RACF Multi-Factor Authentication Support:
V1.00 APARs: OA48359, OA48650
V2.00 APAR: OA50016
V3.00 APARs: OA50930, OA50931
V4.00 APAR: OA53002, OA53013
V5.00 APAR: OA54920

Summary of Changes		
Version	**Date**	**Nature of Change**
V1.00	03/23/16	Initial version: RACF APAR OA48359 & SAF APAR OA48650
V1.01	03/29/16	Various clarifications
V2.00	06/22/16	Version 2: RACF APAR OA50016 Support for IBM Touch Token: • New R_Factor function code 4: Set user factor data • New ALTUSER Messages: ◦ ICH21055I, ICH21056I, ICH21057I, ICH21058I • New DELUSER Messages: ◦ ICH04019I, ICH04020I, ICH04021I, ICH04022I RACF Support for IBM MFA application bypass: • The MFADEF Class is changed from GENERIC=DISALLOWED to GENERIC=ALLOWED • New R_GenSec callable service subfunction code 3 – Evaluate PassTicket Extended • New R_TicketServ callable service Ticket_options value 3 – Evaluate PassTicket Extended
V3.00	11/16/16	Version 3: RACF APAR OA50930 & SAF APAR OA50931 Support for Out-of-band authentication: • New MFA RACF database fields • Updated R_Factor function code 3: Get user factor data • New R_Factor function code 5: Get policy data • New ALTUSER keywords: ADDPOLICY & DELPOLICY • New ALTUSER Message: ICH21059I • LISTUSER lists new MFA user data • New RLIST, RDEFINE and RALTER keywords: ◦ FACTORS, ADDFACTORS, DELFACTORS, NOFACTORS, REUSE, TOKENTIMEOUT • New RACROUTE REQUEST=VERIFY parameter value: PASSCHCK=NOMFA

V4.00	11/14/2017	Version 4: RACF APAR OA53002 & SAF APAR OA53013 Support for compound authentication: • RACROUTE REQUEST=VERIFY support for compound authentication. Support for MFA integration: • New R_Factor function code 6: Get CTC
V5.00	5/1/2018	Version 5: RACF APAR OA54920 Enhancement to compound authentication: • RACROUTE REQUEST=VERIFY support for compound authentication with the password or password phrase as the initial part of the compound credential.

1 Introduction

The most common method for authenticating users to z/OS systems is by the use of passwords or password phrases. Unfortunately, passwords can present a relatively simple point of attack for exploitation. In order for systems which rely on passwords to be secure they must enforce password controls and provide user education. User's tend to pick common passwords, write down passwords and unintentionally install malware which can key log passwords. Additionally, building a extremely powerful dedicated password cracking computer system has become trivial and low-cost. Clients are looking for ways to raise the assurance level of their systems by requiring additional authentication factors for users.

Multi-factor Authentication (MFA):

A Multi-factor authentication system requires that multiple authentication factors be presented during logon in order to verify a user's identity. Each authentication factor must be from a separate category of credential types:

1) **Something you know:** A password or security question
2) **Something you have:** An ID badge or Cryptographic token device
3) **Something you are:** Fingerprint or other biometric data

By requiring multiple authentication factors, a user's account can not be compromised even if one of their factors is discovered.

MFA on z/OS:
IBM Multi-Factor Authentication for z/OS together with the RACF Security Server infrastructure create a layered defense by requiring selected z/OS users to logon with multiple authentication factors.

2 Overview of new functions

Before using the new functions, read the planning considerations below. A brief overview of the functions is included here to provide context for the subsequent documentation. Additional information is in the Updated RACF publications section below.

The main components of the support for MFA on z/OS consists of IBM Multi-Factor Authentication for z/OS and RACF enablement infrastructure.

IBM Multi-Factor Authentication for z/OS (IBM MFA):
IBM Multi-Factor Authentication for z/OS is a new product which provides support for authenticating with different authentication factors. RACF users can be configured to require authentication though IBM MFA. For these select users, RACF will call IBM MFA to help make the authentication decision during logon processing.

RACF MFA Infrastructure:
RACF is enhanced to provide infrastructure to enable IBM Multi-Factor Authentication for z/OS to integrate directly with the security server. The RACF MFA infrastructure consists of updates to the database, commands, callable services, logon processing and utilities.

Restriction: The ISPF panels and TSO helps are not updated for the new command operands with OA48359 and OA50930.

3 Planning

While the MFA support for RACF can be implemented on a per user basis, consider the following before making changes.

- Create a backup copy of your RACF database.
- Apply the RACF MFA APARs to all systems sharing the RACF database.
- Read the IBM Multi-Factor Authentication for z/OS documentation

3.1 Create a backup copy of your RACF database

Creating a backup of the RACF database is recommended whenever significant changes are being made to RACF and the RACF database.

3.2 Apply the RACF MFA APARs to all systems that share the RACF database

Make sure that the service is applied on all sharing systems, and that all the ++HOLD documentation has been reviewed.

3.3 RACF exit considerations.

Similar a to PassTicket behavior, a RACROUTE REQUEST=VERIFY pre-processing exit will not have any indication that the contents of the password fields are actually password data or data for IBM Multi-Factor Authentication for z/OS. The VERIFY post-processing exit will be able to determine if the user successfully authenticated with z/OS MFA by checking the new ACEEMFAA bit.

3.4 Performance considerations

Authentication requests using MFA may be slower than non MFA authentication requests. At the very least, MFA authentication will incur extra path length when calling IBM Multi-Factor Authentication for z/OS. Depending on the factor type, there may be additional considerations such as network calls to external authentication servers. Non MFA authentication requests should have little to no noticeable performance degradation.

4 Updated RACF publications

Chapters of the following RACF publications are affected by the new function:

Publication Name	Publication Number
z/OS Security Server RACF Security Administrator's Guide	SA23-2289
z/OS Security Server RACF Command Language Reference	SA23-2292
z/OS Security Server RACROUTE Macro Reference	SA23-2294
z/OS Security Server RACF Auditor's Guide	SA23-2290
z/OS Security Server RACF Callable Services	SA23-2293
z/OS Security Server RACF Macros and Interfaces	SA23-2288
z/OS Security Server RACF Data Areas	GA32-0885
z/OS Security Server RACF Messages and Codes	SA23-2291
z/OS Security Server RACF General User's Guide	SA23-2298

In the following sections, highlighting is used to denote changed information in existing documentation. Sections, tables, messages, command keywords, etc. without highlighting contain new information.

4.1 z/OS Security Server RACF Security Administrator's Guide

This information supplements information in the following chapters:

- Multi-factor Authentication – New Chapter is added
- *Operating Considerations* – Section *RACF commands for flushing a VLF cache* section is updated

4.1.1 Multi-Factor Authentication

Today, the most common way for users to access z/OS systems is by the use of passwords or password phrases. Due to the simplicity of passwords, they can present a relatively simple point of attack for exploitation. In order for systems that rely on passwords to be secure, they must enforce password controls and provide user education. Some of the common problems with a simple password is that users tend to: choose common passwords, write down their passwords, or unintentionally install malware that can key log passwords. A more secure option is for systems to require multiple authentication factors to verify the user's identity.

What is Multi-Factor Authentication?

A multi-factor authentication system requires that multiple authentication factors be presented during logon to verify a user's identity. Each authentication factor must be from a separate category of credential types:

- Something you know: A password or security question
- Something you have: An ID badge or cryptographic token device
- Something you are: Fingerprint or other biometric data

By requiring multiple authentication factors, a user's account cannot be compromised if one of their factors is discovered.

IBM MFA on z/OS

With APAR OA48359 and IBM Multi-Factor Authentication for z/OS, RACF provides support for authenticating with multiple authentication factors. RACF users can be configured to require authentication through IBM MFA. For these select users, RACF will call IBM MFA to assist in making the authentication decision during logon processing.

Configuring RACF for IBM MFA

There are a number of steps to be completed in order to begin using IBM Multi-Factor Authentication for z/OS with RACF. IBM MFA should be installed as described in the IBM Multi-Factor Authentication for z/OS Installation and Customization. Then, perform the following steps to configure RACF for MFA:

1. Define the factor to RACF:
   An IBM MFA factor is defined by creating an MFADEF class profile with the name

FACTOR.factor-name. Supported authentication factors are listed in the IBM Multi-Factor Authentication for for z/OS product documentation. Note that a single factor name may enforce multiple authentication factors during longon.

For example, to define the RSA SecurID factor supported by IBM MFA:
```
RDEFINE MFADEF FACTOR.AZFSIDP1
```

2. Assign the factor to users:
   ALTUSER command. The factor must be defined in the MFADEF class before this step can be completed. The sub-keywords of MFA are:

FACTOR/DELFACTOR

Use the FACTOR keyword to identify the name of the factor that is being added or modified.

Use the DELFACTOR keyword to delete a factor from a user profile.

ACTIVE/NOACTIVE

Use the ACTIVE keyword to activate a factor for use during logon.

Use the NOACTIVE keyword to disable a factor and revert to password checking.

TAGS/DELTAGS/NOTAGS

Use the TAGS keyword to assign configuration data that is specific to the factor. The data is specified in name:value format. The IBM Multi-Factor Authentication for z/OS product documentation contains information on supported tags. IBM MFA is called to validate the data. The MFA started task must be available when assigning tags, or the ALTUSER command fails.

Use the DELTAGS keyword to delete specific tags.

Use the NOTAGS keyword to delete all tags for the specified factor.

PWFALLBACK/NOPWFALLBACK

Use the PWFALLBACK keyword to allow the user to logon with a RACF password or password phrase whenever the ability to perform multi-factor authentication is not available (for example, the MFA started task is down).

PWFALLBACK is not factor-specific.

Use NOPWFALLBACK to require the user to always authenticate using MFA.

NOMFA

Use the NOMFA keyword to remove all MFA data from a user's profile. See the z/OS Security Server RACF Command Language Reference for more information on the MFA keywords.

Example:

To require a user to authenticate with RSA SecurID, but allow the user to logon with their RACF password when MFA is unavailable:

```
ALTUSER SLJAXON MFA(FACTOR(AZFSIDP1) ACTIVE PWFALLBACK
TAGS(SIDUSERID:SamLJ))
```


3. Activate MFA checking:

When setup is complete, activate the MFADEF class.

```
SETROPTS CLASSACT(MFADEF)
```

When this is completed, RACF will call IBM Multi-Factor Authentication for z/OS to perform user authentication for any user who has an active MFA factor.

