HP OpenView Configuration Server Using Radia

for the Windows operating system

Software Version: 4.1

Database Upgrade Procedures Guide

June 2005

Legal Notices

Warranty

Hewlett-Packard makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company
United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Copyright Notices

© Copyright 2005 Hewlett-Packard Development Company, L.P.

No part of this document may be copied, reproduced, or translated into another language without the prior written consent of Hewlett-Packard Company. The information contained in this material is subject to change without notice.

Trademark Notices

Linux is a registered trademark of Linus Torvalds.

Microsoft®, Windows®, and Windows® XP are U.S. registered trademarks of Microsoft Corporation.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Acknowledgements

PREBOOT EXECUTION ENVIRONMENT (PXE) SERVER
Copyright © 1996-1999 Intel Corporation.

Support

Please visit the HP OpenView web site at:

http://www.managementsoftware.hp.com/

This web site provides contact information and details about the products, services, and support that HP OpenView offers.

You can also go directly to the support web site at:

http://support.openview.hp.com/

HP OpenView online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valuable support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track progress on support cases
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and log in. Many also require a support contract.

To find more information about access levels, go to:

http://support.openview.hp.com/access_level.jsp

To register for an HP Passport ID, go to:

https://passport2.hp.com/hpp/newuser.do

Contents

1 Upgrading the Radia Database

 Be sure to back up your existing database before beginning the upgrade process.

Before you upgrade your Radia Database, it is strongly recommended that you use the Verify Database Utility— a non-invasive database analysis tool—to check the status and integrity of the elements in the Database. In general, it is recommended that you periodically verify the status of your Database and perform any necessary maintenance or cleanup.

Once your existing Database has been verified and you have performed any recommended maintenance, use the Database Difference Utility to compare your existing Radia Database to the new Radia Database. This utility creates a file that contains the verbs that will be used to update your Database. You should then review this file and specify what changes to include or exclude. When finished reviewing the file, complete the several procedures which are included in this chapter to create the new Database.

System Requirements for the Radia Database

- Windows NT 4.0 Server with Service Pack 6a, or

- Windows 2000 Server, Server 2003, and XP.

- You must have free space that is equal to two times the size of your Radia Database.

Using the Verify Database Utility

When using the Verify Database Utility to check the status and integrity of the elements in the Database, you will:

- Create a backup copy of your current Radia Database.

- Run the Verify Database Utility to verify the contents of your Database.

- Review the resulting log and perform the recommended maintenance.

To use the Verify Database Utility

1 Stop the Radia Configuration Server service.

2 Create a new directory, such as **MyDatabase**, that you will use to store a backup of your Radia Database and a copy of the Verify Database Utility.

3 Copy your Radia Configuration Server database directory to the newly created **MyDatabase** directory**.**
Default database location is
SystemDrive:\Radia\ConfigurationServer\DB

4 Create another directory in **MyDatabase** to store the Verify Database Utility, such as **UpdateDB** .

Your directory structure should appear similar to the following figure.
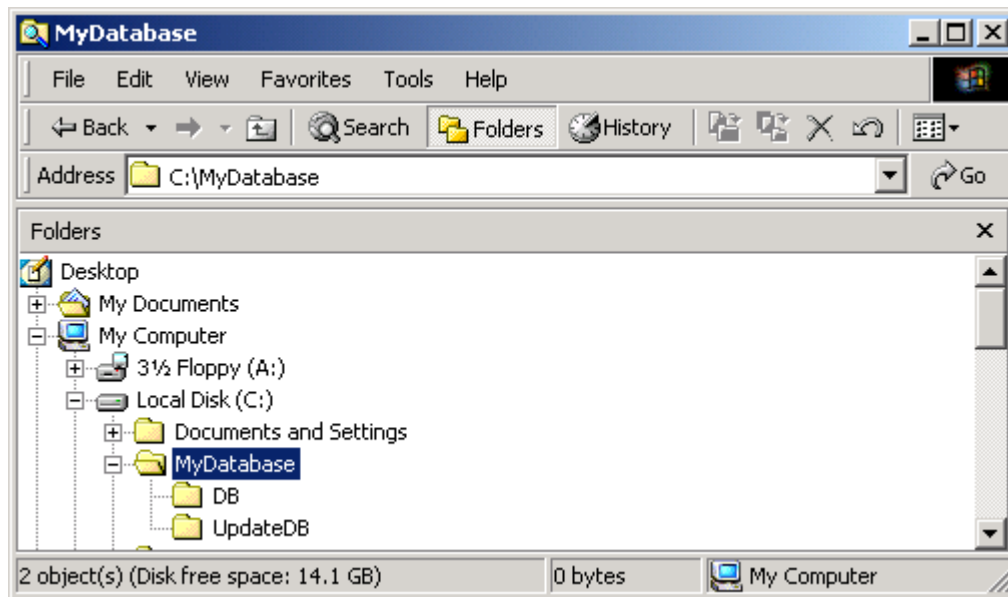
5 Copy your **EDMPROF** file to the **UpdateDB** directory.

