

caArray 054 - Step-By-Step Guide to Install and Use caArray Importer

Step-By-Step Guide to Install and Use caArray Importer

Topic: caArray Installation and Upgrade

Release: caArray 2.3 and 2.4.x

Date entered: 03/19/2010

Contents of this Page

- [About caArray Importer](#)
- [Download caArray Importer](#)
- [caArray Importer Compatibility](#)
- [caArray Importer Distribution Contents](#)
- [Pre-Requisites](#)
- [Initial Configuration](#)
- [Preliminary Steps for caArray Versions 2.4 and Higher](#)
 - [For ALL Versions of caArray 2.4 And Higher](#)
 - [For caArray Version 2.4.0.x ONLY](#)
 - [For caArray Version 2.4.1 ONLY](#)
- [Repackaging and Redeploying the caArray Server](#)
 - [Step 1. Update the caarray.ear file](#)
 - [Step 2. Deploy the updated caarray.ear file](#)
 - [Step 3. Updating the caArray Client Library](#)
- [Using the caArray Importer API](#)
 - [Scenario 1. Upload Affymetrix 250K Nsp and Sty SNP Data](#)
 - [Scenario 2. Upload Affymetrix Expression Files](#)
- [Have a comment?](#)

About caArray Importer

Developed by Dr. Matthew Eldridge, Head of the Bioinformatics Core facility at Cancer Research UK's Cambridge Research Institute, caArray Importer v1.0.2 is an open-source package that extends the caArray Java API to provide programmatic access to the following functionality:

1. Retrieval of imported array designs
2. Creation, modification and deletion of projects
3. Upload, validation and import of array data files
4. Modification of file type
5. File removal

Download caArray Importer

caArray Importer v1.0.2 can be [downloaded](#).

caArray Importer Compatibility

This version of the caArray API extension is compatible with versions 2.3 and later. It is not compatible with caArray version 2.2.1 or earlier releases.

Note that, for versions 2.4 and later, some of the source code and build files in the caArray Importer distribution must be modified before they can be compiled and deployed. The modifications, which are described later in this article, are slightly different depending on the exact version of caArray you are running (i.e., 2.4.0.2 versus 2.4.1).

caArray Importer Distribution Contents

caArray Importer Distribution Contents

Components	Description
LICENSE.txt	The Apache License version 2.0 under which the source code in this distribution is released.
build.xml	Apache Ant build file for compiling source code, updating the caArray server and client library, and running the example client application.

build.properties	Properties file for the Ant build in which the locations of the caArray server ear and client distribution should be set.
apidocs	Contains the Javadoc documentation for both server and client classes that constitute the caArray Importer API as well as the example client application.
caarray-ejb	Contains source code for the Enterprise JavaBean (EJB) that provides programmatic access to caArray functionality for creating projects, uploading, validating and importing array data files, etc.
caarray-client	Contains source code for the modified CaArrayServer class (no longer marked final to enable subclassing) used to connect to and access the caArray server and a subclass that provides access to the interface for the caArray Importer EJB.
example	Contains source code for an example client application that demonstrates how to use the caArray Importer API.

Pre-Requisites

caArray Importer Pre-requisites

Components	Description
Java 2 Platform Standard Edition 5.0 Update 10 (J2SE 5.0)	Building and running Java applications
Apache Ant 1.7.0	Building and running Java applications
Set JAVA_HOME and ANT_HOME environment variables	Refer to the caArray installation guide for further details
caArray legacy client distribution	Download from the caArray GForge files archive (login required)

Initial Configuration

The initial configuration of caArray Importer is specified by multiple parameters in the `build.properties` file, which is located in the root level of the caArray Importer distribution archive. For the parameters below, in most cases a user only needs to update the `server.base.dir` property. Note that the value for this property varies depending on the particular version of caArray you are running.

