CAARRAY

DATA MIGRATION AND CLEANUP

Instructions for Upgrading to caArray Release 1.3.1

NATIONAL CANCER INSTITUTE®

Center for Bioinformatics

Revised March 31, 2006

Table of Contents

Introduction

The primary purpose of this document is to provide detailed instructions on how to effectively migrate and cleanse legacy (pre- 1.3.1) caArray data repositories to the caArray 1.3.1 instance. Data Migration and Cleanup includes:

- Migrating to the caArray 1.3.1 database schema
- Cleanup Array Designs provided with pre-caArray 1.3.1 versions as seed data

This document also includes instructions and guidance on how to delete previously submitted (pre-version 1.3.1) Array Designs and when to re-submit Array Designs. If you are installing caArray as a fresh install, these instructions do not apply to you. These procedures are only for sites that have a prior installation which they would like to upgrade. If you are unsure if you should upgrade your current installation or install a fresh version of caArray, you should contact NCICB Application Support ncicb@pop.nci.nih.gov to discuss your options and the relative merits of upgrading versus a fresh install.

NOTE:	These instructions apply to an installation that uses ORACLE database only.

Target Audience

This document is intended for administrators who have installed previous versions (pre 1.3.1) of caArray. This document is not intended for users of the NCI caArray instance, because the NCI caArray instance is maintained by the NCI caArray Development Team and Application Support.

Document Organization

This document is organized in the following sections:

- Overview of Migration and Cleanup
- Description of Migration and Cleanup Scripts
- Procedures for Migrating and Cleanup of pre-1.3.1 caArray Instances

Overview

What is migration?

caArray 1.3.1 has a new feature that supports the audit trail tracking of files being uploaded into caArray. This additional feature required additions to the database schema. Previous versions of caArray must add these additional schema elements to the caArray database to run caArray 1.3.1. The migration procedure will add the new schema elements into your caArray installation without affecting the current datasets residing in the database.

For a more complete description of the migration process, please refer to *Migration and Cleanup* Procedures *on page 7*.

With the release of caArray 1.3.1, the caArray project is introducing a new rigor in data curation and database and file integrity. This new emphasis on data and files allows for the auditing of files introduced into the system in such a way that the steps that were used to introduce the data originally can be redone if resubmission or migration to a new environment ever becomes necessary. There are numerous software fixes and enhancements associated with caArray 1.3.1, and the path forward to newer releases of caArray will depend on some of the features being introduced during this release.

What is cleanup?

Analysis of the caArray production database has been performed, and it has been determined that some of the Array Design entries in previous versions of caArray were not 100% correctly persisted in the database. The removal of invalid Array Designs that were included in pre-caArray 1.3.1 seed data and creation of new Array Designs for previously submitted hybridizations is required for proper operation of the caArray application as it moves forward into newer releases. Cleanup of your database is required because of the previously provided seed data designs that were persisted incorrectly. Cleanup entails the deletion of the Array Designs in preparation for potential resubmission.

What is seed data?

For each release, caArray has provided “seed” data, which consists of a variety of data required to begin using the caArray application. This seed data is meant to provide external installations with a starting point to begin using caArray. Since loading of Affymetrix designs has been a time consuming process, seed data was also originally meant to bypass the time-consuming step of loading the designs from MAGE-ML.

All of the seed data Affymetrix Array Design information provided prior to this release were incorrectly persisted and will be deleted as part of the caArray 1.3.1 installation process. Other seed data information such as initial user information and other initial data structures will remain.

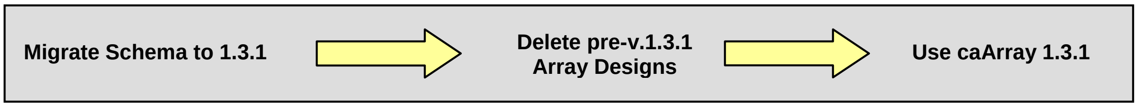
With caArray 1.3.1, there is a script which deletes earlier seed data Array Designs. If you have submitted hybridization files associated with these Array Designs, contact the Application Support team at NCICB ncicb@pop.nci.nih.gov for help in reconnecting these files once the Array Design has been properly persisted on your system.

caArray does not provide seed Array Designs with version 1.3.1, because the process for submitting these designs into the 1.3.1 system is quicker than with previous versions. (MAGE-ML parsing is deferred in caArray 1.3.1).

