z/OS 2.4 IBM Education Assistant (IEA)

Solution (Epic) Name: Custom Fields for General Resource and Dataset Profiles

Element(s)/Component(s): RACF

IBM LinuxONE™

Agenda

- Trademarks
- Session Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Migration & Coexistence Considerations
- Installation
- Session Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.
- Additional Trademarks:
  - None

Session Objectives

- Describe the new functions
  - Custom fields for DATASET and general resource profile
  - R_admin/IRRXUTIL support for DATASET profiles
  - IRRXUTIL support for Class Descriptor Table (CDT) entries

Overview

- Who (Audience)
  - Security administrators, application programmers
- What (Solution)
  - Ability to define your own RACF profile fields for DATASET and general resource profiles
  - Ability to retrieve DATASET profile contents in a structured manner from an unauthorized assembler program or rexx exec
  - Ability to retrieve RACF class definition attributes and SETROPTS settings for any class, IBM or client-defined (static or dynamic)
- Wow (Benefit / Value, Need Addressed)
  - Complete and consistent solution for modifying the RACF ‘schema’
    - With ability to perform validation in a rexx exec!
  - Complete and consistent solution for retrieving RACF profile and configuration information

Overview … Custom Fields

- Custom fields can be added to general resource and data set profile the same way you're familiar with for users and groups
  - *Profile-type*.CSDATA.*field-name* with a CFDEF segment
  - Smart ISPF panels for CSDATA (extracts and displays current field names/values)
  - Dynamic assembler exit IRRVAF01 can validate field values as they are assigned to the CSDATA segment
  - IRRDBU00, R_admin, IRRXUTIL support
- NEW! For all profile types, field values can be validated in a system rexx exec!

Usage & Invocation … Custom fields

- Example of defining custom field to a user profile

```
RDEFINE CFIELD USER.CSDATA.USRFIELD CFDEF(TYPE(CHAR) MAXLENGTH(50))
IRRDPI00 UPDATE
ALTUSER JOE CSDATA(USRFIELD(BlahBlahBlah))
```

- Example of defining custom field to a group profile

```
RDEFINE CFIELD GROUP.CSDATA.GRPFIELD CFDEF(TYPE(CHAR) MAXLENGTH(50))
ALTGROUP SOMEGRP CSDATA(GRPFIELD(BlahBlahBlah))
```

- (new!) Example of defining custom field to a data set profile

```
RDEFINE CFIELD DATASET.CSDATA.DSFIELD CFDEF(TYPE(CHAR) MAXLENGTH(50))
ALTDSD ‘MY.DATASET.PROFILE’ GENERIC CSDATA(DSFIELD(BlahBlahBlah))
```

- (new!) Example of defining custom field to a general resource profile

```
RDEFINE CFIELD GENERAL.CSDATA.GENFIELD CFDEF(TYPE(CHAR) MAXLENGTH(50))
RALTER MYCLASS MY.GENERAL.PROFILE CSDATA(GENFIELD(BlahBlahBlah))
```

Applies to all general resource classes by default, but it’s easy to restrict it …

Usage & Invocation … Custom fields

- System rexx validation
  - Identify name of exec to be used for individual field definition using new VALREXX keyword in the CFDEF segment

    ```
    RALTER CFIELD GENERAL.CSDATA.GENFIELD CFDEF(VALREXX(VALMYFLD))
    ```

  - Put the Rexx exec named VALMYFLD in the system Rexx concatenation before values are assigned to the field
- You could have a single exec for *all* custom fields, a unique exec for *each* field, or anywhere in between
  - And here is where you could restrict the use of certain fields to certain general resource classes
- As soon as the rexx code is stored in system rexx, it is active. No IPL or restart is necessary.

Usage & Invocation … Custom fields

- Parameters passed to the exit
  1. Command operation (ADD or ALTER)
  2. Class name (USER, GROUP, DATASET, or any general resource class)
  3. Profile name (Profile to which the field is being assigned. For DATASET, the name is fully qualified.)
  4. Segment name (CSDATA)
  5. Custom field name (The field name as defined in the CFIELD class)
  6. Custom field type. A single character: C – Character F – Flag (YES or NO)
     H – Hexadecimal N – Numeric
  7. Custom field length (0 for numeric fields)
  8. Custom field value
  9. Generic profile indicator (0 or 1)
  10. Data set volume (For class name DATASET only. For the ADDSD command, this is a blank-separated list of volume names specified on the VOLUME keyword. For the ALTDSD command, this is the single volume name specified on the VOLUME keyword to identify the discrete data set.)