6 Extract the contents of the Verify Database Utility zip file (**win32_ams_vrfydb.zip** located withint eh migrate_db folder) to the **UpdateDB** directory.

   This zip file contains:

   — ZEDMAMS.exe

   — Seven DLLs

7 Use a text editor and open the **EDMPROF** file in the **UpdateDB** directory.

8 Go to the **[MGR_DIRECTORIES]** section and use the **DBPATH** argument to specify the path to the backup of your Radia Database.

   ```
   [MGR_DIRECTORIES]

   DBPATH = C:\MyDatabase\DB

   REXX_PATH = C:\Radia\ConfigurationServer\rexx

   METHOD_PATH = C:\Radia\ConfigurationServer\bin
   ```

9 Save and close the file.

10 Open a command prompt and change to the directory where you stored the Verify Database Utility (such as C:\MyDatabase\UpdateDB).

11 To ensure the BEHAVIOR class follows standard naming conventions, run the following command against the database you will be migrating (such as c:\MyDatabase\bin):

```
ZEDMAMS
VERB=EDIT_CLASS_PREFIX,FILE=PRIMARY,DOMAIN=SOFTWARE,CLASS=BEHAV
IOR,PREVIEW=NO,KEEPDATE=YES,FIELD=CLASTYPE,VALUE=T
```

12 Run the Verify Database Utility. The following is a sample command that you might use to verify the integrity of all of the domains in the Database:

```
ZEDMAMS
VERB=VERIFY_DATABASE,DOMAIN=ALL,LOGFILE=VerifyDb.log,DEPTH=
RESOURCE,ZFILE=VERIFYZFILE.BAT
```

Table 1: **Parameters for Verify Database Utility**

Parameter	Explanation
VERB	Specifies the verb that you are running. In this case, you must use VERIFY_DATABASE.
DOMAIN	Specifies the domain that you want to verify. DOMAIN=ALL Verifies all domains, except for the PROFILE domain, which cannot be verified using this utility. DOMAIN=*DomainName* Verifies the domain that you specify, such as SOFTWARE.
LOGFILE	An optional parameter that can be used to specify the name of the log that contains information about the status of your Database. By default, the log file is ZEDMAMS.log. This log is stored in the same directory as the Verify Database Utility.
DEPTH	Use the depth parameter to restrict the verification in the database. The following are acceptable values: resource instance class domain Usage: depth=(Domain/class/instance/resource)

Parameter	Explanation
ZFILE	Use the zfile parameter to create a batch file with all of the verbs reported in the veify_database log to fix the inconsistencies in the database. Output can be executed using the following command. (Note: ZFILE is case sensitive): ZEDMAMS ZFILE (FILENAME) Type a fully qualified path and filename to designate the batch file.

The Verify Database Utility may run for several minutes or several hours depending on the size of your Database.

13 Open the log (by default, ZEDMAMS.log) to review information about the state of your Radia Database. For more information about any of the verbs in the recommended solutions see HP's Technical Support site.

14 Review the newly created `ZFILE` batch file (created when you ran ZEDMAMS with the ZFILE parameter in step 12).

While reviewing output, note that two classes SVR4DEP and SDDEP will be removed. Be sure to allow these classes to be removed from your database by leaving those lines uncommented.

15 Review output.dat.

16 When you are satisfied, run the file against the database by typing: `ZEDMAMS ZFILE "output.dat"` and pressing Enter.

17 After performing the recommended solutions, run the Verify Database Utility again and then review your ZEDMAMS.LOG file to ensure that the issues have been resolved. Messages containing **Improper Instance Name** in the ZEDMAMS log file can be ignored. A sample message follows:

```
<18:02:04:308>   Improper Instance Name <SAMPLE_INSTALL_REX> in
Component Class <PRIMARY.SOFTWARE.BEHAVIOR>
```

 Run the Verify Database Utility as many times as needed to assist you in determining the health of your database.

If you continue to package and publish to your production Database, be sure that create another backup copy and run the Verify Database Utility again immediately before performing the upgrade. This ensures that all potential issues are analyzed.