Upgrade and Cleanup Scripts

The Upgrade and Cleanup scripts only support the migration of the caArray database to version 1.3.1 and the deletion of aArray seed data Array Designs.

The process for existing installations to begin using caArray 1.3.1 upgrading from 1.3 is as follows:



Upgrade Scripts

Upgrade scripts facilitate the migration of pre-1.3.1 caArray databases to the 1.3.1 database schema. The following upgrade scripts are provided: Upgrade scripts for migrating the caArray 1.2 database schema to the 1.3 database schema (required for local caArray 1.2 installations)
Upgrade scripts for migrating the caArray 1.3 database to the 1.3.1 database schema.

See
Migration and Cleanup Procedures on page 7 for more information about update scripts.

Verification Scripts

The verification script programs compare raw data values with those in the database and also with extracted values. The scripts have been used to validate that the Affymetrix Array Designs contain the correct number of features, reporters and composite sequences.

This section describes the use of validation, Affymetrix cleanup, and Array Design deletion scripts/programs. These programs validate the correct persistence of Array Designs in the database and verify that netCDF files are being correctly generated from the hybridization files that are based on the Array Design.

Affymetrix Array Design Verification

The Affymetrix Array Designs in the caArray database are generally considered correct when the total number of features, reporters, and composite sequences persisted in the database are equal to their respective numbers when those totals are extracted via other means.

Current caArray Validation Utilities

SQL Script:

- The following SQL Script queries the caArray Database for the number of features, composites, and reporters for a given Array Design.
- This script is used in conjunction with a shell script program which parses a MAGE_ML file to gather reporters, composites and sequences.

To retrieve the number of features, composites, and reports present in the database for a specific Affymetrix Array Design, follow these steps:

Step	Action
1	Login to the Oracle schema using sqlplus or toad.

Step	Action

2	Create the following indexes:
	`create index feature_fgrp_idx on feature(feature_group_id);`
	`create index feature_zone_idx on feature(zone_id);`
	`create index reporter_rgrp_idx on reporter(reporter_group_id);`
	`create index dem_report_id_idx on designelementmap`
	`     (reporter_id);`
	`create index dem_com_seq_id_idx on designelementmap`
	`     (composite_sequence_id);`
	`create index f_ctrlled_f_cfid_idx on featurecontrolledfeature`
	`     (controlled_feature_id);`
	`create index f_ctrl_f_cfid_idx on featurecontrolfeature`
	`     (control_feature_id);`
	NOTE: Depending on which client you use, each sql statement should end with a ";" or "/"

3	Gather schema statistics. This is required so that Oracle cost-based-optimizer can take advantage of the newly created indexes.
	`exec dbms_stats.gather_schema_stats('', estimate_percent=> 15,`
	`                                    cascade=>true)`

4	Run the following script:
	`select id, name, NUMBEROFFEATURES,`
	`   ( select count(*) from ARRAYDESIGNFEATUREGRP adfg,`
	`                         FEATURE f`
	`          where adfg.array_design_id = ad.id`
	`          and adfg.FEATURE_GROUP_ID = f.FEATURE_GROUP_ID)`
	`          n_feature,`
	`   ( select count(*) from ARRAYDESIGNCOMPOSITEGRP adcg,`
	`                         COMPOSITESEQUENCE cs`
	`        where adcg.array_design_id = ad.id`
	`        and adcg.COMPOSITE_GROUP_ID = cs.COMPOSITE_GROUP_ID )`
	`        n_comp_sequence,`
	`   (select count(*)from ARRAYDESIGNREPORTERGRP adrg, REPORTER r`
	`        where adrg.array_design_id = ad.id`
	`        and adrg.reporter_group_id = r.reporter_group_id )`
	`        n_reporter`
	`  from arraydesign ad`
	`/`

Shell Script

- Run the shell Script which extracts Array Design statistics from MAGE_ML.
- Save it under a file named `arrayDesignStats.csh`.

To run the script (from a UNIX box or cygwin under windows), follow these steps:

Step	Action
1	Enter cd to the directory that contains all the MAGE-ML files (*.XML).