MFA checking can be disabled for all users by deactivating the MDADEF class:

```
SETROPTS NOCLASSACT(MFADEF)
```

MFA considerations for the RACF password and password phrase

MFA information cannot be assigned to a PROTECTED user, and thus an MFA user must have a password or password phrase.

When the user is assigned the NOPWFALLBACK attribute, the password/phrase cannot be used to logon. In this case, consider assigning the user a long, random password phrase.

When the user is assigned the PWFALLBACK option, the user needs to maintain the password as usual. However, the password will not be able to be changed during logon unless MFA is unavailable, the user's password is expired, and the application prompts the user to enter a new password. The user's password can be changed using the PASSWORD or ALTUSER command. Additionally, the user's password can be changed by a password reset application which uses the RACROUTE REQUEST=VERIFY PASSCHK=NOMFA parameter.

MFA Application Bypass:
In some cases it may be desirable to bypass select z/OS applications from MFA processing. IBM MFA provides controls to allow the RACF administrator to name applications for which MFA authentication will not be enforced. The MFA bypass controls allow for different bypass policy for different RACF users. Refer to the IBM MFA publications for more details on the MFA application bypass features.

MFA Policy:
Installations can create MFA policies to define a set of rules that users must follow when authenticating with IBM MFA. The policy attributes are defined in the MFPOLICY segment of profiles in the MFADEF class. These policies can be associated with individual users with the ALTUSER ADDPOLICY keyword. Refer to the IBM MFA publications for more details on MFA Policies.

MFA Compound In-Band:

MFA users may be provisioned so that they are required to authenticate with both a token device code and their RACF authenticator (password or password phrase). Users configured for compound in-band may enter their token code and RACF authenticator concatenated together in the password phrase field of an application with a separator character between the two values. When the RACF authenticator is expired it can be

changed by passing the new value into the new password or new password phrase field of the application by itself without the token code portion. Refer to the IBM MFA publications for more information on MFA compound in-band support.

4.1.2 RACF commands for flushing a VLF cache

The new MFADEF class is added to the list of classes which can result in a VLF cache flush:

...
In an installation where no RACF database sharing occurs, issuing commands that deal with certain general resource classes or profiles can delete all stored security environments. Examples of this include activating, deactivating, or issuing SETROPTS NORACLIST(classname) or SETROPTS RACLIST(classname) REFRESH for these classes:

- APPCPORT
- APPL
- CONSOLE
- FACILITY (only when SETROPTS MLS is in effect)
- GTERMINL
- JESINPUT
- MFADEF
- SECLABEL
- SERVAUTH
- TERMINAL

4.2 z/OS Security Server RACF Command Language Reference

This information supplements information for the following RACF commands:

- ALTUSER
- RDEFINE
- RALTER
- RLIST
- LISTUSER

4.2.1 ALTUSER

The ALTUSER command is enhanced to support adding multi-factor information to the base segment of a user profile. Any user with an ACTIVE MFA factor will be authenticated though IBM Multi-Factor Authentication for z/OS during logon.

Authorization required

If you have the SPECIAL attribute, you can use the MFA operands.

If the owner of the user profile is within the scope of a group in which you have the group-SPECIAL attribute, you can use the MFA operands.

If you are the owner of the user's profile, you can use the MFA operands.

Syntax

[MFA(
[PWFALLBACK | NOPWFALLBACK]
[FACTOR(factor-name) | DELFACTOR(factor-name)]
[ACTIVE | NOACTIVE]
[TAGS(*tag-name*:*tag-value* …)
| DELTAGS(*tag-name …*)
| NOTAGS]
[ADDPOLICY(policy-name …)
| DELPOLICY(policy-name … | *)]
)
| NOMFA]

MFA

Specifies multi-factor authentication information for the user profile being changed. Information is stored in the base segment of the user's profile.

MFA can not be specified for a PROTECTED user.

The MFADEF class must be active before a user can logon with IBM MFA.

See z/OS Security Server RACF Security Administrator's Guide.

PWFALLBACK | NOPWFALLBACK

PWFALLBACK

When IBM MFA is unavailable or is unable to determine the validity of an ACTIVE factor, this user can logon to the system using any existing RACF authenticators such as their password, password phrase or PassTicket.

NOPWFALLBACK

When IBM MFA in unavailable or is unable to determine the validity of an ACTIVE factor, this user will not be able to logon to the system with any existing RACF authenticators.

NOPWFALLBACK is the default.

FACTOR | DELFACTOR

FACTOR(*factor-name***)**

Specifies an authentication factor for a user. If the user is not already registered to the factor, the factor will be added to the user. The specified factor must be defined in an MFADEF class profile named FACTOR.<factorName>. Other ALTUSER keywords such as ACTIVE and TAGS are specific to this specified factor.

Factor-name is a 1 - 20 character identifier. The characters can be alphabetic, numeric, or national.

A user is limited to 10 total factors. Only one factor may be specified in a single ALTUSER command.

DELFACTOR(*factor-name***)**

Deletes the specified factor from the list of authentication factors registered to this user.

Factor-name is a 1 - 20 character identifier. The characters can be alphabetic, numeric, or national.

ACTIVE | NOACTIVE

ACTIVE

The user is required to authenticate to IBM MFA with the specified factor to logon to the system when the MFADEF class is active.

NOACTIVE

The user is not required to authenticate to IBM MFA with the specified factor to logon to the system. NOACTIVE is the default.

TAGS | DELTAGS | NOTAGS

TAGS(*tag-name*:*tag-value* ...**)**

Specifies tags and values for the specified factor.

If the *tag-value* you specify contains any blanks, the tag-name:tag-value pair must be enclosed in quotation marks.

The *tag-name* and *tag-value* pairs are factor specific and are defined by IBM MFA. ALTUSER calls IBM MFA to validate tag-names and tag-values. IBM MFA must be available for RACF to process the TAGS keyword. IBM MFA may reject a tag-name or tag-value during ALTUSER processing. IBM MFA may utilize these values during logon processing to authenticate a user. Refer to IBM Multi-Factor Authentication for z/OS User's Guide for documentation of each factor's configuration data parameters.

When the tag-name is not already present in the TAGS for the specified factor the tag-name is added. When the tag-name is already present for the specified factor, it is replaced with the new tag-value.

The *tag-name* is a 1 - 20 character case insensitive identifier and can consist of alphabetic or numeric characters. A factor is limited to 20 total tags.

The *tag-value* can be 1 - 1024 characters and can consist of any character.

DELTAGS(*tag-name ...***)**

Deletes specific tags for the specified factor.

The *tag-name* is a 1 - 20 character identifier and can consist of alphabetic or numeric characters.

The *tag-name* is ignored when it does not already exist for a specified factor.

NOTAGS

Removes all tags for the specified factor.

ADDPOLICY | DELPOLICY

ADDPOLICY(*policy-name …*)

Adds to this user's list of MFA authentication policies where *policy-name* is the name of an MFA policy profile defined in the MFADEF class. Policy-name is specified as only the unique name portion of the policy profile after the initial “POLICY.” qualifier.

A policy name must be between 1 and 20 characters.

Each user is limited to a maximum of 10 policy names.

DELPOLICY(*policy-name … **)

Deletes only the specified authentication policies from this user's list of policies.

Specifying the * character will delete all existing policies.

NOMFA

Specifies that RACF delete all MFA fields from the user's profile. The user is no longer required to provide additional authentication factors when logging on.

4.2.2 RDEFINE

The MFA segment contains factor specific multi-factor information for use by IBM Multi-Factor Authentication for z/OS.

Syntax

[MFA]

MFA

Specifies that RACF create an MFA segment in the MFADEF class profile. The MFA segment is intended to be updated only by IBM Multi-Factor Authentication for z/OS.

The MFPOLICY segment contains MFA authentication policy information for use by IBM Multi-Factor Authentication for z/OS.

Syntax

[MFPOLICY(
[FACTORS(*factor-name …*)]
[TOKENTIMEOUT(*timeout-seconds*)]
[REUSE(YES | NO)]
)]

MFPOLICY

Specifies multi-factor authentication policy information for the MFADEF class profile being changed.

FACTORS(*factor-name1 …*)

Specifies the list of factors names that are required for this authentication policy.

TOKENTIMEOUT(timeout-seconds)

Specifies the number of seconds for which out-of-band authentication with the policy is valid. That is, after having authenticated out-of-band with the policy to IBM MFA, the user must logon to a z/OS application within this number of seconds or the out-of-band authentication record will time out. When an out-of-band authentication record times out, a user must authenticate out-of-band again to the IBM MFA in order to logon.

The value of timeout-seconds can be between 1 and 86,400 (the number of seconds in a

day).

The default value is 300 (5 minutes).

REUSE(YES | NO)

Specifies whether this out-of-band authentication policy allows multiple z/OS logons using the out-of-band token within the TOKENTIMEOUT setting. When REUSE(NO) is specified the user must authenticate out-of-band with the policy prior to each z/OS logon.

REUSE(NO) is the default.

4.2.3 RALTER

The MFA segment contains factor specific multi-factor information for use by IBM Multi-Factor Authentication for z/OS.

Syntax

[MFA | NOMFA]

The MFA segment is intended to be updated only by IBM Multi-Factor Authentication for z/OS.

MFA

Specifies that RACF create an MFA segment in the MFADEF profile.

NOMFA

Specifies that RACF delete the MFA segment from the MFADEF profile.

The MFPOLICY segment contains MFA authentication policy information for use by IBM Multi-Factor Authentication for z/OS.

Syntax

[MFPOLICY(
[FACTORS(*factor-name ...*)
| ADDFACTORS(*factor-name …*)
| DELFACTORS(*factor-name …*)
| NOFACTORS]
[TOKENTIMEOUT(*timeout-seconds*)]
[REUSE(YES | NO)]
)

NOMFPOLICY]

MFPOLICY

Specifies multi-factor authentication policy information for the MFADEF class profile being changed.

FACTORS | ADDFACTORS | DELFACTORS | NOFACTORS

Specifies the list of factors that are required to satisfy this authentication policy.

FACTORS(*factor-name1 …*)

Specifies the list of factors names that are required in order to satisfy this authentication policy.

ADDFACTORS(*factor-name1 …*)

Adds to the list of factors names that are required in order to satisfy this authentication policy.

DELFACTORS(*factor-name1 …*)

Deletes the names from the list of factors names that are required in order to satisfy this authentication policy.

NOFACTORS

Removes the list of factor names from this authentication policy.