caArray Importer Initial Configuration

Parameters	Description
<code>client.legacy.zip=caarray-client-legacy.zip</code>	The location of the caArray Java client distribution zip file. By default it is assumed to be in the current working directory and named <code>caarray-client-legacy.zip</code> .
<ul style="list-style-type: none"> <code>server.base.dir=C:/programs/caarray/caarray2_3_0</code> (for v2.3) <code>server.base.dir=C:/programs/caarray/caarray2_4_0_1</code> (for v2.4.0.1) <code>server.base.dir=C:/programs/caarray/caarray2_4_1</code> (for v2.4.1) 	The base directory in which the caArray server was installed. The directory provided depends on the exact caArray version you are running.
<code>server.deploy.dir=\${server.base.dir}/jboss-4.0.5.GA/server/default/deploy</code>	The actual deployment directory. Normally this does not need to be changed. This can be left empty if you do not have write access to the deployment directory. In that case, you must manually copy it over after the <code>caarray.ear</code> file is updated.
<code>server.ear=\${server.deploy.dir}/caarray.ear</code>	The target caarray server ear file. Normally this does not need to be changed.

Preliminary Steps for caArray Versions 2.4 and Higher

If you are running caArray v2.4 or higher, you must perform the steps outlined below before proceeding to repackage and redeploy the caArray server enterprise archive. Note that some of the steps vary depending on the exact version of caArray you are running.

For ALL Versions of caArray 2.4 And Higher

The build.xml file included in the stock caArray Importer distribution does not work with caArray v2.4 and higher. You must either download this modified [build.xml](#) file or manually edit the stock build.xml as described below:

1) In the property declarations towards the top of the file, you will see the following line of code which declares the property `client.libs` :

```
<property name="client.libs" value="* /lib/java/ *.jar"/>
```

Add the following line of code beneath it to declare another property called 'client.libs.grid':

```
<property name="client.libs.grid" value="* /lib/grid/ *.jar"/>
```

2) Then, locate the following snippet of code further down in the build.xml file...

```
<path id="example.classpath">
  <pathelement location="{example.classes.dir}"/>
  <pathelement location="{updated.client.jar}"/>
  <fileset dir="{client.dist.dir}">
    <include name="{client.libs}"/>
  </fileset>
</path>
```

...and add the newly declared 'client.libs.grid' property by inserting the following line of code after `<include name="{client.libs}"/>` '...

```
<include name="{client.libs.grid}"/>
```

...so that the modified code snippet will appear as follows (inserted line of code highlighted in bold):

```
<path id="example.classpath">
  <pathelement location="{example.classes.dir}"/>
  <pathelement location="{updated.client.jar}"/>
  <fileset dir="{client.dist.dir}">
    <include name="{client.libs}"/>
    <include name="{client.libs.grid}"/>
  </fileset>
</path>
```

For caArray Version 2.4.0.x ONLY

You must revise a single line of code in the server-side Enterprise Java Bean (EJB) source file `CaArrayImporterServiceBean.java` as described below. This file is located at the following path within the caArray Importer distribution archive: `\caarray-ejb\src\main\java\org\cruk\cri\caarray\services` :

Replace line 401 in the original source:

```
Project project = projectManagementService.getProject(projectId);
```

With the following line of code:

```
Project project = projectManagementService.getProjectByPublicId("EXP-"+projectId.toString());
```

For caArray Version 2.4.1 ONLY

You must revise three different lines of code in the server-side Enterprise Java Bean (EJB) source file `CaArrayImporterServiceBean.java` as described below. This file is located at the following path within the caArray Importer distribution archive: `\caarray-ejb\src\main\java\org\cruk\cri\caarray\services` :

First, replace line 46 in the original source file:

```
import gov.nih.nci.caarray.util.UsernameHolder;
```

with this one:

```
import gov.nih.nci.caarray.util.CaArrayUsernameHolder;
```

Then, replace line 188 in the original source file:

```
Person person = new Person(UsernameHolder.getCsmUser());
```

With this one:

```
Person person = new Person(CaArrayUsernameHolder.getCsmUser());
```

Finally, replace line 401 in the original source file:

```
Project project = projectManagementService.getProject(projectId);
```

With the following line of code:

```
Project project = projectManagementService.getProjectByPublicId("EXP-"+projectId.toString());
```

Repackaging and Redeploying the caArray Server

The following steps repack the caArray server enterprise archive, `caarray.ear`, with the additional EJB and associated classes that constitute the caArray Importer API extension.

