Your Business ▪ Your Agenda ▪ Our Passion
Maximizing the value of IT together

Dictionary Clean-Up for the Future

Joan Dunn
Senior Technical Instructor
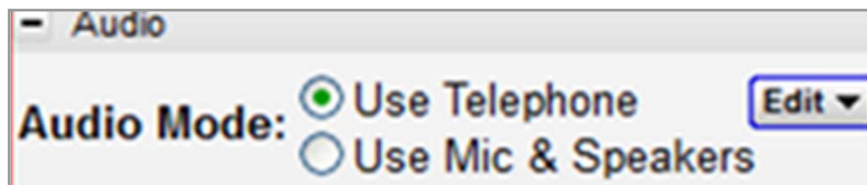
Rocket | Database Servers and Tools

Webinar Opening Procedure

- Orange arrow = control GoTo Webinar control panel

- This webinar will be recorded and posted to our website for you to replay at a later date
  - You will be notified via email once .pdf and video are posted

- If using phone – don't select "Use Mic & Speakers"

- All attendees are muted during the presentation

- Brief Q&A session – after the presentation

Joan Dunn's Biography

- Joined Unidata in 1994 as Support Manager responsible for Vendor Education

- Current role as U2 Senior Technical Instructor

- Responsible for creating, writing, and presenting U2 (UniData and UniVerse) courseware worldwide for U2 Education

- Expertise across the product line including tools such as UniObjects, XAdmin, and U2 Replication

- Over 22 years experience with MultiValue databases

- Over 18 years experience in the education role

Abstract

If you want to use third-party relational tools to connect to U2 or want to get the most out of Rocket's products and third party tools, you will really benefit from well-defined dictionaries.

This session will give you guidance on the best approach to dictionary clean-up and talk about the benefits you will get out of doing it.

Agenda

- Why clean up the dictionary

- Preparing for dictionary cleaning

- What issues can exist in your dictionary

- Common methods for cleaning a dictionary

- Verifying and debugging dictionary issues

Why Clean Up the Dictionary

- Background on dictionary

- Prepare for third party tools
  - Minimize errors
  - Proper data typing
  - Performance

What Issues Can Exist

- Field name with '.'

- Invalid derived fields
  - Self-referencing
  - Invalid syntax
  - Calls to missing subroutines

- Many references to the same data

- Incorrect formatting or justification

Quoted Identifiers – Overcoming the Dot

- Always supported in UniVerse

- UniData
  - SQL Command line since 7.2
  - ODBC/JDBC support in 7.3
  - U2MDM eliminates the need by resolving dictionary names

- Example of quoted identifier:

SELECT "FIRST.NAME", "LAST.NAME" FROM CUSTOMER;

8

Preparing for Dictionary Clean-up

- Fixing in the production account
  - Can impact production
  - May not be able to reverse easily
  - **Not** recommended

- Fixing a copy of the DICT item
  - Keeps production safe
  - Easy to reset back to original
  - **Recommended** way

Preparing for Dictionary Clean-up

- Create a new account (two options)
  - Development/Test
    - Make exact **copy** of the account
    - **OR** get a good **backup** of the account
  - Production copy
    - Create a **new account**
    - **Copy** the **dictionary** files to the account
    - **Create** the **VOC pointers** to point to the production data

Options for Inspecting DICT

- Quickest way to check a dictionary:
  - LISTDICT (UniData) or LIST DICT (UniVerse)
  - Visually check the primary dictionary fields

```
SORT DICT CUSTOMER TYP LOC CONV MNAME FORMAT SM ASSOC BY TYP BY LOC BY @ID
22:54
:58 Apr 29 2012 1
@ID............ TYP LOC.......... CONV MNAME.......... FORMAT SM ASSOC.....

@ID             D             0       Cust            10R    S
NAME            D             1       Customer Name   15T    S
ADDRESS         D             2       Address         20L    M
CITY            D             3       City            10L    S
STATE           D             4       ST              2L     S
ZIP             D             5       Zip Code        5R     S
TAPES_RENTED    D             7       Tapes           10L    MV TAPE_INFO
DATE_OUT        D             8 D2/   Date Out        8L     MV TAPE_INFO
```

Common Methods for Cleaning

- Identify DICT issues with the following:
  - HS.ADMIN / HS.SCRUB
  - CONVERT.SQL
  - COMPILE.DICT

  - U2 Metadata Manager (U2MDM)

HS.ADMIN

- Why use the HS.ADMIN Utility
  - Helps make accounts and files acceptable for relational access
  - Needed for HS.SCRUB

- Accessed by logging to the HS.ADMIN account
  - >LOGTO HS.ADMIN
  - >HS.ADMIN

- Menu-driven tool

- Uses the dictionary to drive the changes

- Can be used to "fix" the dictionary and the data. Caution!

