

IBM

Using custom fields in the RACF database



@business on demand software

© 2008 IBM Corporation

This is a presentation that will briefly explain what custom fields are and how to implement custom fields.

Objectives

- To understand how to use custom fields
 - ▶ How to define custom fields
 - ▶ How to activate custom fields
 - ▶ How to modify custom fields
 - ▶ How to display custom fields
- Where to go for more information about custom fields

The objective of this presentation is to explain how to use custom fields. You will learn how to define, activate, modify, and display custom fields.

Table of contents

- Overview
- Defining a custom field
- Examples of defining a custom field
- Activating a custom field
- Storing data in a custom field
- Displaying custom field data
- Displaying user custom field data
- Displaying group custom field data
- Controlling who can define custom fields
- Controlling who can use custom fields
- References

In the Overview you will learn the basics of what a custom field is. Next, how to use custom fields is explained. This will include defining a custom field

to your installation, activation of a newly defined custom field, storing data into a custom field, displaying the data stored in a custom field, controlling who can

define a custom field, and finally controlling who can view the contents of a custom field.

Overview

- Custom fields allow you to define and populate your own RACF database fields for USER and GROUP profiles

A custom field is an installation defined field name which can hold data associated with a USER or GROUP profile and this information is stored in the RACF database.

Defining a custom field

- Define the name and attributes of your custom field with a profile in the CFIELD class:
 - ▶ *profile-type.CSDATA.custom-field-name*
 - ▶ the CFDEF segment defines the attributes of the custom field

TYPE	MAXLENGTH
FIRST	OTHER
MINVALUE	MAXVALUE
MIXED	HELP
LISTHEAD	

The first step in the process of using custom fields is to define the custom field to your installation. First, you need to determine the name of the custom field. Next, you need to determine whether this custom field is to be associated with USER or GROUP profiles. Also, you need to determine the attributes of the data that can be stored in this custom field. These attributes are defined with the nine keywords listed on the slide. The **TYPE** keyword defines the type of data which your custom field can store. There are four possible types of data: character, numeric, hexadecimal, and flag. The **MAXLENGTH** keyword defines the maximum amount of data that the custom field can store. Note that valid maximum length values vary depending upon the type of data specified. The **LISTHEAD** keyword defines the descriptive text which is used as a label when displaying the custom field. If no LISTHEAD is specified, it defaults to the name of the custom field followed by an equal sign. The **HELP** keyword defines the TWO help text associated with this custom field. For character data, the **FIRST** and **OTHER** keywords allow you to restrict what character values can be specified for the first character and subsequent characters of the data stored in the custom field. For character data, the **MIXED** keyword defines whether the data is case sensitive or always stored as upper case. For numeric data, the **MINVALUE** and **MAXVALUE** keywords define the minimum and maximum allowable values. Once you have determined your required custom field attributes, you then define this custom field using a profile in the CFIELD class. The profile name is of the form *profile-type.CSDATA.custom-field-name* where *profile-type* is either **USER** or **GROUP** and *custom-field-name* is the name of the field being defined. The attributes of this custom field are specified using corresponding keywords in the CFDEF segment for this profile.

Examples of defining a custom field

- Define a user custom field to hold character data
 - ▶ RDEFINE CFIELD USER.CSDATA.ADDRESS
CFDEF(TYPE(CHAR) MAXLENGTH(100) FIRST(ANY)
OTHER(ANY) MIXED(YES) HELP('HOME ADDRESS,
UP TO 100 CHARS') LISTHEAD('HOME ADDRESS='))
- Define a group custom field to hold numeric data
 - ▶ RDEFINE CFIELD GROUP.CSDATA.LEVEL
CFDEF(TYPE(NUM) MAXLENGTH(1) MINVALUE(1)
MAXVALUE(5) HELP('SECURITY ACCESS LEVEL,
FROM 1 TO 5. LEVEL 5 IS THE HIGHEST SECURITY
CLEARANCE.') LISTHEAD('SECURITY ACCESS
LEVEL='))

The first example defines a custom field named "ADDRESS" which can be specified for a user. This ADDRESS field can contain up to 100 characters and it will honor the case (upper or lower) of the data that is specified. Also, the characters that can be specified are any characters allowed by TSO parse. This includes blanks and punctuation, but the specified character string must be enclosed in single quotation marks (note that quoted values are only allowed if FIRST(ANY) and OTHER(ANY) are specified as attributes). When this custom field is displayed using the LISTUSER command, the data is prefixed by the LISTHEAD text "HOME ADDRESS=". Finally, the TSO help text associated with this custom field is specified using the HELP keyword.