After you start the Radia Configuration Server with the migrated database, the Configuration Server log file may contain many duplicate object IDs within the Base Instance. These messages can be ignored. An example message follows:

```
ERROR, DUPLICATE OBJECT ID <DABCB40992AB> FOUND WHILE ADDING
<PRIMARY.SOFTWARE.BEHAVIOR._BASE_INSTANCE_>
```

Using the Database Difference Utility

Use the Database Difference Utility to compare your existing Radia Database to the new Radia Database. You will need to do the following:

- Create export decks of your existing Radia Database.

- Run the Database Difference Utility to compare your export decks (created from your backup Database) against the new export decks.

- Review and edit the resulting file, which contains the verbs that will be used to update your Database.

Creating Export Decks of Your Existing Database

Create the following export decks from your existing Database:

- **Radia Database Classes (.XPC)**
  Used with the Database Difference Utility to compare your existing Radia Database against the new database. Later, this is the source class deck that is imported into the new Database.

- **Radia Database Instances (.XPI)**
  This is the source instance deck that is imported into the new Database.

- **Radia Database Resources (.XPR)**
  This is the source resource deck that is imported into the new Database.

Use the export decks as a:

- Base for your comparison against the new Radia Database.
- Back-up file of your current Radia Database.

To create an export deck of your current Radia Database

1   Create a new directory, such as **ExpDeck.**

2   If necessary, stop the Radia Configuration Server service.

3   Open a command prompt.

4   Change the directory to where you stored the Verify Database Utility, which contains the latest ZEDMAMS.EXE, such as UpdateDB.

5   Create three export decks—one deck for the classes, one deck for the resources and one deck for the instances— in the directory that you created.

Each time you run ZEDMAMS, a new log file is created (by default ZEDMAMS.log). If you would like to save the individual log files from each ZEDMAMS session, either rename or move the default log file after you run ZEDMAMS, or use the ZEDMAMS logfile parameter and specify a new log file name.

— For example, to create an export deck for your classes named **oldexport.xpc**, type:
```
ZEDMAMS
VERB=EXPORT_CLASS,FILE=PRIMARY,PREVIEW=NO,OUTPUT=C:\ExpDeck
\OLDEXPORT.XPC
```

Review the ZEDMAMS.LOG (located in the directory where you stored the Verify Database Utility, such as /work/MyDatabase/UpdateDB) to ensure that there were no errors, and that your export completed with RC 0.

— For example, to create an export deck for your instances named **oldexport.xpi**, type:
```
ZEDMAMS
```

```
VERB=EXPORT_INSTANCE,FILE=PRIMARY,PREVIEW=NO,OUTPUT=
C:\ExpDeck\OLDEXPORT.XPI
```

> Review the ZEDMAMS.LOG LOG (located in the directory where
> you stored the Verify Database Utility, such as
> C:\MyDatabase\UpdateDB) to ensure that there were no errors,
> and that your export completed with RC 0.

— For example, to create an export deck for your resources named
oldexport.xpr, type:
```
ZEDMAMS
VERB=EXPORT_RESOURCE,FILE=PRIMARY,PREVIEW=NO,OUTPUT=
C:\ExpDeck\OLDEXPORT.XPR
```

 You may receive a warning that says "no resource found". These
warnings are reported on the instances, which have no resources
(non data-bearing instances).

> Review the ZEDMAMS.LOG LOG (located in the directory where
> you stored the Verify Database Utility, such as
> C:\MyDatabase\UpdateDB) to ensure that there were no errors,
> and that your export completed with RC 0 or RC 4.

 It may take a long time to create export decks of your entire Radia
Database.

Now, you can use the Database Difference Utility to compare your existing
Radia Database with the new Radia Database.

Comparing Databases

Use the Database Difference Utility (dbdiffc.tcl) to compare the two class
export decks.

To compare the databases

1   Create a new directory, such as **DBDiff**.

2   Copy the following files into this directory.

- — Your .XPC file (such as oldexport.xpc)
- — Supplied .XPC file (CORE.XPC)
- — Database Difference Utility (dbdiffc.tcl unzipped from win32_dbdiff.zip)
- — nvdkit.exe (unzipped from win32_dbdiff.zip)

3 Open a command prompt.

4 Navigate to the directory that you created.

5 Run the Database Difference Utility. The following is an example of the command line that you must run.

```
nvdkit.exe dbdiffc.tcl
CUSTOMER_CLASS_DECK=YOURFILENAME.XPC,DEFAULT_CLASS_DECK=FILENAM
E.XPC,BATCH_OUT=OUTPUT.DAT,DELETE=Y
```

Table 2: **Required arguments for the Database Difference Utility**

Argument	
CUSTOMER_CLASS_DECK	The filename of the export deck that you created from your existing Database.
DEFAULT_CLASS_DECK	The filename of the supplied export deck.
BATCH_OUT	The name of the file that contains a list of the verbs to be executed after you review the file. Several text files will be created in the same directory where this file is stored. These files represent the new classes that will be added to the Database.
DELETE	If a field or class exists in the Customer XPC file, but does not exist in the Novadigm XPC file: Set DELETE=Y to delete the fields or classes. Set DELETE=N to keep the fields or classes.