2	Source `<path_to_arrayDesignStats.csh>/arrayDesignStats.csh > output.txt.`
	
	`#/bin/sh`
	`foreach f (``find . -name '*.XML' -print``)`
	` set feature = ``grep '<Feature identifier=' $f | wc -l`` `
	` set reporter = ``grep '<Reporter identifier=' $f | wc -l`` `
	` set cs = ``grep '<CompositeSequence ' $f | wc -l`` `
	`  echo $f, $feature, $reporter, $cs`
	`end`

The shell script is run against the raw MAGE-ML to extract, using regular expressions, the number of features, composites, and reporters**.** This is then compared with the numbers extracted from the SQL script above. If the numbers match, the design was persisted correctly.

Array Design Cleanup Scripts

Array Design Cleanup scripts include:

- Array Design Deletion Scripts – Deleting invalid Array Designs
- Affymetrix Array Design Verification Scripts – Verifying the persistence of Affymetrix Array Designs

The Cleanup process primarily involves the ability to identify and delete faulty Array Design instances from the caArray database. This feature is discussed in *Array Design Cleanup Script* on page 10.

To perform cleanup, follow these steps:

Step	Action
1	Remove invalid Affymetrix Array Designs that do not have associated hybridizations via the deletion script.
2	If hybridizations exist, it may be possible to enter a replacement Array Design in the caArray 1.3.1 instance, after which you can re-associate existing hybridizations.
3	Delete the original invalid Array Design.

Migration and Cleanup Procedures

Introduction

The primary goal of caArray 1.3.1 is to provide a robust file submission/ retrieval strategy, allowing users to download the original files that were submitted as well as annotations. This means CaArray 1.3.1 ensures that users can successfully submit files (Array Designs, hybridizations, MAGE-ML documents, other experiment files) and associated annotations, and retrieve submitted files and associated annotations.

NOTE: All functionality associated with file processing – NetCDF, MAGE-ML import, etc – is disabled in this release of caArray. Refer to caArray 1.3.1 Release Notes for more details. An upcoming release of caArray will re-enable file processing steps so that end users can leverage the bio-data-cube for analysis.

The migration and cleanup procedures provided below are typically used by system administrators and members of the technical staff as a reference guide to perform an upgrade of an existing caArray instance to caArray 1.3.1. Prior to performing the 1.3.1 upgrade, administrators should consult the following caArray documentation:

- *caArray 1.3.1 Release Notes*
- *caArray 1.3.1 Installation Guide*

Additional documentation for caArray 1.3.1:

- *caArray 1.3.1 Technical Guide*
- *caArray 1.3.1 User guide*

Prerequisites and Pre-install Instructions

Prerequisites to execute the upgrade program:

- Ability to start and stop caArray jboss server.
- Login privileges to the caArray account on the UNIX server, which typically holds the caArray home directory.
- Database account information for the caArray 1.3 database
- Database utilities, for backup and SQL shell (like sqlplus)
- Ant and jre142.

Upgrading caArray 1.2 Databases to caArray 1.3

NOTE:	This step needs to be performed only if all three of the following conditions apply: • If the current version of caArray is version 1.2 • If the current installation has existing data in database • If the current installation has a file system pertinent to hybridization file uploads (ie. .chp, .gpr files, etc). The script in this section migrates the data from deprecated tables to the database tables used in the caArray 1.3 version for managing hybridization file-related data. This is a preliminary step for moving into the caArray 1.3.1 environment and **must be performed if you are migrating from 1.2 to 1.3.1**.

BEFORE YOU BEGIN	Before you begin migration, follow these pre-migration instructions: 1. Ensure that Java is available. (Use the command `java -version` to verify.) 2. Ensure that `classes12.jar` (or `ojdbc14.jar`) is available. 3. The file system on which the uploaded microarrayfiles are kept must be accessible from the directory where you are running the migration program.

Follow these steps to complete the 1.2 to 1.3 migration:

Step	Action
1	Unzip the `caArrayUpgrade1.2.zip` package.
2	Enter the appropriate property values in the `migration.properties` file.
3	Compile the code with the following command: `Javac -classpath .;classes12.jar *.java`
4	Run the program with the following command: `Java -classpath .;classes12.jar MigrateHybridizationData`

Upgrading caArray 1.3 Databases to caArray 1.3.1

NOTE:

This step needs to be performed only if all three of the following conditions apply:

- If your current version of caArray is version 1.3
- If the current installation has existing data in database
- If the current installation has a filesystem pertinent to hybridization file uploads (ie. .chp, .gpr files etc).

If your installation does not meet the above criteria, you should probably install a fresh copy of caArray. If you are unsure about your site's status in this regard, contact NCICB Application Support (ncicb@pop.nci.nih.gov) for assistance in determining what the proper install path is correct for your site.