TOKENTIMEOUT(timeout-seconds)

Specifies the number of seconds for which out-of-band authentication with the policy is valid. That is, after having authenticated out-of-band with the policy to IBM MFA, the user must logon to a z/OS application within this number of seconds or the out-of-band authentication record will time out. When a out-of-band authentication record times out, a user must authenticate out-of-band again to IBM MFA in order to logon.

The value of timeout-seconds can be between 1 and 86,400 (the number of seconds in a day).

The default value is 300 (5 minutes).

REUSE(YES | NO)

Specifies whether this out-of-band authentication policy allows multiple z/OS logons using the out-of-band token within the TOKENTIMEOUT setting. When REUSE(NO) is specified the user must authenticate out-of-band with the policy prior to every z/OS logon.

REUSE(NO) is the default.

NOMFPOLICY

Specifies that RACF deletes the MFPOLICY segment from the MFADEF profile.

4.2.4 RLIST

A new MFA segment is added. RLIST is enhanced to display MFA information.

Syntax

[MFA]

MFA

Specifies that the MFA segment information should be listed for profiles in the MFADEF class profile.

Example RLIST output for the MFA segment

```
RLIST MFADEF FACTOR.FACT01 MFA
...
MFA INFORMATION:
----------------------------
MFADATA is defined.
```

A new MFPOLICY segment is added. RLIST is enhanced to display MFA authentication policy information.

Syntax

[MFPOLICY]

MFPOLICY
Specifies that the MFPOLICY segment information should be listed for profiles in the MFADEF class.

Example RLIST output for the MFPOLICY segment

```
RLIST MFADEF POLICY.GENUSER1 MFPOLICY
...
MFPOLICY INFORMATION:
----------------------------
FACTORS = FACTOR1 FACTOR2
TOKENTIMEOUT = 00000120
```



```
REUSE = YES
```

4.2.5 LISTUSER

LISTUSER is updated with a new MFA keyword to display MFA information in the base segment of a user profile.

Syntax

[MFA]

MFA

Specifies multi-factor authentication information should be listed for the user. The MFA keyword is ignored when NORACF is specified.

Example output for LISTUSER MFA when MFA information exists

```
LISTUSER USER01 MFA
...

MULTIFACTOR AUTHENTICATION INFORMATION:
---------------------------------------
  PASSWORD FALLBACK IS NOT ALLOWED
  AUTHENTICATION POLICIES:
    USERPOL1
    USERPOL2
  FACTOR = FACTORA
    STATUS = ACTIVE
    FACTOR TAGS =
      TAGONE:ABC
      TAGTWO:1234
  FACTOR = COLUMBUS
    STATUS = INACTIVE
    FACTOR TAGS =
      SHIP1:Nina
      SHIP2:Pinta
```



```
SHIP3:Santa Maria
```


4.3 z/OS Security Server RACF Messages and Codes

This information supplements RACF messages.

4.3.1 ALTUSER command messages

ICH21046I MFA cannot be specified for PROTECTED user user-ID.

Explanation: The MFA keywords are used to configure multi-factor authentication data and are not meaningful for a PROTECTED user.

System action: All MFA information is ignored. Command processing continues.

User response: Specify a different user ID, or assign the user ID a password or, preferably, a password phrase.

ICH21047I The FACTOR keyword must be specified when specifying other factor related keywords. No MFA data is updated.

Explanation: The FACTOR operand of the MFA keyword is required when specifying ACTIVE, NOACTIVE, TAGS, DELTAGS, or NOTAGS.

System action: All MFA information is ignored, including fields that are not factor-specific. Command processing continues.

User response: Correct the command.

ICH21048I Factor name factor-name cannot be added until the profile-name profile is created in the MFADEF class.

Explanation: The use of a given factor is enabled for the system when the security administrator defines the factor name in the MFADEF class, in the format demonstrated by profile-name. This must be a discrete profile. Until the profile is defined, the factor cannot be assigned to any users.

System action: All MFA information is ignored, including fields that are not factor-specific. Command processing continues.

User response: Either correct the factor name or define the factor in the MFADEF class for system use.

ICH21049I A maximum of *max-factor* factors can be specified for user *user-ID*.

Explanation: The command attempted to assign a factor that would exceed the limit of *max-factor* factors for user *user-ID*.

System action: All MFA information is ignored, including fields that are not factor-specific. Command processing continues.

User response: Either correct the factor name, or remove a different factor from the user's profile.

ICH21050I A maximum of *max-tag* tags can be specified for factor *factor-name* and user *user-ID*.

Explanation: The command attempted to assign a tag that would exceed the limit of *max-tag* tags for the specified factor name and user-ID.

System action: All MFA information is ignored, including fields that are not factor-specific. Command processing continues.

User response: Either correct the tag name, or remove a different tag from the factor definition in the user's profile.

ICH21051I IBM MFA detected an error in the *name-or-value* of tag *tag-name* with the following message: *MFA-msg*

Explanation: RACF contacted IBM Multi-Factor Authentication for z/OS to validate the tag name and value specified, and IBM MFA reflected an error as described the text of *MFA-msg*. If no message is returned, *MFA-msg* will contain the string “'*No message returned*'”.

System action: All MFA information is ignored, including fields that are not factor-specific. Command processing continues.

User response: Look up the message in the IBM Multi-Factor Authentication for z/OS documentation for additional information.

ICH21052I Unable to contact IBM MFA to validate tag data. No MFA data is updated.

Explanation: RACF could not contact IBM Multi-Factor Authentication for z/OS to validate the tag name(s) and value(s) specified. RACF uses a PC service to pass the tag data to IBM MFA. IBM MFA provides the PC number using a name/token pair. RACF received a non-zero return code when using the IEANTRT service to obtain the PC value.

System action: All MFA information is ignored, including fields that are not factor-specific. Command processing continues.

User response: Ensure that IBM MFA address space is active and has been configured properly. See IBM Multi-Factor Authentication for z/OS publications for additional information.

ICH21053I Unexpected return code=*return-code* and reason code=*reason-code* from IBM MFA while processing user *user-id*.

Explanation: RACF encountered an unexpected error from IBM Multi-Factor Authentication for z/OS while attempting to validate the tag name(s) and value(s) specified. RACF uses a PC service to pass the tag data to IBM MFA. The IBM MFA PC returned unexpected return codes.

System action: All MFA information is ignored, including fields that are not factor-specific. Command processing continues.

User response: Ensure that IBM MFA address space is active and has been configured properly. See IBM Multi-Factor Authentication for z/OS publications for additional information.

ICH21054I Factor *factor-name* for user *user-ID* contains tag data which is not valid. Use the NOTAGS operand to remove the tag data.

Explanation: The tag data associated with the specified factor and user in the RACF database is not valid.

System action: All MFA information is ignored for the specified user, including fields that are not factor-specific. Command processing continues.

User response: Use the ALTUSER command with the NOTAGS operand to remove the tag data which is not valid. For example, issue the following command:
ALTUSER *user-ID* MFA(FACTOR(*factor-name*) NOTAGS)

ICH21055I Unable to notify IBM MFA of tag deletion for user *user-ID* and factor *factor-name*. Tag data is deleted.

Explanation: RACF attempted to notify IBM Multi-Factor Authentication for z/OS for the deletion of tag data, but the notification failed. RACF uses a PC service to pass the tag data to IBM MFA. IBM MFA provides the PC number using a name/token pair. RACF received a non-zero return code when using the IEANTRT service to obtain the PC value.

System action: The tag data is deleted from the RACF database. Command processing continues.

User response: Determine the problem with IBM Multi-Factor Authentication for z/OS.

ICH21056I Error during notification of IBM MFA for deletion of tag *tag-name* for user *user-ID*and factor *factor-name*with the following message: *MFA-msg*

Explanation: RACF contacted IBM Multi-Factor Authentication for z/OS to delete the tag name noted in the message, and IBM MFA reflected an error as described in the text of *MFA-msg*. If no message is returned, *MFA-msg* will contain the string "'*No message returned*'".

System action: The tag data is deleted from the RACF database. Command processing continues.

User response: Look up the message in the IBM Multi-Factor Authentication for z/OS documentation for additional information.

ICH21057I Unexpected return code=*return-code* and reason code=*reason-code* from IBM MFA during tag deletion notification for user *user-id*and factor *factor-name*. Tag data is deleted.

Explanation: RACF encountered an unexpected error from IBM Multi-Factor Authentication for z/OS while attempting to notify IBM MFA that tag data has been deleted for the user and factor noted in the message. RACF uses a PC service to pass the tag data to IBM MFA. The IBM MFA PC returned unexpected return codes.

System action: The tag data is deleted from the RACF database. Command processing continues.

User response: Determine the problem with IBM Multi-Factor Authentication for z/OS.

ICH21058I Factor *factor-name* for user *user-ID* contains tag data which is not valid. Tag data is deleted and IBM MFA is not notified.

Explanation: While deleting tag data for the specified factor and user, RACF detected tag data which is not valid. IBM Multi-Factor Authentication for z/OS is usually notified when tag

data is deleted; since the tag data is not valid, notification to IBM MFA is not attempted.

System action: The tag data is deleted from the RACF database. Command processing continues.

User response: No further action is required.

ICH21059I A maximum of *max-policies* policy names can be specified. *policy-name* not added to user userid.

Explanation: The *policy-name* policy is ignored.

System action: Command processing stops with no update to the user.

User response: Remove an existing policy before attempting to add another policy.

4.3.2 LISTUSER messages

ICH30016I Tag data is not valid. Use the ALTUSER command with the NOTAGS operand to remove the tag data.

Explanation: The tag data in the RACF database associated with the user and factor currently being displayed is not valid.

System action: The tag data is not displayed for the user and factor. Command processing continues.

Problem determination: Use the ALTUSER command with the NOTAGS operand to remove the tag data that is not valid. For example, issue the following command: ALTUSER user-ID MFA(FACTOR(factorname) NOTAGS).

4.3.3 DELUSER messages

ICH04019I Unable to notify IBM MFA of tag deletion for user *user-id* and factor *factor-name*. Tag data is deleted.

Explanation: RACF attempted to notify IBM Multi-Factor Authentication for z/OS for the deletion of tag data, but the notification failed. RACF uses a PC service to pass the tag data to IBM MFA. IBM MFA provides the PC number using a name/token pair. RACF received a non-zero return code when using the IEANTRT service to obtain the PC value.

System action: The tag data is deleted from the RACF database. Command processing continues.

User response: Determine the problem with IBM Multi-Factor Authentication for z/OS.