6   Use a text editor to open the output file (specified in the BATCH_OUT argument), such as output.dat.

7   Review this file and determine what changes you would like to keep. If you want to comment out any changes, place an asterisk (*) in front of the verbs that you do not want to run.

While reviewing output, note that two classes SVR4DEP and SDDEP will be removed. Be sure to allow these classes to be removed from your database by leaving those lines uncommented.

Below is a table that defines some of the most common ZEDMAMS verbs that you will encounter when reviewing the output file.

Table 3: Common Verbs in the Output File

Verb	Description
ADD_FIELD	Adds a variable at the end of a template. This verb has a KEEPDATE= option which allows for an update of a variable without updating the ZOBJDATE/ZOBJTIME of the Base_Instance.
CHANGE_FIELDNAME	Changes variable names in the class template. This verb has a KEEPDATE= option which allows for an update of the class without updating the ZOBJDATE/ZOBJTIME instance.
CHANGE_FLD_VALUE	Changes a template's variable length, type Configuration Server, and client flags. This verb has a KEEPDATE= option which allows for the value to be updated without updating the ZOBJDATE/ZOBJTIME instance.

Verb	Description
CHANGE_INS_FIELD	Changes one field in each instance of a class and verifies connects. This verb has a KEEPDATE= option which allows for the field to be updated without updating the ZOBJDATE/ZOBJTIME instance.
CHANGE_INST_DATA	Globally changes data in instance records by class. This verb has a KEEPDATE= option which allows for the instance to be updated without updating the ZOBJDATE/ZOBJTIME instance.
COPY_FIELD	Copies attribute data to a new attribute or to an existing attribute. This verb has a KEEPDATE= option which allows for the field to be updated without updating the ZOBJDATE/ZOBJTIME instance.
DELETE_FIELD	Deletes an attribute from a template. This value does not alter the ZOBJDATE/ZOBJTIME instance.
DELETE_CLASS	Deletes a class from the database.

8   Save and close the file.

Once you have determined the differences between the base Radia Database and your current Database, you are ready to update your existing Radia Database.

Creating a New Database Directory

In this section, you will create a new directory for your new Database.

1 Navigate to your Radia Configuration Server directory.
   Default location, *SystemDrive*:\Radia\ConfigurationServer

2 Create a new directory, such as
   SystemDrive:\Radia\ConfigurationServer\New_DB.

3 Within this new directory, create two additional directories:

   SystemDrive:\Radia\ConfigurationServer\New_DB\PRIMARY

   SystemDrive:\Radia\ConfigurationServer\New_DB\RESOURCE

 These directories must be named PRIMARY and RESOURCE and the names must be capitalized.

4 Go to your previously created verify database directory (UpdateDB).

5 Backup your **EDMPROF.DAT** file.

6 Use a text editor to open **EDMPROF.DAT**.

7 Find the argument **DBPATH** in the [MGR_DIRECTORIES] section and change the value to point to the new Database directory that you created.

   ```
   [MGR_DIRECTORIES]

   DBPATH = C:\Radia\ConfigurationServer\New_DB

   REXX_PATH = C:\ Radia\ConfigurationServer\rexx

   METHOD_PATH = C:\ Radia\ConfigurationServer\bin
   ```

8 Save and close the file.

Importing Your Class Deck

In this section, you will import your source class deck (XPC) into the new Database. This provides a starting point for your new Database.

1 If you want, copy your source class deck (XPC), such as C:\ExpDeck\oldexport.xpc, into C:\MyDatabase\UpdateDB.

2 Open a command prompt.

3   Change the directory to C:\MyDatabase\UpdateDB.

4   Type `ZEDMAMS`
    `VERB=IMPORT_CLASS,FILE=C:\ExpDeck\OLDEXPORT.XPC,PREVIEW=NO`

    The filename used in this example represents the name of the export deck
    that you created earlier. If you did not copy this file, be sure to include
    the fully qualified path in the command line.

5   Press ENTER.

6   Review the ZEDMAMS.LOG to ensure that there were no errors, and
    that your import completed with RC 0.

7   Close the command prompt.

Running the Database Difference File

In this section, you will run the Database Difference file that you created
when you ran the Database Difference Utility.

