

# caArray 080 - A Step-by-Step Guide to Create a MAGE-TAB Template Specific to RPLA Project

## A Step-by-Step Guide to Create a MAGE-TAB Template Specific to RPLA Project

**Topic:** caArray Installation and Upgrade

**Release:** caArray 2.4.0

**Date entered:** 04/20/2011

### Contents of this Page

- [Introduction](#)
- [Create new provider\(s\)](#)
- [Create new Assay Type](#)
- [Check if caArray is a valid Term Source](#)
- [Add a new terminology term](#)
- [Create a dummy Array Design File for RPLA](#)
- [Upload the Dummy Array Design file in caArray](#)
- [Update Array\\_Design Table](#)
- [Create New File Types for RPLA data files](#)
- [Have a comment?](#)

## Introduction

Developed by Leandro Hermida, this step-by-step guide will help RPLA project user to create a RPLA specific MAGE-TAB template.

## Create new provider(s)

Create RPLA providers in the contact table as for other array providers:

```
insert into contact (discriminator, provider, name) values ('O', 1, 'Zeptosens');
```

After the insert, use the statement below to find the ID for the contact just created:

```
Select ID from contact where name = 'Zeptosens';
```

## Create new Assay Type

Create a new RPLA assay type in the database in the assay\_type table:

```
insert into assay_type (name) values ('RPLA');
```

## Check if caArray is a valid Term Source

Since a new technical term: **reverse\_phase\_protein\_array\_features** would be inserted into term\_source table under the source of caArray, it is necessary to check if the source of \*caArray\* has already been added in term\_source table. During the normal caArray installation, caArray should have been added to the term\_source table with a default ID =4. To verify, apply the select statement below:

```
select * from term_source where name='caArray'
```

If **caArray** is not a valid source in the term\_source table, it should be added with the insert statement below:

```
insert into term_source (name, version, url) values ('caArray', '2.0', 'https://array.nci.nih.gov')
```

## Add a new terminology term

After the source **caArray** has been added, use the SELECT statement above to find the ID for the source **caArray** just created in the **term\_source** table. The id will be used in the insert statement below to add the new terminology **reverse\_phase\_protein\_array\_features** to the term table:

```
insert into term (value, source) values('reverse_phase_protein_array_features', <id of term_source inserted above>);
```

Use the SELECT statement below to find the ID for the term just created:

```
SELECT ID from term where value = 'reverse_phase_protein_array_features';
```

Use this ID in the INSERT statement below:

```
insert into term_categories (term_id, category_id) values (<id of term inserted above>, 224);
```

## Create a dummy Array Design File for RPLA

Create MAGE-TAB ADF dummy array design files (.ADF files) for the array provider **Zeptosens**, created in Step 1. The dummy file can be a empty file with a simple header.

## Upload the Dummy Array Design file in caArray

Upload new array designs for each ADF using the caArray web application (For more details, please refer to [Using caArray: Uploading Array Design File](#)). The new file will have the final status of "Imported no Parsed".

## Update Array\_Design Table

After uploading dummy array designs into caArray, update their array\_design table entries in the database to lsid\_authority to be Zeptosens.com, lsid\_namespace to be PhysicalArrayDesign. This SQL statement will work only after the array design file is added to caArray.

```
update array_design set lsid_authority = 'Zeptosens.com',  
lsid_namespace = 'PhysicalArrayDesign' where provider = <id of contact inserted above>;
```

## Create New File Types for RPLA data files

The next step is to create new file types for the RPLA data file formats you have. This modification is not at the database level, but file level. The file is:

```
caarray_src_2_3_1_1/caarray.war/src/main/java/ApplicationResources.properties
```

Following is an example the RPLA data file in text format:

```
experiment.files.filetype.Zeptosens_TXT = Zeptosens TXT
```

After the properties is edited, caArray must be rebuilt and redeployed.

## Have a comment?

Please leave your comment in the [caArray End User Forum](#).