ICH04020I Error during notification of IBM MFA for deletion of tag *tag-name* for user *user-ID* and factor *factor-name* with the following message: *MFA-msg*

Explanation: RACF contacted IBM Multi-Factor Authentication for z/OS to delete the tag name noted in the message, and IBM MFA reflected an error as described the text of *MFA-msg*. If no message is returned, *MFA-msg* will contain the string "'*No message returned*'".

System action: The tag data is deleted from the RACF database. Command processing continues.

User response: Look up the message in the IBM Multi-Factor Authentication for z/OS documentation for additional information.

ICH04021I Unexpected return code=*return-code* and reason code=*reason-code* from IBM MFA during tag deletion notification for user *user-id*and factor *factor-name*. Tag data is deleted.

Explanation: RACF encountered an unexpected error from IBM Multi-Factor Authentication for z/OS while attempting to notify IBM MFA that tag data has been deleted for the user and factor noted in the message. RACF uses a PC service to pass the tag data to IBM MFA. The IBM MFA PC returned unexpected return codes.

System action: The tag data is deleted from the RACF database. Command processing continues.

User response: Determine the problem with IBM Multi-Factor Authentication for z/OS.

ICH04022I Factor *factor-name* for user *user-ID* contains tag data which is not valid. Tag data is deleted and IBM MFA is not notified.

Explanation: While deleting tag data for the specified factor and user, RACF detected tag data which is not valid. IBM Multi-Factor Authentication for z/OS is usually notified when tag data is deleted; since the tag data is not valid, notification to IBM MFA is not attempted.

System action: The tag data is deleted from the RACF database. Command processing continues.

User response: No further action is required.

4.3.4 Dynamic parse (IRRDPI00) messages

IRR52221I *keyword-name* is intended to be updated only by IBM MFA. Command processing terminated.

Explanation: The named keyword-name in the MFA segment is not allowed to be specified by RACF commands. Some fields in the MFA segment are intended to be updated only by IBM Multi-Factor Authentication for z/OS.

System Action: Command processing ends.

User Response: Correct the command.

4.3.5 RACROUTE REQUEST=VERIFY Messages

ICH70008I IBM MFA Message:
mfa-message

Explanation: RACROUTE REQEST=VERIFY received message text from IBM Multi-Factor Authentication for z/OS while processing a request to authenticate an with an ACTIVE MFA factor.

User Response: See IBM Multi-Factor Authentication for z/OS User's Guide to evaluate the *mfa-message* and take the appropriate action.

ICH408I LOGON/JOB INITIATION - MULTIFACTOR AUTHENTICATION FAILURE

Explanation: A user with active multifactor authentication factors attempted to log on with invalid credentials as determined by IBM Multi-Factor Authentication for z/OS.

System Action: RACF prevents the user from logging on.

User Response: Correct any errors in the credentials and try again.

ICH408I LOGON/JOB INITIATION - MULTIFACTOR AUTHENTICATION UNAVAILABLE

Explanation: A user with active multifactor authentication factors attempted to log on but either IBM Multi-Factor Authentication for z/OS was unavailable to verify them, or RACF was unable to contact IBM MFA. The user is not allowed to fall back to the use of a password or password phrase. The SMF record contains additional information regarding the unavailability of IBM MFA.

System Action: RACF prevents the user from logging on.

User Response: If the problem persists, contact a system administrator.

4.4 z/OS Security Server RACF RACROUTE Macro Reference

This information supplements the information in Chapter *System Macros* in the *RACROUTE REQUEST=VERIFY* and *RACROUTE REQUEST=VERIFYX* sections.

4.4.1 RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX

The NEWPHRASE and NEWPASS parameters descriptions are updated to indicate:

A new password phrase cannot be set using a password for authentication, nor can a new password be set using a password phrase for authentication. However, the application should code the password- and phrase-related keywords as appropriate depending on the length of user-entered data, and let RACROUTE determine its validity.

The PASSCHK=YES parameter description is updated to indicate:

For a user subject to multi-factor authentication (MFA), RACF passes the contents of the PASSWRD=, NEWPASS=, PHRASE=, and NEWPHRASE= keywords to the MFA started task, where they are evaluated as MFA credentials. If the credentials are unable to be evaluated as MFA credentials (for example, if the MFA started task is unavailable), they are evaluated as RACF credentials if the user is allowed to fall back to password-based authentication.

A new value 'NOMFA' is added for the PASSCHK parameter. The PASSCHK parameter is updated as follows:

,PASSCHK=YES
,PASSCHK=NO
,PASSCHK=NOMFA

specifies whether the user's password, password phrase, MFA credentials or OIDCARD is to be verified.

YES RACROUTE REQUEST=VERIFY verifies the user's password, password phrase, MFA credentials or OIDCARD.
There are some circumstances where verification does not occur even though PASSCHK=YES is specified. Some examples are surrogate processing (see z/OS Security Server RACF Security Administrator's Guide) or when the START or the ENVRIN keywords are specified.

NO The user's password, password phrase, MFA credentials or OIDCARD is not verified. And, if the logon is successful, no message is issued.

NOMFA Same as YES, except password and password phrase parameters are always verified as a password or password phrase, not as MFA credentials, even for user's who have an ACTIVE MFA factor.
Use of the PASSCHK=NOMFA parameter requires that RELEASE=1.9 or later be specified.

New RACROUTE REQUEST=VERIFY return codes and reason codes:

SAF Return Code: 8
RACF Return Code (hex): 68 – Indicates that an error occurred while processing an MFA request.
RACF Reason code (hex): 0004yyyy – An error occurred while RACROUTE REQUEST=VERIFY was processing the results of an MFA authentication request. “yyyy” contains diagnostic data.

SAF Return Code: 8
RACF Return Code (hex): 68 – Indicates that an error occurred while processing an MFA request.
RACF Reason code (hex): 0008yyyy – An error occurred while RACROUTE REQUEST=VERIFY was processing the results of an MFA authentication request. “yyyy” contains diagnostic data.

4.5 z/OS Security Server RACF Auditor's Guide

This information supplements the information in Chapter *The RACF auditor* in the logging section.

4.5.1 Logging

A new condition is added to the list of events which are always logged:

RACF always logs information about certain events because knowing about these events is essential to an effective data-security mechanism. The events that RACF always logs are:
...

- A successful RACROUTE REQUEST=VERIFY is possible under the following conditions:
  - SETROPTS AUDIT(USER) is active and a user's password or password phrase is changed
  - authentication using a PassTicket
  - authentication of an IBM Multi-Factor Authentication user using a password or password phrase.

4.6 z/OS Security Server RACF General User's Guide

This information supplements the information in Chapter *What is RACF* in the Identifying and verifying users section.

4.6.1 Identifying and verifying users

...
You can be required to authenticate with multiple authentication factors instead of a password or password phrase. In this case, what you enter when logging on is determined by IBM Multi-Factor Authentication for z/OS. For example, you may be required to enter an RSA SecurID token code and PIN. See IBM Multi-Factor Authentication for z/OS User's Guide for more information.

4.7 z/OS Security Server RACF Callable Services

This information supplements the information in Chapter *Callable services descriptions*

- The *R_Admin* section is updated to add new MFA fields
- The *R_Factor* section is added
- The *R_Gensec* section is updated to add a new subfunction code
- The *R_Ticketserv* section is updated to add a new Ticket_options value

4.7.1 R_Admin (IRRSEQ00): Authentication Factor Service

The update-user function and extract-user functions are updated to support the new MFA fields in the BASE segment.

The R_admin reference appendix is updated:

- The table *BASE segment fields* is updated to add the MFA fields.
- The table *MFA fields* is added

Base segment fields:

Field name	Flag byte value	ADDUSER/ALTUSER keyword reference, or LISTUSER heading (for output-only fields)	Allowed on add requests	Allowed on alter requests	Returned on extract requests
The MFA fields FACTORN (including subfields), MFAPOLNM, and MFAFLBK are represented differently on input (ADMN_ALT_USER) vs. output (ADMN_XTR_USER).					
FACTORN	N/A	**Note:** This is the list header field for the 42-dimensional array consisting of the following three fields.	No	No	Yes
FACTOR	'Y'	MFA(FACTOR(xx))	No	Yes	Yes
	'D'	MFA(DELFACTOR(xx))	No	Yes	
FACACTV (boolean)	'Y'	MFA(ACTIVE)	No	Yes	Yes
	'N'	MFA(NOACTIVE)	No	Yes	
FACTAGnn	N/A	FACTOR TAGS =	No	No	Yes
FACVALnn	N/A	FACTOR TAGS =	No	No	Yes
Note: On output for a multi-factor authentication user, each tag is represented using separate fields for the tag name and tag value (for example, FACTAG01 contains the name of the first tag and FACVAL01 contains the value of the first tag). Twenty pairs of these fields are returned for every MFA user, regardless of how many tags actually exist. A non-zero length indicates the actual existence of the tag in the user profile.					
MFAFLBK (boolean)	'Y'	MFA(PWFALLBACK)	No	Yes	Yes
	'N'	MFA(NOPWFALLBACK)	No	Yes	Yes

MFAPOLNM (list MFAPOLN)	'A'	MFA(ADDPOLICY(xx …))	No	Yes	Yes
	'D'	MFA(DELPOLICY(xx …))	No	Yes	

MFA segment fields:

Field name	Flag byte value	ADDUSER/ALTUSER keyword reference, or LISTUSER heading (for output-only fields)	Allowed on add requests	Allowed on alter requests	Returned on extract requests
FACTOR	'Y'	MFA(FACTOR(xx))	No	Yes	Yes
	'D'	MFA(DELFACTOR(xx))	No	Yes	
FACACTV (boolean)	'Y'	MFA(ACTIVE)	No	Yes	Yes
	'N'	MFA(NOACTIVE)	No	Yes	
FACTAGS	'Y'	MFA(TAGS(xx ...))	No	Yes	No
	'D'	MFA(DELTAGS(xx ...))	No	Yes	
	'N'	MFA(NOTAGS)	No	Yes	
MFAFLBK (boolean)	'Y'	MFA(PWFALLBACK)	No	Yes	Yes
	'N'	MFA(NOPWFALLBACK)	No	Yes	
MFAPOLNM (list MFAPOLN)	'A'	MFA(ADDPOLICY(xx …))	No	Yes	Yes
	'D'	MFA(DELPOLICY(xx …))	No	Yes	

MFPOLICY segment fields:

Field name	Flag byte value	RDEFINE/RALTER keyword reference	Allowed on add requests	Allowed on alter requests	Returned on extract requests
FACTORS (list FACTORSN)	'Y'	MFPOLICY(FACTORS(xx ...))	Yes	Yes	Yes
	'A'	MFPOLICY(ADDFACTORS(xx …))	No	Yes	
	'D'	MFPOLICY(DELFACTORS(xx …))	No	Yes	
	'N'	MFPOLICY(NOFACTORS))	No	Yes	
TIMEOUT	'Y'	MFPOLICY(TOKENTIMEOUT(xx))	Yes	Yes	Yes
REUSE (boolean)	'Y'	MFPOLICY(REUSE(YES))	Yes	Yes	Yes
	'N'	MFPOLICY(REUSE(NO))	Yes	Yes	

Notes:

- For the ADMN_ALT_USER function, MFA fields are treated as a non-BASE segment even though the fields reside in the BASE segment of the user profile.
- For the ADMN_XTR_USER function, MFA fields are returned in the BASE segment.

