ant test:client</code> |

| Step | Action |
|------|--|
| 6 | To see how to use the search criteria classes, look at <code>test/src/java/gov/nih/nci/mageom/search/DirectableSearchCriteriaTest.java</code> . The <code>testDirectable</code> test shows how to point at multiple servers. |

All the tests should pass. If not, see [Troubleshooting](#) (number 4) on page 13 for more information. See [Appendix B](#) for experiences from the caArray user community on installing and testing MAGE-OM.

Troubleshooting

The following suggestions are included to help you troubleshoot should you have any problems while following the procedures in this installation guide. If these suggestions do not solve your issues, contact NCICB Application Support. See [Contacting Application Support](#) on page 20.

1. One of the most common errors found when MAGE-OM does not work is your `java.policy` file is not set up correctly. To see if this is your problem, add the following to the top of the file and comment out anything else to grant all permissions:

```
grant {
    permission java.security.AllPermission;
};
```

Since `AllPermission` grants permissions to everything, you should refine the permissions for your environment. There is a `java.policy` file included in the source code in

```
{MAGE-OM_ROOT}/conf/unix/java.policy and
{MAGE-OM_ROOT}/conf/windows/java.policy.
```

2. If you receive an out of memory error on the server, then try running MAGE-OM with more memory. Try setting a larger `-Xmx` in `startmgrs.sh` (`.bat`)


```
jcmd="{JAVA_HOME}/bin/java -Xmx512m".
```
3. If you receive an out of memory error when compiling, then try increasing the max heap size of Ant by setting the environment variable `ANT_OPTS=-Xmx640m`. In addition, you may need to increase the maximum memory allocated to the `<iacj>` task by specifying the `maxmem="256M"` attribute, for example:

```
<iacj sourceroots="{source.java.dir}"
  destDir="{build.classes.dir}" fork="true" maxmem="256M">
```

4. It is important to use the `xerces.jar` included in the distribution. Otherwise, you may receive OJB parser runtime errors while having `xerces.jar` in the path.
5. If you have problems starting MAGE-OM, then try the following:
 - a. Make sure you have the right to execute it.

- b. If you are on a UNIX machine, then you might need to convert files to UNIX by using the `dos2unix` command.
- c. Make sure there are no control characters in any files.
- d. Reference the MAGE-OM log files, `mage.log` and `nohup.out` in `{MAGE-OM_ROOT}/target/{target.dir}/server` to see if MAGE-OM has started or if there are other messages.

Appendix A Example build.properties file

```
#####
# Build Properties #
#####

source.dir=src
source.java.dir=${source.dir}/java
generated.source.java.dir=${build.dir}/generated
source.resource.dir=conf

docs.dir=docs
javadoc.dir=target/javadoc
deployment.docs.dir=${docs.dir}/Deployment

build.dir=build
build.lib.dir=lib
build.classes.dir=${build.dir}/classes/
build.resources.dir=${build.dir}/resources/

ext.dir=ext

#<rmic> task compiler attribute value [sun|kaffe|weblogic]
build.rmic=sun

target.dir=local

# Valid values are 'unix' or 'windows'
target.platform=windows
target.shell=sh

client.jar.name=mageom-client
server.jar.name=mageom-server

#Path to JAVA_HOME root directory that the server use.
#The default is ${env.JAVA_HOME}, which is based on
#running the server on the local machine.
server.java_home=${env.JAVA_HOME}

mageom.version=1.6

#####
# RMI Properties #
#####

# The host and port that RMI clients can connect to.
# It should be the public DNS name that RMI clients
# can connect to.
# These two properties will make up the default searchLocation URL
rmi.public.host=localhost
rmi.public.port=8999

# The host and port that the RMI server binds to.
# Will become value of java.rmi.server.hostname
```

```

rmi.server.host=localhost
# Will become value of java.rmi.registry.port
rmi.server.port=8999
# Will become value of java.rmi.data.port
# If clients will not be running outside of the a
# firewall, then set this to -1.
#rmi.server.data.port=9999
rmi.server.data.port=-1

rmi.log=stderr.log

# The port at which the server will listed for shutdown.
server.shutdown.port=5468

#####
# OJB Properties #
#####

jcdAlias=caPathway
dbmsName=Oracle
jdbcLevel=1.0
#for production use the Oracle Driver
jdbcRuntimeDriver=oracle.jdbc.driver.OracleDriver
#for SQL profiling of OJB use the following
#jdbcRuntimeDriver=com.p6spy.engine.spy.P6SpyDriver
urlProtocol=jdbc
urlSubprotocol=oracle
urlDbalias=thin:@localhost:1521:caarray
databaseUser=caarrayop
databasePassword=password

#The SqlInLimit entry limits the number
#of values in IN-sql statement, -1 for no limits.
#The default value for Oracle is 1000.
SqlInLimit=1000

#####
# caArray Properties #
#####
# file.storage.destination is the local directory in which where caArray
data and
# experiment design files are stored
# Windows users should use forward slashes, for example
c:/myfiles/caarray/files
# Do not use a trailing slash
# The application creates a containing directory, called "caarrayftp," and
appropriate subdirectories,
# within the directory specified below
#
# example:
# file.storage.destination=c:/mydocuments/caarray/files
#####
file.storage.destination=/share/content/caarray

#####
# Security Properties #
#####