   Be sure that you have reviewed this file thoroughly and commented
    out any verbs that you do not want to be executed.

To run the text input file

1   Go to the directory where your BATCH_OUT file (such as
    C:\DBDiff\output.dat) is stored. Remember, this directory will also
    contain several files that represent the new classes that will be added to
    the Database.

2   Copy **output.dat** to the UpdateDB directory.

   You do not need to copy the text files to the BIN directory. The
    output.dat file contains a fully qualified path for these files.

3   Open a command prompt.

4   Change the UpdateDB directory and type:

    `ZEDMAMS ZFILE "output.dat"`

where *output.dat* is the name of the file that you specified in the BATCH_OUT parameter when you ran the Database Difference Utility.

 ZFILE must be in all caps

5 Press ENTER.

6 Close the command prompt.

7 Review the ZEDMAMS.LOG to ensure that there were no errors, and that your import completed with RC 0.

Importing the Instances and Resources to the Database

Finally, you must import the remaining source export decks that you created earlier, to finish building the Database.

To import the instance and resource decks

1 Copy the .XPI and .XPR files from the **ExpDeck** directory into the **UpdateDB** directory.

2 Open a command prompt.

3 Change the directory to UpdateDB and type:
```
ZEDMAMS VERB=IMPORT_INSTANCE,FILE=
C:\ExpDeck\OLDEXPORT.XPI,TIME=OLD,PREVIEW=NO
```

 Each time you run ZEDMAMS, a new log file is created (by default ZEDMAMS.log). If you would like to save the individual log files from each ZEDMAMS session, either rename or move the default log file after you run ZEDMAMS, or use the ZEDMAMS logfile parameter and specify a new log file name.

The filename used in this example represents the name of the export deck that you created earlier.

4 Press ENTER.

5 Review the ZEDMAMS.LOG to ensure that there were no errors, and
   that your import completed with RC 0.

6 When the instances have been imported, type `ZEDMAMS`
   `VERB=IMPORT_RESOURCE,FILE= C:\ExpDeck\OLDEXPORT.XPR,PREVIEW=NO`

   The filename used in this example represents the name of the export deck
   that you created earlier.

7 Press ENTER.

8 Review the ZEDMAMS.LOG to ensure that there were no errors, and
   that your import completed with RC 0.

9 Run the Verify Database utility and make sure no errors are reported in
   the log file other than Improper Instance Name errors, which can be
   ignored).

10 Close the command prompt.

11 Copy your edmprof.dat file from the UpdateDB directory and place it in
   your new database bin directory.

12 Start the Radia Configuration Server.

13 If you want to confirm that the upgrade completed successfully, perform a
   Radia Client Connect (e.g., open the Radia Software Manager catalog). If
   you receive the same results as you did prior to the upgrade, then the
   upgrade was successful.

Your Radia Database upgrade is now complete.

If you currently use Radia Distributed Configuration Servers and
have recently upgraded your Radia Database, the first time you
synchronize your servers you must use CRC differencing. The
default synchronization method (Control Differencing) is not
sufficient. For instructions on how to change the differencing
method, refer to the section, Differencing Techniques, in the Release
Notes and Addendum for EDM DMA v4.04, available on the
Novadigm Technical support Web site.

 The connection to the client computer will fail if any object in storage is larger than 6k bytes. A Radia client cannot receive a heap size larger than 6k bytes (or 247 attributes). This does note include the README field, since the README field is not included in the transfer.

Use the Radia System Explorer to view actual instance sizes. Note that the object size may grow if more variables are added to the object.

To avoid a client connection failure during data transfer:
Turn on the **STATS** [MGR_TRACE] option to print the size of the persistent object. The Radia Configuration Server reports the object-size when it is deleted. Monitor this for any object that may grow larger than the current limit of 6k bytes.

Additionally, the **VAR** trace option will display variable sizes that are being added to an object.

For more information, refer to the *Radia Configuration Server Guide*.