BEFORE YOU BEGIN

Before upgrading the caArray 1.3 databases and file repository, follow these pre-migration steps:

1. Stop the caArray server
2. Perform a cold backup of caArray 1.3 database. For Oracle 9.x, use the following command to perform a cold backup of the schema

   ```
   exp <user_id>/<password>@<sid>
   ```

   and follow the on-screen instructions, usually accepting the defaults.
3. Perform a backup of the caArray 1.3 file system. On a UNIX system, one way to do this is to tar the `/share/content` directory and move the tar to a different disk/partition.

Performing the Upgrade

Verify completion of the database and file system backup. Then, follow these steps to run the upgrade script:

Step	Action
1	Download the `caArrayUpgrade131.zip` to the caArray home directory.
2	Unzip `caArrayUpgrade131.zip` using gzip or an equivalent utility.

Step	Action
3	Change (cd) to the directory where the `caArrayUpgrade131.zip` file was extracted.
4	Login to caArray Oracle schema using sqlplus or Toad and run `caArray131_ddl.sql`.
5	Delete the `designElement` cache files. For example, `rm -rf /share/content/caarray/caarrayftp/microarrayfiles/designelements/*.txt`

Array Design Cleanup Script

Affymetrix Array Designs

While the Array Design Deletion script works for all Array Design types, the verification scripts apply to Affymetrix Array Designs only. The Affymetrix Array Design verification script identifies the cases where the number of reporters persisted in the Array Design section of the database was less than expected, based on comparison with the MAGE-ML file. If this case occurred in remote sites that submitted Array Designs, it may be possible to fix this by using a script supplied by Application Support, ncicb@pop.nci.nih.gov.

GAL-Based Designs

If a GAL-based Array Design is faulty in any way, one would simply delete the Array Design from the database and resubmit it. If the verification scripts indicate that Affymetrix Array Design persistence of features or composites does not match the numbers reported by the verification scripts, then currently, there is no alternative but to delete and reload the Affymetrix Array Design.

The caArray processes for submission of hybridization files now allows the submission of multiple files, making the process of resubmitting hybridization files easier, should you need to resubmit the Gal or Affymetrix Array Designs.

Array Design Deletion Script Details

The deletion script cycles through the Array Design table and iteratively deletes all records associated with a specific Array Design entry. The script deletes Array Design entries of any type and purges Array Design information from the caArray 1.3 Oracle database.

The deletion script deletes objects under the following MAGE-OM packages:

- ArrayDesign
- DesignElement
- BioSequence

Then, it updates the references under Array to ArrayDesign (set to NULL). Furthermore, the procedure does not delete any "good data". The premise is that any associated experiments, arrays, bioassays are considered good. Upon reload of the Array Design, the references to Array Design can be set to the new ID. To avoid excessive undo generation (rollback segment), it commits after each delete. On the other hand, the procedure can easily be modified to run in a single transaction when there is a sufficiently large rollback segment.

NOTES	• caArray does not disable or drop any constraints. This ensures data integrity and consistency. • This procedure may not be dealing with all the tables involved in the MAGE-OM packages. The script has successfully deleted approximately 40 Array Designs at the stage environment here at NCICB.

The following indexes are recommended to be created for better query (delete) performance. (un-indexed foreign keys can cause various problems during DML update/delete operations, as well as for a simple query).

```
create index feature_fgrp_idx on feature(feature_group_id);
create index feature_zone_idx on feature(zone_id);
create index reporter_rgrp_idx on reporter(reporter_group_id);
create index dem_report_id_idx on designelementmap (reporter_id);
create index dem_com_seq_id_idx on designelementmap (composite_sequence_id);
create index f_ctrlled_f_cfid_idx on featurecontrolledfeature
   (controlled_feature_id);
create index f_ctrl_f_cfid_idx on featurecontrolfeature (control_feature_id);
```