```

```

#java.naming.provider.url=localhost:1099
# This is for cbioqa.nci.nih.gov
java.naming.provider.url=localhost:1099

#####
# Logging Properties #
#####
log4j.rootCategory=ERROR, A1, A2
log4j.logger.org.apache.obj.broker.accesslayer.sql.SqlGeneratorDefaultImpl=DE
BUG
log4j.logger.org.apache.obj.broker.accesslayer.JdbcAccessImpl=INFO
log4j.logger.org.apache.obj.broker.core.PersistenceBrokerImpl=DEBUG
log4j.logger.org.apache.obj.broker.platforms.PlatformOracleImpl=DEBUG
log4j.logger.org.apache.obj.broker.query.QueryFactory=DEBUG
log4j.logger.org.apache.obj.broker.metadata.RepositoryXmlHandler=DEBUG
log4j.logger.org.apache.obj.broker.accesslayer.ConnectionManager=ERROR
log4j.logger.org.apache.obj.broker.accesslayer.StatementManager=DEBUG
log4j.logger.org.apache.obj.odmg=ERROR

log4j.logger.gov.nih.nci.common.remote.rmi.RMIBinder=DEBUG
log4j.logger.gov.nih.nci.common.remote.rmi.RMIsearchCriteriaHandlerProxy=DEBU
G
log4j.logger.gov.nih.nci.common.persistence.Query2PBQ=INFO
log4j.logger.gov.nih.nci.common.persistence.SC2Query=INFO
log4j.logger.gov.nih.nci.mageom.util.MAGEDiscriminatorConverter=INFO
log4j.logger.gov.nih.nci.common.persistence.SecureSessionPersistence=DEBUG
log4j.logger.gov.nih.nci.common.persistence.SecurityManagerFactory=DEBUG
log4j.logger.gov.nih.nci.common.persistence.SearchInterceptor=DEBUG
log4j.logger.gov.nih.nci.common.persistence.SecurityFilter=DEBUG

#####
# Testing Properties #
#####
junit.report.dir=target/test/junit-reports
junit.build.dir=target/test/build
source.junit.dir=test/src/java

client.test.user=testcaarray
client.test.password=testcaarray

# Used by test.properties file
# The location of test data files.
# Typically this is the local directory of the caarraytestdata module from
CVS
# (e.g. C:/ncicb_cvs/caarraytestdata )
test.data.dir=/ncicb_cvs/caarraytestdata


# Used by test.properties file
# TODO: add description
test.username=testuser

# Used by test.properties file
# TODO: add description
test.password=password

```

```
#####  
# Extend Build Properties #  
#####  
reports.dir=target/reports  
  
#####  
# Cobertura Properties #  
#####  
# Coverage reports are deposited into these directories  
coverage.xml.dir=${reports.dir}/cobertura-xml  
coverage.html.dir=${reports.dir}/cobertura-html
```

Appendix B XML Parser Issue 1.01

| | |
|--|--|
| <p>WARNING!</p>  | <p>These notes have not been tested at NCICB but are here for your use. They have been submitted by the caArray user community (for more information, see caArray Listserv for developers and installers at: http://list.nih.gov/archives/caarray_developers-l.html).</p> |
|--|--|

The following information is quoted, by permission, directly from emails to the caArray Listserv.

From: For caArray developers and installers
 [mailto:CAARRAY_DEVELOPERS-L@LIST.NIH.GOV] On Behalf Of Scott Li
 Sent: Wednesday, April 06, 2005 1:45 PM
 To: CAARRAY_DEVELOPERS-L@LIST.NIH.GOV
 Subject: Re: Error while running mageom at local --OJB issues
 ...Here is what I think what we should pay attention to:

- a. When compiling mageom, the ant script use xerces.jar, see below code:

```
<path id="cp.compile">
  <fileset dir="${build.lib.dir}">
    <patternset refid="server-ct-jars"/>
    <include name="xerces.jar"/>
  </fileset>
</path>
```

This is a problem when I reuse the same path code for running the mageom server, (I am sorry here: I have to rewrite the ant script for run-server and shutdown-server, instead of use the default generated script, because that won't work with painful struggle)

2. I think the xerces.jar should not be included in the cp.compile path, because if you do, the OJB would pick this up at runtime as xml sax parser for reading OJB mapping files, that would lead to something strange as seen in email before. Therefore, exclude this jar at runtime, and at compile time for mage classes, this should also be excluded, otherwise you will get AbstractMethod error while clients send request to server, the Abstract Method errors are mostly likely caused by different version of jar files used during compile and runtime. In our case, the killer is xerces.jar.

3. But, if we exclude the xerces.jar from cp.compile, the compile would fail because the file gov.nih.nci.mageom.util.DOMWriter needs this. Strangely, the DOMWriter is the single class independent of other code. At least when I removed this file and compiled again without any errors.

Summary:

Remove gov.nih.nci.mageom.util.DOMWriter file and specifically include jar files in your classpath for both compile and runtime, making sure the jar file xerces.jar are excluded and all other jars are same version. You got it run!!

Contacting Application Support

NCICB <http://ncicb.nci.nih.gov/NCICB/support>
Application Telephone: 301-451-4384
Support Toll free: 888-478-4423