4.7.2 R_Factor (IRRSFA64): Authentication Factor Service

Function

The **R_Factor** service provides functions required by multi-factor authentication applications to store and retrieve associated data in the RACF database.

1. Get general factor data
2. Set general factor data
3. Get user factor data
4. Set user factor data
5. Get general policy data
6. Get cached token credential

Requirements

Authorization:
Any PSW key in supervisor state or problem state
Dispatchable unit mode:
Task or user
Cross memory mode:
PASN = HASN
AMODE:
64
3. RMODE:
Any
ASC mode:
Primary or AR mode
Recovery mode:
ESTAE. Caller cannot have an FRR active.
Serialization:
Enabled for interrupts
Locks:
No locks held
Control parameters:
The parameter list and the work area must be in the primary address space. The words containing ALETs must be in the primary address. The Num_parms parameter must be in the primary address space.

Linkage conventions

Callers in 64-bit addressing mode should link-edit the IRRSFA64 stub module with their code and use the IRRPCOMY mapping macro.

RACF authorization

Callers running in system key or supervisor state may specify any function code. Non-system key problem-state callers require the following authorization for each function code:

1. Get general factor data
READ access to the resource IRR.RFACTOR.MFADEF.*factorName* in the FACILITY class, where factorName matches a profile defined in the MFADEF class.
2. Set general factor data
UPDATE access to the resource IRR.RFACTOR.MFADEF.*factorName* in the FACILITY class, where *factorName* matches a profile defined in the MFADEF.
3. Get user factor data
READ access to the resource IRR.RFACTOR.USER in the FACILITY class.
4. Set user factor data
Update access to the resource IRR.RFACTOR.USER in the FACILITY class.
5. Get general policy data
READ access to the resource IRR.RFACTOR.POLICY.*policyName* in the FACILITY class, where policyName matches a profile defined in the MFADEF class.
6. Get cached token credential (CTC)
READ access to the resource IRR.RFACTOR.GETCTC in the FACILITY class.

Format

```
CALL IRRSFA64     (Work_area,
                   ALET, SAF_return_code,
                   ALET, RACF_return_code,
                   ALET, RACF_reason_code,
                   Num_parms,
                   Parm_ALET, Function_code,
                   Function_parmlist
                   )
```

Parameters

Work_area

The name of a 1024-byte work area for SAF. The work area must be in the primary address space.

ALET

The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. The words containing the ALETs must be in the primary address space.

SAF_Return_Code

The name of a fullword in which the SAF router returns the SAF return code.

RACF_Return_Code

The name of a fullword in which the service routine stores the return code.

RACF_Reason_Code

The name of a fullword in which the service routine stores the reason code.

Num_parms

Specifies the name of a fullword that contains the total number of parameters in the parameter list. The contents of this field must be set to eleven (x'0000000B').

Parm_ALET

The name of a word containing the ALET for all the parameters that follow, including function code specific parameter lists and areas referenced from them.

Function_code

The name of a 2-byte area containing the function code. The function code has one of the folowing values:

x'0001' - Get general factor data
x'0002' - Set general factor data
x'0003' - Get user factor data
x'0004' - Set user factor data
x'0005' - Get general policy data
x'0006' – Get Cached Token Credential

Function_parmlist

Specifies the name of the function code specific parameter list area for the function_code specified.

All address fields are 8-byte addresses. When referring to 31-bit storage addresses, the caller must make sure that the high-order word of the address field is set to binary zeros.

Function parmlist for x'0001' – Get general factor data

Offset	Length	Name	Description
0	0	FACT_GETF_PLIST	Name of structure
0	4	FACT_GETF_OPTIONS	Reserved. Must be initialized to zeroes.
4	4	FACT_GETF_FACTOR_LENGTH	Length (in bytes) of the factor name
8	8	FACT_GETF_FACTOR@	Address of the factor name.
16	4	*	Reserved. Must be initialized to zeroes
20	4	FACT_GETF_AF_LENGTH	Length (in bytes) of free-form application data area
24	8	FACF_GETF_AF@	Address of free-form application data area The area must be pre-allocated by the caller and its size specified in FACT_GETF_AF_LENGTH.

Function parmlist for x'0002' – Set general factor data

Offset	Len	Name	Description

0	0	FACT_SETF_PLIST	Name of structure
0	4	FACT_SETF_OPTIONS	Reserved. Must be initialized to zeroes.
4	4	FACT_SETF_FACTOR_LENGTH	Length (in bytes) of the factor name
8	8	FACT_SETF_FACTOR@	Address of the factor name
16	4	*	Reserved. Must be initialized to zeroes
20	4	FACT_SETF_AF_LENGTH	Length (in bytes) of free-form application data area. Must not exceed 4096. Specify 0 to delete the current value.
24	8	FACT_SETF_AF@	Address of free-form application data area The area must be pre-allocated by the caller and its size specified in FACT_SETF_AF_LENGTH.

Function parmlist for x'0003' – Get user factor data

Offset	Len	Name	Description
0	0	FACT_GETU_PLIST	Name of structure
0	4	FACT_GETU_OPTIONS	x'00000000' – Return application data area only. x'80000000' – Return FACT_UFT_POL in FACT_GETU_UF@.
4	4	FACT_GETU_UF_COUNT	Number of user factors. Must be initialized to zero.
8	4	*	Reserved. Must be initialized to zero.
12	4	FACT_GETU_UF_LENGTH	Total length (in bytes) of the user factor area, a contiguous block of storage for the user factor list, user factor field lists, user factor tag lists, and other variable-length data referenced by those lists.
16	8	FACT_GETU_UF@	Address of user factor area (see FACT_UF_ENTRY) The area must be pre-allocated by the caller and its size specified in FACT_GETU_UF_LENGTH.
24	1	FACT_GETU_USER_LENGTH	Length of User ID. Value must be from 1 to 8.
25	8	FACT_GETU_USER	User ID padded on the right with blanks
33	1	FACT_GETU_FALL_BACK	Value must be initialized to zero. On output, value may be -- x'01' – User can fall back x'02' -- User cannot fall back
34	14	*	Reserved.
48	4	FACT_GETU_POL_COUNT	Number of policies.
52	4	FACT_GETU_POL_OFFSET	Offset to policy list in the user factor area (FACT_GETU_UF@).

Function parmlist for x'0004' – Set user factor data

Offset	Len	Name	Description
0	0	FACT_SETU_PLIST	Name of structure
0	4	FACT_SETU_OPTIONS	Reserved. Must be initialized to zeroes.

4	4	FACT_SETU_UF_COUNT	Number of user factors. No more than 10 factors may be defined in the user profile.
8	4	*	Reserved. Must be initialized to zero.
12	4	FACT_SETU_UF_LENGTH	Total length (in bytes) of the user factor area, a contiguous block of storage for the user factor list, user factor field lists, user factor tag lists, and other variable-length data referenced by those lists.
16	8	FACT_SETU_UF@	Address of user factor area (see FACT_UF_ENTRY) The area must be pre-allocated by the caller and its size specified in FACT_SETU_UF_LENGTH.
24	1	FACT_SETU_USER_LENGTH	Length of User ID
25	8	FACT_SETU_USER	User ID padded on the right with blanks
33	1	FACT_SETU_FALL_BACK	The value must be x'00' indicating no change to the current setting.

Function parmlist for x'0005' – Get general policy data

Offset	Length	Name	Description
0	0	FACT_GETP_PLIST	Name of structure
0	4	FACT_GETP_OPTIONS	Reserved. Must be initialized to zeroes.
4	4	FACT_GETP_POLICY_LENGTH	Length (in bytes) of the policy name
8	8	FACT_GETP_POLICY@	Address of the policy name.
16	4	FACT_GETP_FL_COUNT	Number of factors. The factor entries start at FACT_GETP_PA@ and mapped by FACT_PF_ENTRY
20	4	FACT_GETP_PA_LENGTH	Length (in bytes) of the policy area
24	8	FACT_GETP_PA@	Address of the policy area, which must be pre-allocated by the caller and its size specified in FACT_GETF_PA_LENGTH.
32	4	FACT_GETP_TIMEOUT	Token time-out value in seconds.
36	1	FACT_GETP_REUSE	Token reuse setting. x'01' – Token can be reused x'02' – Token cannot be reused
37	15	*	Reserved. Must be initialized to zeroes.

Function parmlist for x'0006' – Get cached token credential (CTC)

Offset	Length	Name	Description
0	0	FACT_GETC_PLIST	Name of structure
0	4	FACT_GETC_OPTIONS	Reserved. Must be initialized to zeroes.
4	1	FACT_GETC_USER_LENGTH	Length of User ID.

5	8	FACT_GETC_USER	User ID. Padded on the right with blanks.
13	8	FACT_GETC_APPL	Application Name. Padded on the right with blanks. Optional. Set to all blanks when not supplied.
21	7	*	Reserved. Must be initialized to zeros.
28	4	FACT_GETC_CRED_LIST_NUM	Number of credentials in the credential list. Optional. Set to 0 to generate without credentials. Maximum number of credentials is 10.
32	8	FACT_GETC_CRED_LIST@	Address of the Credential_list. The credential_list is mapped by FACT_CRED_LIST.
40	8	FACT_GETC_CTC@	Output CTC area address. Area must be preallocated and 8 bytes in length.
48	4	FACT_GETC_POLICY_LEN	Length of MFA Policy Name. Optional. Set to 0 for default policy name. Maximum MFA Policy Name length is 20.
52	20	FACT_GETC_POLICY_NAME	MFA Policy Name. Padded on the right with blanks.
72	16	*	Reserved. Must be initialized to zeros.