The second example defines a custom field named "LEVEL" which can be specified for a group. This LEVEL field can contain a numeric value from 1 to 5. When this custom field is displayed using the LISTGRP command, the data is prefixed by the LISTHEAD text "SECURITY ACCESS LEVEL=". Finally, the TSO help text associated with this custom field is specified using the HELP keyword.

Activating a custom field

- Activate the CFIELD class
 - ▶ SETROPTS CLASSACT(CFIELD)
- Verify your custom field definitions are not in error
 - ▶ IRRDPI00 CHECK
 - allocate file(sysut1) dataset('sys1.samplib(irrdpsds)') shr
- Activate the custom field
 - ▶ IRRDPI00 UPDATE
- View all the custom fields in effect on your system
 - ▶ IRRDPI00 LIST (USER CSDATA)
 - ▶ IRRDPI00 LIST (GROUP CSDATA)

Now that your custom field has been defined using a CFIELD profile, you need to activate this custom field (“activate” means the custom field can be used for storing and displaying values).

To activate the custom field, you need to activate the CFIELD class and then successfully run a dynamic parse update by issuing the IRRDPI00 UPDATE command. If you define the custom field attributes (CFDEF keyword specifications) with any incompatible definitions then running IRRDPI00 UPDATE is not successful. You can check for any problems with your CFDEF definitions by running a IRRDPI00 CHECK command.

Note that you can view all the custom fields that are active on your system by issuing the IRRDPI00 LIST command.

Upon successful activation, the custom field can then be recognized as a valid keyword for the CSDATA segment for RACF commands: ADDUSER, ALTUSER, ADDGROUP, and ALTGROUP.

The IRRDPI00 command can be invoked as a TSO command, but requires that the SYSUT1 DD value be defined as the location (dataset member) of the dynamic parse definitions. In this foil, the “allocate file(sysut1) dataset('sys1.samplib(irrdpsds)') shr” is an example of defining SYSUT1, but this example should be modified to specify the location of your installation’s dynamic parse definitions. See Chapter 4 in the RACF System Programmer’s Guide for more details about the IRRDPI00 command.

Storing data in a custom field

- Use ADDUSER or ALTUSER with the CSDATA segment to add or replace data in a user's custom field
 - ▶ ALTUSER NICOLE CSDATA(ADDRESS('5 Main Street'))
- Use ADDGROUP or ALTGROUP with the CSDATA segment to add or replace data in a group's custom field
 - ▶ ALTGROUP FINANCE CSDATA(LEVEL(4))

Now that your custom field is defined and active, you can use this custom field to store data in the CSDATA segment of any user or group.

For custom fields for users, you can store data using the ADDUSER and ALTUSER commands. In this ALTUSER example, user NICOLE will have the ADDRESS custom field set to '5 Main Street'.

For custom fields for groups, you can store data using the ADDGROUP and ALTGROUP commands. In this ALTGROUP example, group FINANCE will have the LEVEL custom field set to 4.

Your data value specification must conform to the attributes you have assigned for this particular custom field. For character type custom fields, you can use single quotation marks around the character text *only* if the CFDEF attributes specify FIRST(ANY) and OTHER(ANY).

Displaying custom field data

- Use LISTUSER or LISTGRP with the CSDATA segment to display all custom field values that have been established for a user or group
 - ▶ You will only see what you are authorized to view
- Command syntax
 - ▶ LISTUSER <user ID> CSDATA NORACF
 - ▶ LISTGRP <group-name> CSDATA NORACF

Now that your custom field is defined and active and contains data, you can display the contents of this custom field.