Step 1. Update the `caarray.ear` file

```
$ ant update-ear
```

Note that '\$' represents the command prompt and may differ depending on your operating system or shell.

There are four steps involved in this process:

1. Copies the `caarray.ear` archive to the current directory
2. Unpacks its contents to a local subdirectory
3. Compiles the caArray Importer EJB and associated classes
4. Repackages the archive with these additional classes

Step 2. Deploy the updated `caarray.ear` file

```
$ ant deploy-ear
```

This step does not work unless you have write access to the caArray JBoss server deployment directory (see [initial Configuration](#)). In that case, you should manually copy the `caarray.ear` file to the `${server.base.dir}/jboss-4.0.5.GA/server/default/deploy` directory on which the caArray server is running.

There is no need to restart the caArray JBoss server. This step should recognize the newly deployed archive and automatically redeploy it. Redeployment may take a few minutes to complete. You can check progress in the server log, `${server.base.dir}/jboss-4.0.5.GA/server/default/log/server.log`

Step 3. Updating the caArray Client Library

This step repackages the client library with additional classes required to make use of the caArray Importer API.

```
$ ant update-client-jar
```

This unpacks the legacy client distribution zip file, compiles the modified CaArray Server class and its subclass (see content description for `caarray-client` above), and repackages the client library jar file with these classes and some additional classes from the caArray server required by the caArray Importer.

Using the caArray Importer API

Scenario 1. Upload Affymetrix 250K Nsp and Sty SNP Data

An example client-side program is provided to demonstrate the creation of a new project and the upload, validation and import of array data files. The source code can be found in the following file:

```
example/src/main/java/org/cruk/cric/caarray/client/CaArrayImporterExample.java
```

The example client program creates a new project and uploads all files in the `arraydata` directory.

To run the example client program, follow the steps below:

1. Make sure that your Java classpath includes the updated client library. (This is taken care of if you run the example using Ant as shown below).
2. Upload the array design files for the Affymetrix Mapping 250K Nsp and Sty chips into caArray using caArray web interface.
3. Create a directory called **arraydata** in the same location as that in which you execute the client-side program.
4. Copy a few Affymetrix 250K data files (CEL, CHP) and optionally MAGE-TAB IDF and SDRF files to the **arraydata** directory .
5. Run command below:

```
$ ant compile-example  
$ ant run-example
```

Scenario 2. Upload Affymetrix Expression Files

To upload Affymetrix expression files, follow the steps below:

1. Upload the related array designs into the caArray web interface.
2. Modify the values for the following constants in `CaArrayImporterExample.java`. Note that the values provided are for illustration purpose only and must be replaced:

```
private static final String SERVER_HOST = "yourCaArrayServerHost";
private static final int SERVER_JNDI_PORT = 31099;
private static final String USERNAME = "yourUsername";
private static final String PASSWORD = "yourPswd";
public static String TITLE = "Test Affymetrix Expression Project";
public static String PROVIDER = "Affymetrix";
public static String ASSAY_TYPE = "Gene Expression";
public static String[] ARRAY_DESIGNS = new String[] \{ "HG-U133_Plus_2" \};
```

3. If you have your data files stored under a directory (for example: `C:/data/caarray/MAGEdataFiles`) different from the **arraydata** created in scenario 1, replace the directory **arraydata** with the path to that directory as shown below in the function "main". It can be either relative or absolute path:

```
client.uploadFiles(project, "C:/data/caarray/MAGEdataFiles");
```

4. Copy a few Affymetrix HG-U133_Plus_2 data files (CEL, CHP) and optionally MAGE-TAB IDF and SDRF files to the **C:/data/caarray/MAGEdataFiles** directory .
5. Run the command below:

```
$ ant compile-example
$ ant run-example
```

Have a comment?

Please leave your comments in [caArray End User Forum](#)