Radia 4.1 Database Changes

AUDIT Domain

- AUDIT.CMETHOD.ZREPAIR updated value to "hide nvdkit method"
- AUDIT.FILE added new attribute ZMD5INFO
- Added new class, UNIXPERM
- AUDIT.ZSERVICE added new attribute ZSVCRASH

PRDMAINT Domain

- PRDMAINT.PATH.DIR value changed to &(ZMASTER.ZROOTDIR)_MAINT_\
- PRDMAINT.ZSERVICE added new attribute ZSVCRASH

POLICY Domain

- POLICY.USER Class - ZVERDT class instance– changed supported values from Y/D/R/I to Y/N.
- Updated class description Country / Region (COUNTRY)
- POLICY.MANUFACT added new class
- POLICY.ROLE added new class
- POLICY.SUBNET added new class
- POLICY.MODEL added new class

SOFTWARE Domain

- SOFTWARE.RPM added new attributes RELYONDB, PKGEPOCH, CONTENTS, INSTOPTS, VRFYOPTS
- Removed class SDDEP

- SOFTWARE.SD added new attributes RELYONDB, CONTENTS, PREREQ, COREQ, EXREQ

- SOFTWARE.SOLPATCH added attributes RELYONDB, CONTENTS, CLIUSTER

- SOFTWARE.AIX added enw attribute RELYONDB

- Removed class SVR4DEP

- Added new class WTSACS

- Added new class WTSCTRL

- SOFTWARE.SVR4 added new attributes RELYONDB, ADMIN, ADMINOBJ, AUTOBOOT, PKGVER, PKGREV, CONTENTS, PREREQ, INCOMP, REVERSE

SYSTEM Domain

- SYSTEM.ZINTENT added new class

- SYSTEM.ZPRODUCT added new class

- SYSTEM.DBVER updated DBVER value to 4.1

PATCHMGR Domain

- BULLETIN class, added new attributes REVISION, RUNMODE, HPPOSTED, HPREVISED, ALLDATA, RATING

- DIALOG class, updated attibute value DOMVER to 2.1

- HPFSET added new class

- LOCATION class, ZSTOP000 attribute - removed default value

- LOCATION class added new attributes ZRSCRSTR, ZCREATE, ZDELETE

- METADATA added new class

- PACKAGE class added new attributes PKGURL, NOTES

- PASTORE class, added new attributes RVENDORS, AVENDORS, SYNC, RATING, REBOOT_U, PATCHNUM, CPU, DELORDER, INSORDER

- PASTORE class, removed attribute NEVER

- PG2PR added new class

- PRODUCT class added new attribute VENDOR

- PROGROUP added new class

- PUBERROR class added new attribute VENDOR

- RELEASE class, added new attributes PARENTC, PARENTI, CHILDC, CHILDI, RELEASE, SYNC, RUNMODE

- RELEASE class, removed attributes DISC_RE, MANAGE

OS Domain

- Added new domain, OS

2 Validating the Radia Database

The following errors provide information about structural issues that may be present in your Database.

 Unmated instances are instances that exist with no connection to a resource.

Table 1: Domain Validation

ERROR MESSAGE Impact	Suggested Solution
VERIFY DOMAIN <> CANNOT VALIDATE DMA INSTANCE COUNT. **High** impact for DMA Customers. **Low** impact for non-DMA Customers.	Run: ZEDMAMS VERB=REFRESH_DMA,PREVIEW=NO
VERIFY DOMAIN < > DMA INSTANCE COUNT DOES NOT EQUAL ACTUAL INSTANCE COUNT. **High** impact for DMA Customers. **Low** impact for non-DMA Customers.	Run: ZEDMAMS VERB=REFRESH_DMA,PREVIEW=NO

Table 2: Class Validation

ERROR MESSAGE Impact	Suggested Solution
MISSING TEMPLATE IN CLASS < >. **High** impact for resolution process and integrity of the class.	Delete the class. Restore the class from the last known good template for that class.
CLASS < > HAS MISSING _BASE_INSTANCE. **High** impact for resolution process and integrity of the class.	Delete the class. Restore the class from the last known good template for that class.
CLASS < > HAS INVALID _BASE_INSTANCE. **High** impact for resolution process and integrity of the class.	Delete the class. Restore the class from the last known good template for that class.
VERIFY CLASS < > CANNOT READ DMA CLASS CONTROL INFORMATION. (INVALID CLASS). **High** impact for resolution process and integrity of the class.	Delete the class. Restore the class from the last known good template for that class.
CLASS < > PREFIX SIZE LESS THAN EXPECTED. (INVALID CLASS). **High** impact for resolution process and integrity of the class.	Delete the class. Restore the class from the last known good template for that class.
CLASS < > PREFIX CONTAINS INVALID DATES. **Low** impact. For internal use only.	Delete the class. Restore the class from the last known good template for that class.
CLASS < > PREFIX SIZE < > LESS THAN EXPECTED. **High** impact for resolution process and integrity of the class.	Delete the class. Restore the class from the last known good template for that class.
DATABASE VERSION MISMATCH FOR CLASS < > **High** impact. Corrupt Class.	Delete the class. Restore the class from the last known good template for that class.