User Factor List
Located at the beginning of the user factor area referenced by FACT_GETU_UF@ or FACT_SETU_UF@, this list is the contiguous set of user factor entries, each mapped by FACT_UF_ENTRY. The number of entries is specified by FACT_GETU_UF_COUNT or FACT_SETU_UF_COUNT.

Offset	Len	Name	Description
0	0	FACT_UF_ENTRY	Name of structure mapping
0	4	FACT_UF_FACTOR_LENGTH	Length of factor name
4	4	FACT_UF_FACTOR_OFFSET	Positive offset from FACT_GETU_UF@ or FACT_SETU_UF@ to the factor name. The factor profile must already exist.
8	4	FACT_UF_FIELDS_COUNT	Number of fields for this factor
12	4	FACT_UF_FIELDS_OFFSET	Positive offset from FACT_GETU_UF@ or FACT_SETU_UF@ to the user factor field list, mapped by FACT_UFF_ENTRY

User Factor Field List
Located at offset FACT_UF_FIELDS_OFFSET from the beginning of the user factor area, this list is a contiguous set of user factor fields entries, each mapped by FACT_UFF_ENTRY. The number of entries is specified by FACT_UF_FIELDS_COUNT in the associated factor entry.

For function 4, if the tag already exists for the factor, the tag and value are replaced; unless the specified value length is zero, in which case the they are deleted. If the tag does not exist in the database and its value length is nonzero, it is added. No more than 20 tags may be specified per user factor.

Offset	Len	Name	Description

0	0	FACT_UFF_ENTRY	Name of structure mapping
0	4	FACT_UFF_FIELD_ID	Numeric identifier of the user field to update. Constant values for field identifiers are defined in IRRPCOMY. FACT_FID_TAGS (variable len) - Tag list, see FACT_UFT_ENTRY FACT_FID_ACTIVE (19 bytes) – User factor active date (UTC) FACT_FID_POLICIES (variable len) – User policies, FACT_UFT_POL
4	4	FACT_UFF_VALUE_LENGTH	Length of the user factor field value.
8	4	FACT_UFF_VALUE_OFFSET	Positive offset from FACT_GETU_UF@ or FACT_SETU_UF@ to the user factor field value

Policy factor list

Located at the beginning of the policy area referenced by FACT_GETP_PA@, this list is the contiguous set of factor entries, each mapped by FACT_PF_ENTRY. The number of entries is specified by FACT_GETP_FL_COUNT.

Offset	Length	Name	Description
0	0	FACF_PF_ENTRY	Name of structure
0	4	FACT_PF_FACTOR_LENGTH	Length of factor name
4	4	FACT_PF_FACTOR_OFFSET	Positive offset from start of policy area to factor name
8	8	*	Reserved

User Policy list

Located at the beginning of the user factor area referenced by FACT_GETU_UF@, this list is the contiguous set of policy entries, each mapped by FACT_UP_ENTRY. The number of entries is specified by FACT_GETU_POL_COUNT.

Offset	Length	Name	Description
0	0	FACT_UP_ENTRY	Name of structure
0	4	FACT_UP_POLICY_LENGTH	Length of policy name
4	4	FACT_UP_POLICY_OFFSET	Positive offset from start of policy area to policy name
8	8	*	Reserved

User Factor Tag List

The user factor tag list begins with a 2-byte header (FACT_UFT_HEADER) which must be initialized to zero. The subsequent fields (FACT_UFT_PAIR_LENGTH through FACT_UFT_VALUE) are repeated as a group for each tag/value pair in the list.

Offset	Len	Name	Description
0	0	FACT_UFT_LIST	Name of structure mapping
0	2	FACT_UFT_HEADER	Reserved. Must be zero.
2	2	FACT_UFT_PAIR_LENGTH	Total length of this tag/value pair entry, not including the length of this field
4	2	FACT_UFT_TAG_ATTRIBUTES	Tag attributes
6	2	FACT_UFT_TAG_LENGTH	Length of tag
8	var	FACT_UFT_TAG	Tag name
*	2	FACT_UFT_VALUE_LENGTH	Length of value. The length may not exceed 1024.
*	var	FACT_UFT_VALUE	Value associated with tag. Type is EBCDIC character data.

User Factor Active Date

The user factor active date is the time after which the factor is considered 'active' for the user. Prior to this time, the user is not required to authenticate with the factor in order to logon to the system.

On input for function 4, the active date may be specified in one of the following ways:

- The 7-character keyword 'CURRENT', which stores the current UTC time.
- A length of zero (FACT_UFF_VALUE_LENGTH = 0), which clears any existing value from the user profile, resulting in the 'noactive' default.

On output for function 3, the service returns a 19-character UTC time of the format 'yyyy-mm-dd hh:mm:ss' if set in the user profile. The service does not return an active date field if the value was cleared or never set.

Credential List

The credential list is a list of authentication credentials for a user. The FACT_CRED_LIST structure repeats for each credential. The total length of all credentials combined must be no greater than 8192 bytes.

Offset	**Len**	**Name**	**Description**
0	0	FACT_CRED_LIST	Name of structure mapping
0	20	FACT_CRED_TYPE	Type of credential. Credential type is the factor name for the input credential value.
20	4	FACT_CRED_LENGTH	Credential value length. Length must be at least 1 byte and no more than 8192 bytes.
24	8	FACT_CRED_VAL_PTR	Pointer to Credential value.
32	16	*	Reserved. Must be set to zero.

Return and reason codes

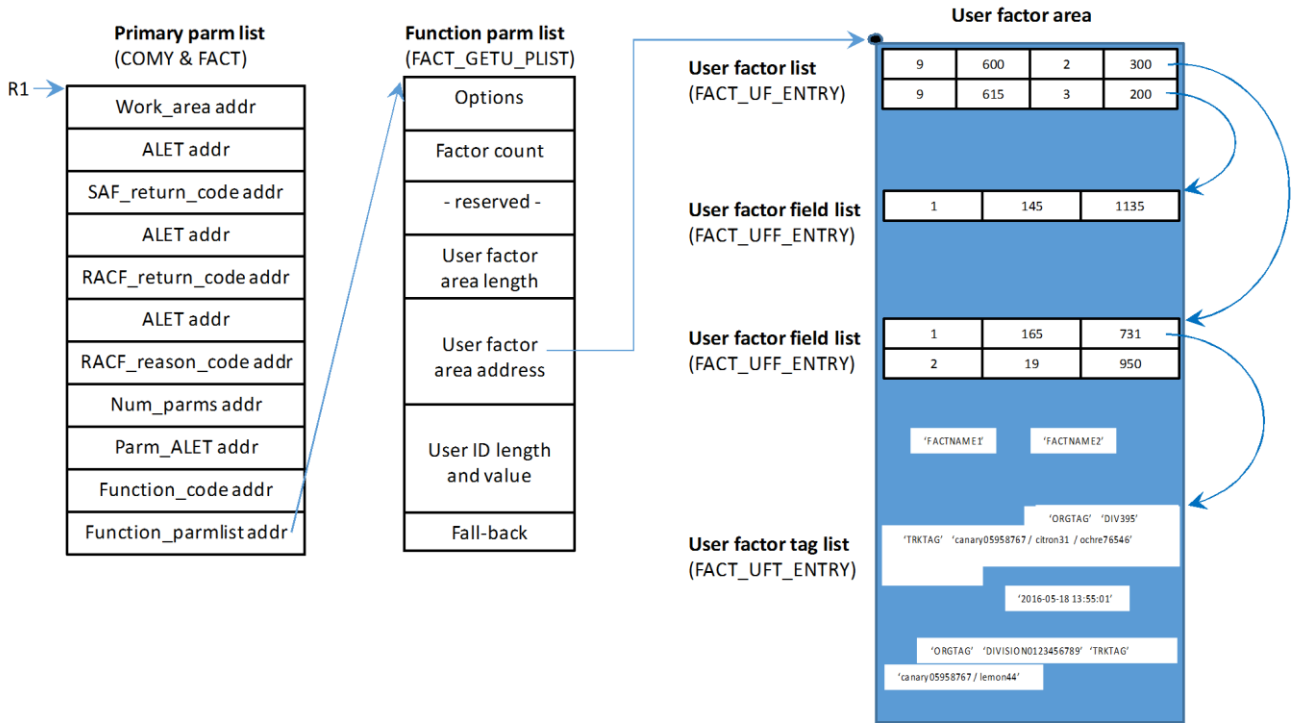
IRRSFA64 returns the following values in the reason and return code parameters:

SAF return code	RACF return code	RACF reason code	Explanation
0	0	0	Successful completion
4	0	0	RACF not installed
8	8	4	An internal error has occurred during RACF processing
8	8	8	Unable to establish recovery
8	8	12	Caller is not authorized
8	8	16	MFADEF class not active.
8	8	20	IBM MFA unavailable.
8	8	24	Error calling IBM MFA.
8	12	4	Factor not defined
8	12	8	User not defined
8	12	12	Policy not found
8	16	n	A RACF ICHEINTY error occurred while retrieving data. The reason code may be useful to IBM service.
8	20	n	A RACF ICHEINTY error occurred while storing data. The reason code may be useful to IBM service.
8	24	0	A tag list error has been detected in the data base
8	24	4	Too many tags for the user factor
8	24	8	Too many factors for the user
8	30	n	An unexpected logic error has been encountered. The reason code may be useful to IBM service.
8	100	n	A parameter list error has been detected. The RACF reason code identifies the parameter in error. The reason code is the ordinal position of the parameter in error, relative to the start of COMY.
8	104	n	A function-specific parameter list (pointed to by the function_parmlist parameter) error has been detected. The RACF reason code identifies the offset of the field in error.
8	108	n	A factor list error has been detected. The reason code is one of the following: 0 – The offset to the factor name plus its length extends beyond the user factor area 4 – The offset to the user factor field entry plus its length extends beyond the user factor area 8 – Too many fields specified for the factor 12 – The factor name is too long
8	112	n	A factor field list error has been detected. The reason code is one of the following:

			0 – The offset to the field value plus its length extends beyond the user factor area 4 – The field identifier is not supported 8 – A tag list error has been detected 12 -- An active date error has been detected. The value must be the 7-character keyword “CURRENT” to set the current UTC time, or have a length of zero to clear the existing value in the user profile.
8	116	n	A tag list error has been detected
8	120	n	Supplied buffer is too small. The reason code identifies the buffer length field, which the service updated with the minimum required length. 10 – FACT_GETF_AF_LENGTH 30 – FACT_GETU_UF_LENGTH 50 - FACT_GETP_PA_LENGTH
8	124	n	IBM MFA has detected an error. The reason code is one of the following: 0 – User ID not defined. 1 – User does not have an authentication policy. 2 – Credential List error. Credential type not valid. 3 – Credential List error. Credential length not valid. 4 – Credential List error. Error parsing credential. 5 – Policy name error.
8	128	0	Credentials invalid for user.
8	132	n	IBM MFA has encountered an error extracting MFA data from the user profile. The reason code may be helpful to IBM service.
8	136	n	IBM MFA has encountered a parameter error. The reason code may be helpful to IBM service.
8	140	n	IBM MFA has encountered a parameter error. The reason code may be helpful to IBM service.
8	144	n	IBM MFA has encountered an internal error. The reason code may be helpful to IBM service.
8	148	n	IBM MFA has encountered an unknown error. The reason code may be helpful to IBM service.