The following table depicts the parent/child relationships among the entities. The procedure deletes child tables and then moves up the ladder to delete rows from parent tables.

```
ARRAYDESIGN
**ZONEGROUP
****ZONE
```

```
******FEATURE
********FEATURECONTROLFEATURE
********FEATURECONTROLLEDFEATURE
********FEATUREDIMENSIONFEATURE
********FEATURELOCATION
********FEATUREINFORMATION
**********MISMATCHINFORMATION
**********MISMATCHINFORMATION
****ARRAYDESIGNCOMPOSITEGRP
****ARRAYDESIGNFEATUREGRP
****ARRAYDESIGNREPORTERGRP
******REPORTER
********IMMOBILIZEDCHARACTERISTICS
********DESIGNELEMENTMAP
**********FEATUREINFORMATION
************MISMATCHINFORMATION
****COMPOSITESEQUENCE
******BIOLOGICALCHARACTERISTICS
******COMPSEQDIMENSIONCOMPSEQ
******DESIGNELEMENTMAP
********FEATUREINFORMATION
**********MISMATCHINFORMATION
```

Cleanup Procedures

This section describes the process for Cleanup existing seed Array Designs.

Pre-requisites and Pre-install Instructions

1. Install caArray 1.3.1
2. Run the caArray 1.2 to 1.31 upgrade scripts (packaged with caArray 1.3.1 and described in the sections starting on page 8).

The following are pre-requisites to execute the clean-up/delete scripts:

- Ability to start and stop caArray jboss server.
- Login privileges to the caArray account on the UNIX server, which typically holds the caArray home directory.
- Database account information (including the password) for the caArray 1.3 database
- Access to database utilities, for backup and SQL shell (like sqlplus)
- Access must be configured for Ant and jre142.

BEFORE YOU BEGIN	Before you begin deletion or clean-up of the caArray 1.3 databases and file repository, it is recommended that you follow these steps:

Step	Action
1	Stop the caArray server.

2	Perform a cold backup of the caArray 1.3 database. For Oracle 9.x, the following command does a cold backup of the schema.
	`exp <user_id>/<password>@<sid>`
	and follow the on-screen instructions, usually accepting the defaults, where user and password are replaced with username and password of database user with sufficient privileges to perform backups and SID is the SID of the caArray database instance. An example might look like this:
	`exp jdoe/passmeThrough@caArray`
	which requests a backup using user "`jdoe`" with password "`passmeThrough`" for the JID of "`caArray`".

3	Perform a backup of the caArray 1.3 file system. On a UNIX system, one way to do this is to tar the `/share/content` directory and move the tar to a different disk/partition. For example: `cd /share/content ;   tar -cvf caArrayFiles.tar . ; mv caArrayFiles.tar  /opt/backupPartition/`

Running the Clean-up Script

If it is determined that you need to run the cleanup script for the Hgu133plus2 Array Design, proceed with the cleanup by following these steps:

Verify completion of the database and file system backup described in the section above. Then, follow these steps:

Step	Action
1	Login to caArray oracle schema using sqlplus or Toad.
2	To run the clean-up script, within sqlplus run "`@ hgu133plus2-cleanup.sql`".
3	Exit sqlplus (or Toad).

Run the Delete Script

Verify completion of the database and file system backup described in the section above. Then, follow these steps:

Step	Action
1	Login to caArray oracle schema using sqlplus or Toad as a privileged user.
2	Run `crarrpurge.sql` to create the stored procedure to delete Array Designs.
3	Follow the Readme to create recommended indexes which accelerate the deletion process.
4	Enter the following SQL command to see the Array Design names and IDs. a. From sqlplus, run `Select id,name from arraydesign;` This command lists all the Array Designs and their IDs. Alternatively if you know the Array design name (after having searched arrayDesigns in the caArray user interface, for example, you could issue a command like: b. From sqlplus, run `Select id,name from arraydesign` where name= `Affymetrix.com:PhysicalArrayDesign:hg_u95av2;` This returns the name and ID of a specific Array Design, which could then be used in the deletion process.

Step	Action

5	To delete an Array Design, execute
	`sqlplus> exec purge_array_design ( <id>)`
	where <id> is the primary key of your Array Design returned from the commands issued above:
	For example:
	`sqlplus> exec purge_array_design (1015897521113350)`
	would delete the Affymetrix.com:PhysicalArrayDesign:hg_u95av2 design.

6	Verify that the Array Design is no longer in the database by issuing the command: a. From sqlplus run `select name, id from arraydesign where id=1015897521113350;` substituting your arraydesign ID for the `1015897521113350` listed above. This should return no records indicating that the Array Design has been successfully completed.

Contacting Application Support

NCICB Application Support

http://ncicbsupport.nci.nih.gov/sw/

Telephone: 301-451-4384
Toll free: 888-478-4423