ERROR MESSAGE Impact	Suggested Solution
CLASS TEMPLATE FIRST ENTRY NOT FOUND. **High** impact for resolution process and integrity of the class.	Delete the class. Restore the class from the last known good template for that class.
CLASS TEMPLATE < > TEMPLATE FIRST ENTRY NAME < > DOES NOT MATCH CLASS NAME.	Delete the class. Restore the class from the last known good template for that class.
CLASS < > TEMPLATE SIZE MISTMATCH. **High** impact for resolution process and integrity of the class.	Delete the class. Restore the class from the last known good template for that class.
CLASS < > HAS HEAP SIZE < > WHICH IS GREATER THAN SUPPORTED MAXIMUM. **High** impact for resolution process and integrity of the class.	Delete the class. Restore the class from the last known good template for that class.

Table 3: Class Validation

ERROR MESSAGE Impact	Suggested Solutions
CLASS < > HAS MISSING _NULL_INSTANCE. * **Low** impact.	Policy needs to be reviewed to see if the instance is used. Administrator can create a _NULL_ instance in the class if necessary.
CLASS < > HAS INVALID _NULL_INSTANCE.* **High** impact.	Delete the class. Restore the class from the last known good template for that class.
VERIFY CLASS < > INSTANCE COUNT IS ZERO. **High** impact. Affects the System Explorer and resolution process.	Run: ZEDMAMS VERB=REFRESH_DMA,PREVIEW=NO
VERIFY CLASS < > CANNOT VALIDATE DMA INSTANCE COUNT. **High** impact. Affects the System Explorer and resolution process.	Run: ZEDMAMS VERB=REFRESH_DMA,PREVIEW=NO
VERIFY CLASS < > DMA INSTANCE COUNT DOES NOT EQUAL ACTUAL INSTANCE COUNT. **High** impact. Affects the System Explorer and resolution process.	Run: ZEDMAMS VERB=REFRESH_DMA,PREVIEW=NO
INSTANCE SIZE LESS THAN MINIMUM REQUIRED. **High** impact. Affects the System Explorer and resolution process.	Delete the class. Restore the class from the last known good template for that class.

Table 4: Instance Validation

ERROR MESSAGE Impact	Suggested Solution
INSTANCE SIZE DOES NOT MATCH CLASS TEMPLATE. **High** impact. The instance is corrupt.	Delete the class. Restore the class from the last known good template for that class.
ZRSCDATE NOT FOUND BUT RESOURCE FLAG SET IN CONTROL INFO. **No** impact**.**	Delete the class. Restore the class from the last known good template for that class.
WARNING: UNMATED INSTANCE < > DOES NOT HAVE CORRESPONDING DATA. **Medium** impact. Resource is missing for an instance	Determine the impact the resource has on your database. If the resource is non-critical, delete. If the resource is critical, set the ZSERVICE.REMOVAL attribute to A and redeploy. (Setting the REMOVAL attribute to A abandons the resource, relinquishing Radia control.) Note: Deleting a resource may have adverse affects on other Radia components. Before deleting a damaged resource, make sure it will not affect any existing components.

ERROR MESSAGE Impact	Suggested Solution
RESOURCE < > SIZE < > IS INVALID FOR ASSOCIATED INSTANCE < >. **High** impact. Damaged Resource.	Determine the impact the resource has on your database. If the resource is non-critical, delete. If the resource is critical, set the ZSERVICE.REMOVAL attribute to A and redeploy. (Setting the REMOVAL attribute to A abandons the resource, relinquishing Radia control.) Note: Deleting a resource may have adverse affects on other Radia components. Before deleting a damaged resource, make sure it will not affect any existing components.
RESOURCE < > ACTUAL SIZE < > NOT EQUAL TO SIZE < > IN ASSOCIATED INSTANCE. **High** impact. Damaged Resource.	Determine the impact the resource has on your database. If the resource is non-critical, delete. If the resource is critical, set the ZSERVICE.REMOVAL attribute to A and redeploy. (Setting the REMOVAL attribute to A abandons the resource, relinquishing Radia control.) Note: Deleting a resource may have adverse affects on other Radia components. Before deleting a damaged resource, make sure it will not affect any existing components.
ERROR: OPENING RESOURCE < > FOR INSTNACE < >.	Delete the class. Restore the class from the last known good template for that class.
ERROR: UNABLE TO READ PREFIX FOR RESOURCE < > FOR ASSOCIATED INSTANCE < >.	Delete the class. Restore the class from the last known good template for that class.