Parameter List Example – Get user factor data

The R_factor caller must allocate storage for the areas shown, and must properly set input values in the primary and function-specific parameter lists. The service will return data in the user factor area, as shown in the following example.



4.7.3 R_GenSec (IRRSGS00 or IRRSGS64): Generic security API interface

R_Gensec is updated to add a new PassTicket Subfunction code value:

3 – Evaluate PassTicket Extended

When Subfunction code 3 is provided, instead of receiving a RACF return code of 16 and a RACF reason code of 32 for an unsuccessful PassTicket evaluation, the following will be returned instead:

SAF return code	RACF return code	RACF reason code	Explanation
8	16	X'*nnnnnnnn*'	PassTicket evaluation extended failure. X'*nnnnnnnn*' is the internal reason code for the evaluation failure.

4.7.4 R_TicketServ (IRRSPK00): Parse or extract

The R_TicketServ Ticket_options parameter is updated to add a new value to indicate the PassTicket failure reason code should be returned:

X'00000003' – Evaluate a PassTicket Extended

When Ticket_options 3 is provided, instead of receiving a RACF return code of 16 and a RACF reason code of 32 for an unsuccessful PassTicket evaluation, the following will be returned instead:

SAF return code	RACF return code	RACF reason code	Explanation
8	16	X'*nnnnnnnn*'	PassTicket evaluation extended failure. X'*nnnnnnnn*' is the internal reason code for the evaluation failure.

4.8 z/OS Security Server RACF Macros and Interfaces

This information supplements information in the following chapters and sections:

- Chapter *RACF database unload* in the Record formats produced by the database unload utility section.
- Chapter *SMF records* in the Format of SMF type 80 records section.
- Chapter *The format of the unloaded SMF type data* in the The JOBINIT record extension section.
- Appendix *Supplied class descriptor table entries.*
- Appendix *RACF database templates* in the User template for the RACF database and General template for the RACF database sections.

4.8.1 RACF database unload

1 Record formats produced by the database unload utility

The user basic data record (0200) is extended.

Field Name	Type	Start	End	Comments
USBD_MFA_FALLBACK	Yes/ No	639	641	This user can use a password or password phrase to logon to the system when IBM MFA is unavailable. Valid Values include "Yes" and "No".

The User MFA factor data record (020A) is added.

The User MFA factor data record defines the basic information about the MFA factor data.

Field Name	Type	Start	End	Comments
USMFA_RECORD_TYPE	Int	1	4	Record type of the user Multifactor authentication data record (020A)
USMFA_NAME	Char	6	13	User ID as taken from the profile name.
USMFA_FACTOR_NAME	Char	15	34	Factor name
USMFA_FACTOR_ACTIVE	Date	36	54	Factor active date. Will be blank if factor is not ACTIVE.

The user MFA policies record (020B) is added.

Field Name	Type	Start	End	Comments
USMPOL_RECORD_TYPE	Int	1	4	Record type of the user Multi-factor authentication policies record (020B)

USMPOL_NAME	Char	6	13	User ID as taken from the profile name.
USMPOL_POLICY_NAME	Char	15	34	MFA Policy name.

The User MFA factor tags data record (1210) is added.

The User MFA factor tags data record defines the basic information about the MFA factor tags data.

Field Name	Type	Start	End	Comments
USMFAC_RECORD_TYPE	Int	1	4	Record type of the user Multifactor authentication factor configuration data record (1210)
USMFAC_NAME	Char	6	13	User ID as taken from the profile name.
USMFAC_FACTOR_NAME	Char	15	34	Factor name.
USMFAC_TAG_NAME	Char	36	55	Tag name associated with the factor
USMFAC_TAG_VALUE	Char	57	1080	Tag value associated with the tag name.

The General resource MFA Definition record (05H0) is added.

The General resource MFA definition record defines the basic information about the MFA definition record.

Field Name	Type	Start	End	Comments
GRMFA_RECORD_TYPE	Int	1	4	Record type of the Multifactor Definition data record (05H0)
GRMFA_NAME	Char	6	251	General resource name as taken from the profile name.
GRMFA_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely MFADEF.
GRMFA_FACTOR_DATA_LEN	Int	262	266	Length of factor data.

The general resource MFPOLICY Definition record (05I0) is added.

Field Name	Type	Start	End	Comments
GRMFP_RECORD_TYPE	Int	1	4	Record type of the Multifactor Definition data record (05I0)
GRMFP_NAME	Char	6	251	General resource name as taken from the profile name.
GRMFP_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely MFADEF.
GRMFP_TOKEN_TIMEOUT	Int	262	271	MFA token timeout setting.
GRMFP_REUSE	Yes/ no	273	275	MFA token reuse setting.

The general resource MFA policy factors record (05I1) is added.

Field Name	Type	Start	End	Comments
GRMPF_RECORD_TYPE	Int	1	4	Record type of the user Multifactor authentication policies record (05I1)
GRMPF_NAME	Char	6	251	General resource name as taken from the profile name.
GRMPF_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely MFADEF.
GRMPF_POL_FACTOR	Char	262	281	Policy factor name.

4.8.2 SMF records

Format of SMF type 80 records: data type 6 command-related data

New fields and flags are added for the ALTUSER command.

The SMF Type 80 relocate section 6 (command-related data) format is changing for ALTUSER. There are currently two sets of 'keywords specified/ignored/violated' fields, the first containing 32 bits and the second 16 bits. These are now full. A third set of bit-string

fields will be defined as 32 bits each. A pair of new bits will be defined in each for the MFA and NOMFA keywords.

<table>
<tr><td>13 (D)</td><td>ALTUSER</td><td colspan="4">* The data for event code 13 is identical to the data for event code 10, with these exceptions.</td></tr>
<tr><td colspan="6">:
etc
:</td></tr>
<tr><td rowspan="20"></td><td rowspan="20"></td><td rowspan="10">4</td><td rowspan="10">Binary</td><td colspan="2">Flags for additional keywords specified:</td></tr>
<tr><td>Bit
Byte 0</td><td>Keyword specified</td></tr>
<tr><td>0</td><td>MFA</td></tr>
<tr><td>1</td><td>NOMFA</td></tr>
<tr><td>2-7</td><td>Reserved for IBM's use</td></tr>
<tr><td>Byte 1</td><td></td></tr>
<tr><td>0-7</td><td>Reserved for IBM's use</td></tr>
<tr><td>Byte 2</td><td></td></tr>
<tr><td>0-7</td><td>Reserved for IBM's use</td></tr>
<tr><td>Byte 3</td><td></td></tr>
<tr><td rowspan="10">4</td><td rowspan="10">Binary</td><td colspan="2">Flags for additional keywords ignored (authorization):</td></tr>
<tr><td>Bit
Byte 0</td><td>Keyword specified</td></tr>
<tr><td>0</td><td>MFA</td></tr>
<tr><td>1</td><td>NOMFA</td></tr>
<tr><td>2-7</td><td>Reserved for IBM's use</td></tr>
<tr><td>Byte 1</td><td></td></tr>
<tr><td>0-7</td><td>Reserved for IBM's use</td></tr>
<tr><td>Byte 2</td><td></td></tr>
<tr><td>0-7</td><td>Reserved for IBM's use</td></tr>
<tr><td>Byte 3</td><td></td></tr>
<tr><td rowspan="10">4</td><td rowspan="10">Binary</td><td colspan="2">Flags for additional keywords ignored because of processing error:</td></tr>
<tr><td>Bit
Byte 0</td><td>Keyword specified</td></tr>
<tr><td>0</td><td>MFA</td></tr>
<tr><td>1</td><td>NOMFA</td></tr>
<tr><td>Byte 1</td><td></td></tr>
<tr><td>0-7</td><td>Reserved for IBM's use</td></tr>
</table>

				Byte 2	
				0-7	Reserved for IBM's use
				Byte 3	
				0-7	Reserved for IBM's use
: etc :					

The MFA subkeywords will not have bits defined in these fields. Rather, a new relocate section will be defined to contain he MFA keyword information that was specified on the command. It will be assumed that either all subkeywords will succeed or they will all fail. No partial updates will be made.

Type 80 event code 1 (RACINIT) record:

New event code qualifiers are added for the Type 80 event code 1 (RACINIT) record:

- 40(28) – SUCCESSM - Successful Multifactor authentication
- 41(29) – INVMFA - Failed Multifactor authentication
- 42(2A) – MFAUNAVL - Failed authentication because no multifactor decision could be made for a MFA user who has the NOPWFALLBACK option

43(2B) – MFAPSUCC - IBM MFA partial success: credentials were not incorrect, but a re-authentication is required.

A successful MFA fallback is audited with existing event code qualifiers. Note that a successful MFA fallback authentication will be unconditionally audited (regardless of what the application specified on LOG=, and regardless of whether SETROPTS AUDIT(USER) is active).