Usage & Invocation … Custom fields

- System rexx arguments can be 512 maximum, but custom fields can be 1100 chars. So,
  - The first 512 chars are passed in the ‘value’ arg
  - An additional stem variable is available to the exit
    - **FullValue.0 contains the number of variables that follow.**
    - **FullValue.1 – FullValue.*n* contain the custom field value split into 512-character-maximum pieces.**

Usage & Invocation … Custom fields

- The command image is provided to the exit in a stem variable only (not as an argument)
    - **OriginalCommand.0 contains the number of variables that follow.**
    - **OriginalCommand.1 – OriginalCommand.*n* contain the command image split into 512-character-maximum pieces.**
- If multiple profile names are specified on a command, the first one is passed to the exit in the ‘profile name’ argument, and the rest are available in the command buffer.
- If the AT or ONLYAT keyword is specified on the command, it does not appear in the command buffer.

Usage & Invocation … Custom fields

- Return codes from the exit
  - 0 - success
  - 4 - Re-prompt the user if in prompt mode
  - 8 - fail
- Optional error message text can be returned to RACF by setting the ExitMessage variable.
  - This text will be appended to the end of IRR52223I when the exit returns with a non-zero value. The length cannot exceed 200 characters. E.g.

```
IRR52223I CSDATA validation failed by exit MYREXVAL. MYFIELD must
start with a digit and be followed by 9 letters.
```

Usage & Invocation … Custom fields

```
/*rexx                                                               */
parse ARG cmd,class,profname,segname,fldname,fldtype,fldlen,fldval,  ,
          dsgen,dsvol
AXRWTO('Dumping input args:' )
AXRWTO('  command name:'                 cmd )
AXRWTO('  class name:'                   class )
AXRWTO('  profile name:'                 profname )
AXRWTO('  segment name:'                 segname )
AXRWTO('  field name:'                   fldname )
AXRWTO('  field type:'                   fldtype )
AXRWTO('  field length:'                 fldlen )
AXRWTO('  field value:'                  fldval )
AXRWTO('  DATASET generic indicator:'    dsgen )
AXRWTO('  DATASET volume:'               dsvol )

do i = 1 to OriginalCommand.0
  AXRWTO('  Command image is: ' OriginalCommand.i)

return 0
```

Sample exit which simply displays the input parameters.

Overview … Dataset Support

- Update existing R_Admin(IRRSEQ00) callable service to support Dataset profile extracts.

- Update IRRXUTIL to support DATASET profiles.

Usage & Invocation .. Dataset Support

- Provide ability to extract DATASET profiles via R_Admin.

Function Code	Value	Description
ADMN_XTR_DS	X’22’	Extracts a data set profile
ADMN_XTR_NEXT_DS	X’23’	Extracts the next data set profile

- Requires READ access to IRR.RADMIN.LISTDSD in the FACILITY class
- Parameter list header (ADMN_PROF_MAP DSECT in IRRPCOMP) updated to include volume name and ‘duplicate discrete data set count’ for extract-next requests
- Works about the way you’d expect it to based on existing function (field tables in Callable Services appendix updated to add output-only fields and other extract-related information)

Usage & Invocation .. XUTIL Dataset Support

```
/* REXX */
  IRRXUTIL_OPTION_PROF_UPPERCASE= 1
  IRRXUTIL_OPTION_NAMEONLY= 1
parse arg name
MYRC=IRRXUTIL("EXTRACT","DATASET",name,"DS",name,"TRUE")
  SAY MYRC
  If myrc = '0 0 0 0 0' Then Do
       SAY "PROFILE CLASS: "DS.CLASS    -> Output: Profile Class: DATASET
       SAY "PROFILE NAME: "DS.PROFILE   -> Output: Profile Name: <name>
  End
```

Overview … CDT Support

- IRRXUTIL now has the ability to retrieve RACF CDT(Class Descriptor Table) attributes into rexx variables.
- IRRXUTIL has been enhanced to support additional IRRXUTIL “environment” variables.