Process for Using HS.ADMIN

- Activate an account for relational access
  - Define account to UV.ACCOUNT

- Activate access to a file
  - HS.SCRUB a file dictionary and possibly data

- Update the File Information Cache
  - HS.UPDATE.FILEINFO

HS.ADMIN Menu

```
UniVerse Server Administration

  1. List activated accounts
  2. Show UniVerse ODBC Config configuration for an account
  3. Activate access to files in an account
  4. Deactivate access to files in an account
  5. Run HS.SCRUB on a File/Table
  6. Update File Information Cache in an account


Which would you like? ( 1 - 6 )  ?█
```

Activate Access to Files in an Account

```
Which would you like? ( 1 - 6 )  ?3
Enter the account name or path, or press Return for a list of accounts in
which file access has already been activated: XDEMO

Welcome to the XDEMO Account.
Activating access to files in account C:\U2\UV\XDEMO
Creating data portion of HS_FILE_ACCESS file.
Creating file "HS_FILE_ACCESS" as Type 30.
Creating dict portion of HS_FILE_ACCESS file.
Creating file "D_HS_FILE_ACCESS" as Type 3, Modulo 1, Separation 2.
Added "@ID", the default record for RetrieVe, to "D_HS_FILE_ACCESS".
Generating file information cache.
Updating UV.ACCOUNT file.
The following record or records in UV.ACCOUNT now have their HS_FILE_ACCESS fiel
d set to YES:
    XDEMO

Activation completed.
```

Error Messages

- CATEGORIES.F's dictionary has **no @ phrase or @SELECT phrase**. "CATEGORIES.F" was written into File Information Cache but may not behave as desired.

- **No columns of association STOCK in file CONCESSIONS.F are visible** (i.e., in @ phrase or @SELECT phrase). "CONCESSIONS.F_STOCK" was not written into File Information Cache.

- **Could not do SELECT * FROM ORDERITEMS.F**. STATUS = -1, SQLSTATE = S1000, CODE = 950220 UniVerse/SQL: Invalid association operation: Column or association name "Detail" does not exist in DICT of file "ORDERITEMS.F". "ORDERITEMS.F" was not written into File Information Cache.

- **Could not open file MYTEST**. STATUS() = -2 "MYTEST" was not written into File Information Cache.

HS.SCRUB – Do Not Fix From Here

```
UniVerse Server Administration

 1. List activated accounts
 2. Show UniVerse ODBC Config configuration for an account
 3. Activate access to files in an account
 4. Deactivate access to files in an account
 5. Run HS.SCRUB on a File/Table
 6. Update File Information Cache in an account

Which would you like? ( 1 - 6 )  ?5
Enter the account name or path, or press Return for a list of accounts in
which file access has been activated: XDEMO


Welcome to the XDEMO Account.
Enter the name of the file or table to scrub: PRODUCTS


HS.SCRUB will generate a report, and can optionally adjust the data and
dictionary.  Please enter the mode of operation as one of the following:
<Return>=report only, FIX=adjust data and dictionary.  Mode:
```

HS.SCRUB Syntax

- Preferred method – run HS.SCRUB in the desired account

- **HS.SCRUB** *filename* [**FIX | AUTOFIX** [**DICT | DATA**]]
  - **No parameters** provides a **report** of anomalies
  - **FIX** provides an **interactive session** where all anomalies for data and dictionary are presented for resolution
  - **AUTOFIX fixes all** anomalies (**no interaction**), but can be directed to just the DATA or DICT with the specific keyword – Caution!

HS.SCRUB Messages

- ACT.NO was analyzed using INT for a data type

- DESCRIPTION was analyzed using VARCHAR(254) for a data type

- ASPECT was analyzed using INT for a data type; **Suggested SQL data type: FLOAT(3)**

- AWARD_NAME was analyzed using CHAR(20) for a data type;   *** field contains 8 item(s) that have subvalues * Character field contains 8 item(s) that exceed the field's specified width**

HS.SCRUB for Cleaning Data

- **Caution!**

- Adjusts dictionary entries to accommodate bad data

- Fixes the data in the file, optionally saving a copy of the original file

HS.UPDATE.FILEINFO

- Command or Option 6 from the HS.ADMIN menu

- Refreshes the **File Information Cache** following changes to an account
  - Adding, changing, or deleting F- or Q-pointers in the VOC
  - Creating or deleting UniVerse files
  - Defining, changing, or deleting association definitions in file dictionaries
  - Adding or deleting unassociated multivalued fields to or from a file

CONVERT.SQL

- **Checks DICTIONARY**
  - Checks for name compliance
  - Checks for format and S/M values
  - Creates synonyms if needed
  - Adds the file to the "privilege" file