<table>
<tr><th>Data type (SMF80TP2) dec(hex)</th><th>Data length (SMF80DL2)</th><th>Format</th><th>Audited by event code</th><th>Description (SMF80DA2)</th></tr>
<tr><td>440(1B8)</td><td>8</td><td>binary</td><td>13</td><td>Byte 1: MFA subkeyword specified flags
<table>
<tr><th>Bit</th><th>Meaning when set</th></tr>
<tr><td>0</td><td>PWFALLBACK specified</td></tr>
<tr><td>1</td><td>NOPWFALLBACK specified</td></tr>
<tr><td>2</td><td>FACTOR specified</td></tr>
<tr><td>3</td><td>DELFACTOR specified</td></tr>
<tr><td>4</td><td>ACTIVE specified</td></tr>
<tr><td>5</td><td>NOACTIVE specified</td></tr>
<tr><td>6</td><td>TAGS specified</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td>
<table>
<tr><td>7</td><td>DELTAGS specified</td></tr>
</table>
Byte 2: MFA subkeyword specified flags
<table>
<tr><th>Bit</th><th>Meaning when set</th></tr>
<tr><td>0</td><td>NOTAGS specified</td></tr>
<tr><td>1</td><td>ADDPOLICY specified</td></tr>
<tr><td>2</td><td>DELPOLICY specified</td></tr>
<tr><td>3-7</td><td>Reserved</td></tr>
</table>
Byte 3-8: Reserved for IBM's use
</td></tr>
<tr><td>441 (1B9)</td><td>variable</td><td>EBCDIC</td><td>13</td><td>Multifactor authentication factor name</td></tr>
<tr><td>442 (1BA)</td><td>variable</td><td>EBCDIC</td><td>13</td><td>MFA tag entry from the TAGS/DELTAGS keyword.
When TAGS is specified, the entry value is the tag name and value separated by a colon (“:”). When DELTAGS is specified, the entry value is the tag name only.</td></tr>
<tr><td>443(1BB)</td><td>variable</td><td>mixed</td><td>1</td><td>Byte 1: Authentication information:
<table>
<tr><th>Bit</th><th>Meaning when set</th></tr>
<tr><td>0</td><td>Authenticated from VLF</td></tr>
<tr><td>1</td><td>User has active MFA factor(s)</td></tr>
<tr><td>2</td><td>MFA user allowed to fall back when no MFA decision can be made</td></tr>
<tr><td>3</td><td>No MFA decision for MFA user</td></tr>
<tr><td>4</td><td>IBMMFA requested that RACROUTE REQUEST=VERIFY return the password-expired return code.</td></tr>
<tr><td>5</td><td>IBM MFA requested that RACROUTE REQUEST=VERIFY return the new-password-invalid return code.</td></tr>
<tr><td>6</td><td>IBM MFA requested that RACROUTE REQUEST=VERIFY return the password-invalid return code, but not to increment the password revoke count (partial</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td>
<table>
<tr><td></td><td>success).</td></tr>
<tr><td>7</td><td>Reserved for IBM's use</td></tr>
</table>

Byte 2: Authenticator used:
<table>
<tr><th>Bit</th><th>Meaning when set</th></tr>
<tr><td>0</td><td>Password Evaluated</td></tr>
<tr><td>1</td><td>Password Successful</td></tr>
<tr><td>2</td><td>Password Phrase Evaluated</td></tr>
<tr><td>3</td><td>Password Phrase Successful</td></tr>
<tr><td>4</td><td>PassTicket Evaluated</td></tr>
<tr><td>5</td><td>PassTicket Successful</td></tr>
<tr><td>6</td><td>MFA authentication successful</td></tr>
<tr><td>7</td><td>MFA authentication unsuccessful</td></tr>
</table>

Byte 3-6: Reason for no MFA decision part 1

Bytes 7-10: Reason for no MFA decision part 2
</td></tr>
<tr><td>444 (1BC)</td><td>variable</td><td>EBCDIC</td><td>13</td><td>MFA policy name entry from the ADDPOLICY/DELPOLICY keyword.</td></tr>
</table>

4.8.3 The format of the unloaded SMF type 80 data

2 The JOBINIT record extension

In the unloaded SMF Type 80 record, the ALTUSER event code D (13 decimal) can have the new MFA keywords appear in the “keywords specified” and “keywords failed” fields.

In addition, the following fields are added to the unloaded JOBINIT record extension, based on information in the new extended relocate section 443 (note that the reason for MFA fallback is not unloaded).

Field Name	Type	Start	End	Comments
INIT_ACEE_VLF	Yes/ No	4540	4543	The ACEE was created from the VLF cache
INIT_MFA_USER	Yes/ No	4545	4548	The user has active MFA factors

INIT_MFA_FALLBACK	Yes/ No	4550	4553	The MFA user is allowed to fall back to password authentication when MFA is unavailable
INIT_MFA_UNAVAIL	Yes/ No	4555	4558	MFA was unavailable to make an authentication decision for the MFA user
INIT_MFA_PWD_EXPIRED	Yes/ No	4560	4563	IBM MFA requested that RACROUTE REQUEST=VERIFY return the password-expired return code
INIT_MFA_NPWD_INV	Yes/ No	4565	4568	IBM MFA requested that RACROUTE REQUEST=VERIFY return the new-password-invalid return code
INIT_MFA_PART_SUCC	Yes/ No	4570	4573	IBM MFA requested that RACROUTE REQUEST=VERIFY return the password-invalid return code, but not to increment the password revoke count (partial success).
INIT_RESERVED_01	Yes/ No	4575	4578	Reserved for IBM's use
INIT_PASSWORD_EVAL	Yes/ No	4580	4583	The supplied password was evaluated
INIT_PASSWORD_SUCC	Yes/ No	4585	4588	The supplied password was evaluated successfully
INIT_PHRASE_EVAL	Yes/ No	4590	4593	The supplied password phrase was evaluated
INIT_PHRASE_SUCC	Yes/ No	4595	4598	The supplied password phrase was evaluated successfully
INIT_PASSTICKET_EVAL	Yes/ No	4600	4603	The supplied password was evaluated as a PassTicket
INIT_PASSTICKET_SUCC	Yes/ No	4605	4608	The supplied password was evaluated successfully as a PassTicket
INIT_MFA_SUCC	Yes/ No	4610	4613	The supplied password phrase/phrase was evaluated successfully as multifactor data
INIT_MFA_FAIL	Yes/ No	4615	4618	The supplied password/phrase was evaluated unsuccessfully as MFA data
INIT_AUTH_RSN1	Char	4620	4627	Authentication reason, part 1. Expressed as hexadecimal number.
INIT_AUTH_RSN2	Char	4629	4636	Authentication reason, part 2. Expressed as hexadecimal number.

4.8.4 RACF database templates

The MFA segment is added to the GENERAL profile and new MFA fields are added to the base segment of the USER profile.

In the USER template:

```
$*  --- The following fields are for Multifactor Authentication support ---
MFAFLBK  109 20 80 00000001 00 MFA - User can fall back to password logon
FACTORN  110 10 80 00000004 00 MFA - Number of defined factors
FACTOR   111 80 80 00000000 00 MFA - Factor name - repeat
FACACDT  112 82 80 00000008 FF MFA - Factor active-on date - repeat'
FACTAGS  113 80 00 00000000 00 MFA - Factor configuration data - repeat'
MFAPOLN  120 10 80 00000004 00 MFA – Number of defined policies
MFAPOLNM 121 80 80 00000000 00 MFA – Policy name – repeat
$*  --- The following are combination fields for the user BASE segment  ---
FACINFO  000 40 00 111 112 113 000 000 *** MFA factor repeat group entry ***
```

In the GENERAL template:

```
$/SEGMENT  018 MFA
MFA      001 00 00 00000000 00 MFA - Start of segment fields
MFDATA   002 00 00 00000000 00 MFA - Free-form factor metadata

$/SEGMENT  019 MFPOLICY
MFPOLICY 001 00 00 00000000 00 MFA - Start of segment fields
MFFCTRN  002 10 00 00000004 00 MFA – Number of factors in policy
MFFCTRS  003 80 00 00000000 00 MFA – Policy Factor list
MFTIMEO  004 00 00 00000004 00 MFA – Policy token timeout
MFREUSE  005 00 00 00000001 00 MFA – Policy reuse setting
```

4.8.5 Class Descriptor Table

A new MFADEF general resource class is added.

ICHERCDE macro keyword
CLASS=MFADEF
POSIT=600
ID=1
PROFDEF=YES
MAXLNTH=246
CASE=UPPER
FIRST=NONATNUM
OTHER=ANY
OPER=NO
KEYQUAL=0
DFTRETC=4
DFTUACC=NONE
RACLIST=ALLOWED
RACLREQ=NO
SIGNAL=NO
GENERIC=ALLOWED
GENLIST=DISALLOWED
SLBLREQ=NO
RVRSMAC=NO
EQUALMAC=NO

4.9 z/OS Security Server RACF Data Areas

This information supplements the information in the *RACF Data Areas* chapter.

4.9.1 RCVT: RACF Communication Vector Table

The RACF Communication Vector Table adds a field to indicate that the MFA Function is available. Other products can check this bit to determine if the current version of RACF has MFA support added either in the base OS or via PTF. A second bit is added to indicate that MFA version 3 functions are available.

Offset (dec)	Offset (Hex)	Type	Len	Name(Dim)	Description
...					
633	279	BITSTRING	1	RCVTFLG3	MISC FLAGS
	…. ..1.	ADDRESS	4	RCVTMFA	MFA Functions are available.
...					
640	280	BITSTRING	1	RCVTFLG4	Function availability bits
	.1..			RCVTMFA3	MFA3 Functions (OA50930) are available.
...					

4.9.2 RIPL: RACROUTE REQUEST=TOKENBLD/VERIFY/VERIFYX Parameter List (Request Section)

The RIPL Table adds a field to indicate that the PASSCHK=NOMFA option was specified.

Offset (dec)	Offset(Hex)	Type	Len	Name(Dim)	Description
...					
58	(3A)	BITSTRING	1	INITFLG3	Miscellaneous Flags
	...1			INITNMFA	PASSCHK=NOMFA was coded
	 1111			*	RESERVED
...					

4.9.3 RIXP: RACROUTE REQUEST=VERIFY/VERIFYX Exit Parameter List

The RIXP table adds a field to indicate that the PASSCHK=NOMFA option was specified.

Offset (dec)	Offset(Hex)	Type	Len	Name(Dim)	Description
...					
200	(CE)	ADDRESS	4	RIXFLAG3	Flag byte address: points to a 1-byte area of the following format:
	1... ….			RACPNMFA	PASSCHK=NOMFA was coded
	.111 1111			*	RESERVED
...					

4.9.4 ACEE (IHAACEE)

The ACEE is updated to add two new flags:

- ACEEMFAU is an indicator that the user must authenticate with MFA. This indicator is on when the user has an active MFA factor and the MFADEF class is active.
- ACEEMFAA is an indicator that the user was authenticated with MFA.

Offset (dec)	Offset(Hex)	Type	Len	Name(Dim)	Description
...					
134	86	BITSTRING	1	ACEEFLG6	More MISC FLAGS
	...1 ….			ACEEMFAU	User must authenticate with MFA. On when the user has an active MFA factor and MFADEF class is active.
	…. 1...			ACEEMFAA	User authenticated with MFA.
	…. .111			*	Reserved
...					

5 Trademarks

IBM®, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.