Usage & Invocation .. CDT Support

- Support the ability to retrieve RACF class definition attributes (both static and dynamic) into rexx variables
  - Allows you to get live settings, rather than extracting dynamic class settings from the CDTINFO segment of CDT class profiles
  - Uses a new “_CDT” pseudo-class (like existing _SETROPTS and _RRSFEXTR classes)
  - Supports EXTRACT and EXTRACTN (so you can cycle through classes)
  - Supports concurrent retrieval of SETROPTS settings for the class (so you don’t have to cobble together results of a separate SETROPTS extract call)
    - **Requires READ access to IRR.RADMIN.SETROPTS.LIST in the FACILITY class**
  - No R_admin support
    - **An assembler interface already exists: RACROUTE REQUEST=STAT**

Usage & Invocation .. CDT Support

```
/* REXX */
parse arg name
If name = '' Then name = "FACILITY"
MYRC=IRRXUTIL("EXTRACT","_CDT",name,"CLS","","FALSE")
SAY MYRC
If myrc = '0 0 0 0 0' Then Do
  SAY "Returned class is:" CLS.CLASSNAME
  SAY "Posit number is:"   CLS.POSIT
  SAY "Grouping class?"    CLS.GROUPING_CLASS
  SAY "Member class?"      CLS.MEMBER_CLASS
  SAY "Profiles allowed?"  CLS.PROFILES_ALLOWED
  SAY "Mixed case profiles allowed?"  CLS.MIXED_CASE
  SAY "Key qualifiers:"    CLS.KEY_QUALIFIERS
  SAY "FIRST="             CLS.FIRSTCHAR
  SAY "OTHER="             CLS.OTHERCHAR
  SAY "Default UACC:"      CLS.UACC
  SAY "OPER=YES?"          CLS.OPERATIONS
  …
End
```

Sample exec

Variable names documented in Macros and Interfaces

Usage & Invocation .. CDT Support

- New “Environment” Variables
  - IRRXUTIL_OPTION_NAMEONLY
  - IRRXUTIL_OPTION_ BASEONLY
  - IRRXUTIL_OPTION_ PROF_UPPERCASE
  - IRRXUTIL_OPTION_ CLASS_SETTINGS

Usage & Invocation .. CDT Support

```
/* REXX */
IRRXUTIL_OPTION_CLASS_SETTINGS = 1
parse arg name
If name = '' Then name = "FACILITY"
MYRC=IRRXUTIL("EXTRACT","_CDT",name,"CLS","","FALSE")
SAY MYRC
If myrc = '0 0 0 0 0' Then Do
  …
    SAY "CLASS ACTIVE: "CLS.CLASS_ACTIVE
    SAY "STAT ACTIVE: "CLS.STATISTICS_ACTIVE
    SAY "GENERIC ACTIVE: "CLS.GENERIC_ACTIVE
    SAY "GENCMD ACTIVE: "CLS.GENCMD_ACTIVE
    SAY "GLOBAL ACTIVE: "CLS.GLOBAL_ACTIVE
    SAY "LOGOPTIONS: "CLS.LOGOPTIONS
    SAY "AUDIT ACTIVE: "CLS.AUDIT_ACTIVE
  End
```

Interactions & Dependencies

- To exploit this item, all systems in the Plex must be at the new z/OS level: No
- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - None

Migration & Coexistence Considerations

- None

Installation

- No considerations

Session Summary

- Custom Fields are now applicable to General Resources and Dataset profiles.
- R_Admin/IRRXUTIL has DATASET extract support.
- IRRXUTIL supports CDT extract.
- IRRXUTIL has enhanced “environment variables”

Appendix

- Andrew Rundall (rundalla@us.ibm.com)

- **Security Server RACF Security Administrator's Guide**
  - **Use of custom fields, including new VALREXX capability**
- **Security Server RACF Command Language Reference**
  - **Interface to VALREXX exits**
- **Security Server RACF Callable Services**
  - **DATASET-extract information, field names in an Appendix**
- **Security Server RACF Macros and Interfaces**
  - **IRRXUTIL documentation, including CDT variable names**
- **Security Server RACF Command Language Reference**
  - **CFDEF and CSDATA segment syntax**