For custom fields for users, you can display the contents of all custom fields associated with a particular user using the LISTUSER command with the CSDATA option.

For custom fields for groups, you can display the contents of all custom fields associated with a particular group using the LISTGRP command with the CSDATA option.

You will only see the custom fields which you are authorized to view.

Displaying user custom field data

- LISTUSER NICOLE CSDATA NORACF

USER=NICOLE

CSDATA INFORMATION

EMPNAME= Nicole Smith

HOME ADDRESS= 5 Main Street, Anywhere, IL 01234

EMPLOYEE SERIAL= 0000256400

HOME PHONE= 555-555-5555

In this example, Nicole's custom field information is displayed. Each custom field is displayed with the LISTHEAD text as a prefix. If no custom fields have been set for Nicole then the LISTUSER output will indicate that no CSDATA information exists.

Displaying group custom field data

- LISTGRP FINANCE CSDATA NORACF

INFORMATION FOR GROUP FINANCE

CSDATA INFORMATION

SECURITY ACCESS LEVEL= 4

In this example, the group FINANCE has its custom field information displayed. Each custom field is displayed with the LISTHEAD text as a prefix. If no custom fields have been set for group FINANCE then the LISTGRP output will indicate that no CSDATA information exists.

You will only see the custom fields which you are authorized to view.

Controlling who can define custom fields

- Authority to define custom fields:
 - ▶ Command authority to issue RACF commands RDEFINE/RALTER for the CFIELD class
 - ▶ CFDEF segment authority
 - Field-Level Access Checking (FLAC) can be used to control CFDEF segments and individual CFDEF keywords

Defining custom fields is controlled by your authority to issue the RDEFINE and RALTER commands with the CFDEF segment specified.

If Field-Level Access Checking (FLAC) is active, then profiles can be defined in the FIELD class to control who can define or modify custom field definitions.

Controlling who can use custom fields

- Authority to store data in custom fields:
 - ▶ Command authority to issue RACF commands
ADDUSER/ALTUSER/ADDGROUP/ALTGROUP
 - ▶ CSDATA segment authority
 - Field-Level Access Checking (FLAC) allows write access to the custom field

- Authority to display custom fields:
 - ▶ Command authority to issue RACF commands
LISTUSER/LISTGROUP
 - ▶ CSDATA segment authority
 - Field-Level Access Checking (FLAC) allows read access to the custom field

The ability to view and store custom field data is controlled by the ability to issue the appropriate RACF commands with the CSDATA segment specified.

If Field-Level Access Checking (FLAC) is active, then profiles can be defined in the FIELD class to control who can view or store individual custom fields. Using generic FLAC profiles allows you to control all custom fields or bunches of like named custom fields. For example, issuing 'RDEF FIELD USER.CSDATA.EMPNAME UACC(READ)' and activating and raelisting the FIELD class allows all users to view the empname custom field information for every RACF user ID.

References

- **Security Server Manuals:**
 - ▶ RACF Command Language Reference (SC22-7687)
 - ▶ RACF Security Administrator's Guide (SC22-7683)
 - Chapter 7 contains Field-Level Access Checking concepts
 - Chapter 24: Defining and using custom fields
 - ▶ RACF System Programmer's Guide (SA22-7681)
 - Chapter 4 contains IRRDPI00 command
- **RACF Web site:**
<http://www.ibm.com/servers/eserver/zseries/zos/racf>

Note that under the RDEFINE and RALTER commands, the CFDEF segment description in the **RACF Command Language Reference** contains very useful information.

Also, there is chapter 7 in the **RACF Security Administrator's Guide** which is devoted exclusively to the subject of custom fields. This chapter has the same basic objective as this education package except the chapter goes into far greater details and insights and is highly recommended.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_IEA-V1R10-RACF-CustomFields.ppt

This module is also available in PDF format at: [../IEA-V1R10-RACF-CustomFields.pdf](http://IEA-V1R10-RACF-CustomFields.pdf)

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM RACF

A current list of other IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.