ERROR MESSAGE Impact	Suggested Solution
RESOURCE < > ACTUAL SIZE < > DOES NOT MATCH EXPECTED SIZE < > FOR ASSOCAITED INSTANCE. **High** impact. Damaged Resource.	Determine the impact the resource has on your database. If the resource is non-critical, delete. If the resource is critical, set the ZSERVICE.REMOVAL attribute to A and redeploy. (Setting the REMOVAL attribute to A abandons the resource, relinquishing Radia control.) Note: Deleting a resource may have adverse affects on other Radia components. Before deleting a damaged resource, make sure it will not affect any existing components.
RESOURCE < > PREFIX CLASS NAME < > DOES NOT MATCH ITS ASSOCIATED INSTANCE < >. **High** impact. Damaged Resource.	Determine the impact the resource has on your database. If the resource is non-critical, delete. If the resource is critical, set the ZSERVICE.REMOVAL attribute to A and redeploy. (Setting the REMOVAL attribute to A abandons the resource, relinquishing Radia control.) Note: Deleting a resource may have adverse affects on other Radia components. Before deleting a damaged resource, make sure it will not affect any existing components.

ERROR MESSAGE Impact	Suggested Solution
RESOURCE < > PREFIX INSTANCE NAME < > DOES NOT MATCH ITS ASSOCIATED NAME < >. **High** impact. Damaged Resource.	Determine the impact the resource has on your database. If the resource is non-critical, delete. If the resource is critical, set the ZSERVICE.REMOVAL attribute to A and redeploy. (Setting the REMOVAL attribute to A abandons the resource, relinquishing Radia control.) Note: Deleting a resource may have adverse affects on other Radia components. Before deleting a damaged resource, make sure it will not affect any existing components.
RESOURCE < > COMPRESSED SIZE < > NOT EQUAL TO SPACE < > IN VARIABLE < > OF ASSOCIATED INSTANCE < >. **Medium** impact. ZCMPSIZE Variable may be compromised.	Delete the class. Restore the class from the last known good template for that class.
RESOURCE < > SIZE < > NOT EQUAL TO SIZE < > IN VARIABLE < > OF ASSOCIATED INSTANCE < >. **Medium** impact. ZRSCSIZE Variable may be compromised.	Delete the class. Restore the class from the last known good template for that class.
RESOURCE < > CRC < > NOT EQUAL TO CRC < > IN VARIABLE < > OF ASSOCIATED INSTANCE < >. **Medium** impact. ZRSCCRC Variable may be compromised.	Delete the class. Restore the class from the last known good template for that class.

Checking for Orphaned Resources

Orphaned resources are resources that exist with no parent instance. The following error provides information about orphaned resources.

Table 5: Orphaned Resource Check

ERROR MESSAGE Impact	Suggested Solution
FOUND ORPHANED RESOURCE – DATA DOES NOT HAVE A CORRESPONDING PARENT INSTANCE < >. THIS RESOURCE WAS PREVIOUSLY PARENTED BY < >. **Low** impact. This resource can be left OR you may choose to remove the unmated resource by using the ZEDMAMS VERB = DELETE_RESOURCE.	Run: ZEDMAMS VERB=DELETE_RESOURCE, PREVIEW=NO

Checking for Duplicate IDs

The following errors provide information about duplicate object IDs that may be present in your Database.

Table 6: Object ID Check

ERROR MESSAGE Impact	Suggested Solution
DUPLICATE OBJECT ID < > WITHIN <COMPONENT> CLASS < > BETWEEN INSTANCE < > AND < >. **High** impact. Possibility of corrupt data.	If either data is corrupted, use: ZEDMAMS VERB = DELETE_INSTANCE, PACKAGE=YES,PREVIEW=NO,DOMAIN=< >, CLASS= < >, INSTANCE =< >
DUPLICATE OBJECT ID < > IN INSTANCES < > AND < >.	If both are from PACKAGE Class: Duplicate Instance object ids in Package Class cannot be fixed. You will need to delete the package. If class is NOT PACKAGE: ZEDMAMS VERB= CHANGE_OBJECTID,FILE=PRIMARY,DOMAIN= <>,CLASS=<>,INSTANCE=<>,PREVIEW =NO
DUPLICATE OBJECT ID < > WITHIN < > CLASS < > BETWEEN INSTANCE < >.	If class is PACKAGE: Duplicate Instance object ids in Package Class cannot be fixed. You will need to delete the package. If class is NOT PACKAGE: ZEDMAMS VERB=CHANGE_OBJECTID,FILE=PRIMARY,DOMAIN= <>,CLASS=<>,INSTANCE=<>,PREVIEW =NO

Index