- **CHECKONLY option**

- **NONAMECONV option**

Phrase Associations

- CONVERT.SQL to identify issues:
  - It will find missing PH-type items
  - An ASSOC name is in the dictionary item but there is not a Phrase item in the dictionary

- CONVERT.SQL will not identify:
  - Items in the PH-type item that are not in the dictionary
  - Items without association

COMPILE.DICT

- Check virtual dictionary/I-descriptors
  - Syntax verification
  - Check for self-referencing

- Syntax: COMPILE.DICT filename [id]

Verifying and Debugging Dictionary Issues

- COMPILE.DICT shows errors
  - Self referencing items
  - Incomplete/invalid syntax
  - Invalid parameters in TRANS (UV only)

- COMPILE.DICT shows no errors
  - Invalid number of parameters
  - Subroutine called does not exist
  - Invalid parameters in TRANS (UD only)

Self-Referencing DICT Items

- **UniData Example:**
  :COMPILE.DICT BADDICT
  SELF=SELF
  Virtual Attribute Error: SELF is self-referential.
  Virtual field SELF has syntax error.

- **UniVerse Example**
  >COMPILE.DICT BADDICT
  Compiling "SELF".
  Circular definition
  SELF syntax error

Incomplete DICT Items

- **UniData Example:**
  :COMPILE.DICT BADDICT
  INCOMP=FIELD(@ID,
  FIELD(@ID,
  --------^
  Virtual Attribute Error: INCOMP syntax error
  Virtual field INCOMP has syntax error.

- **UniVerse Example**
  Compiling "INCOMP".
  FIELD (@ID , syntax error

Valid DICT Items?

- COMPILE.DICT does not find all issues

- Syntactically correct may not be valid
  - Verify with LIST or SORT
  - Verify with JDBC Select

Valid DICT Items?

- Example of Syntactically correct

@ID...........	TYP	LOC..........	CONV	MNAME..........	FORMAT	SM
MYSUB	I	SUBR("MYSUB", @ID, "EXTRA PARAM")		BAD	10L	S
BADPROG	I	SUBR("DOES.NOT.EXIST", @ID)		INVALID	10L	S
TRANS	I	TRANS("BADFILE",@ID,"BAD", "X")		BAD	10L	S

Validate with LIST/SORT

- **Example of LIST/SORT error message**
  - MYSUB

    In C:\U2\ud73\demo\CTLG\MYSUB at line 1 Error, **SUBROUTINE 'MYSUB' has different number of arguments than its CALL statement**.

  - BADPROG

    'DOES.NOT.EXIST' is not cataloged.

    Virtual Attribute Error: DOES.NOT.EXIST Indirect call error.

  - TRANS

    'BADFILE' does not exist in VOC file.

Validate with JDBC

- UCC-17 - How to run a SQL SELECT statement in a GUI interface using the U2 JDBC driver without installing the SUN JDK or JRE
  - Download the Zip file
    - It contains setup.exe
    - Installs runjdbcxselectp
    - Program accessible from the Rocket Software menu
    - Connects to both UniData and UniVerse

U2MDM - Features of the First Release

- Generation of 1NF mapping and the corresponding relational schema with ease

- Built-in schema testing tool

- Graphical representation of file metadata and relational schema

- Repository of unique metadata definitions for each selected file

- Gradual upgrade from older tools and existing environments

- Schema deployment

U2MDM and U2 Releases

- First release:
  - UniData 7.3 and
  - March release of DBTools

- Work in progress for support with UniVerse

Summary

- In this session we covered
  - Why clean up the dictionary
  - Preparing for dictionary cleaning
  - What issues can exist in your dictionary
  - Common methods for cleaning a dictionary
  - Verifying and debugging dictionary issues

Question and Answer Session

To ask a question:
- Click on hand icon with green arrow and we will call your name

Or you may email us your question later at:

U2Training@rocketsoftware.com

Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

IN ADDITION, THIS INFORMATION IS BASED ON ROCKET SOFTWARE'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY ROCKET SOFTWAREWITHOUT NOTICE.

ROCKET SOFTWARE SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANTY OR REPRESENTATION FROM ROCKET SOFTWARE(OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR

- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF ROCKET SOFTWARE.

Trademarks and Acknowledgements

The following are trademarks or registered trademarks of Rocket Software, Inc.: Dynamic Connect, SystemBuilder, U2, U2 Web Development Environment, UniData, UniVerse, and wIntegrate.

IBM, the IBM logo, AIX, and DB2 are trademarks of IBM in the United States and other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Microsoft, SQL Server, Windows, and Excel are trademarks of the Microsoft group of companies.

UNIX is a registered trademark of The Open Group.

Other company, product, and service names mentioned herein may be trademarks or service marks of others.

Thank you!