RSA Ready Implementation Guide for RSA SecurID

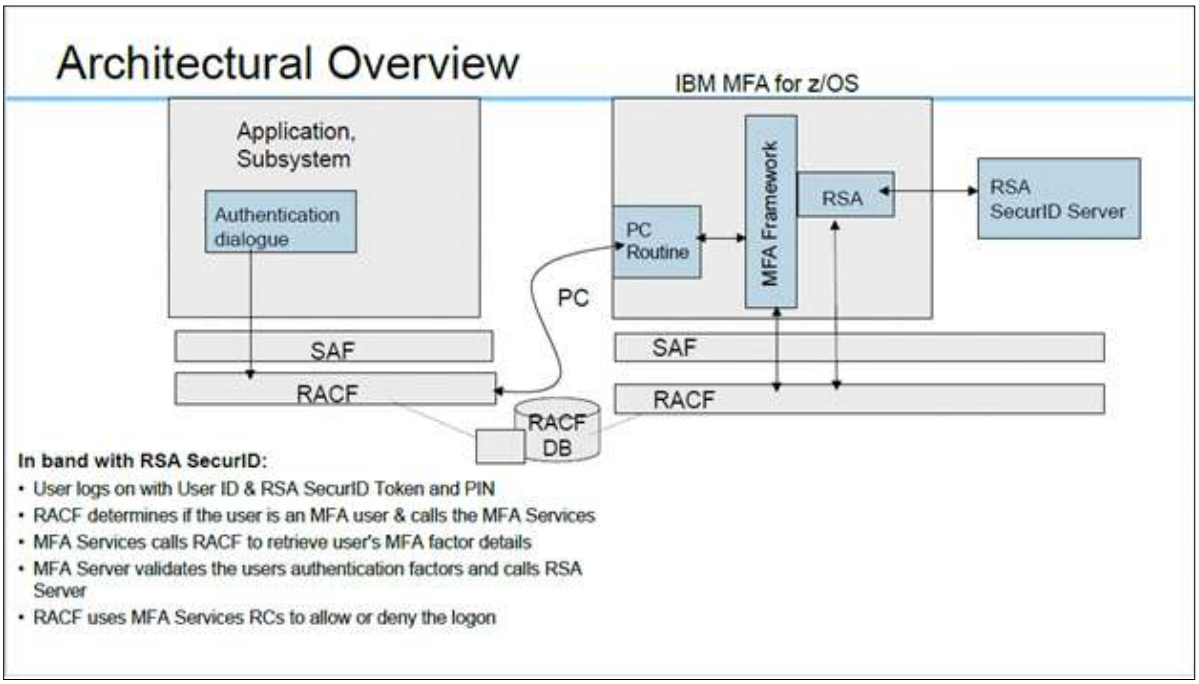
IBM Multi-Factor Authentication for z/OS V1R1

John Sammon, RSA Partner Engineering
Last Modified: 4/7/16

Solution Summary

IBM Multi-Factor Authentication for z/OS, which is referred to in this document as IBM MFA, provides an alternate authentication mechanism for z/OS networks that are used in conjunction with RSA SecurID-based authentication systems. IBM MFA allows RACF to use RSA SecurID authentication mechanisms in place of the standard z/OS password. 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. Clients are looking for ways to raise the assurance level of their systems by requiring additional authentication factors for users.

RSA Authentication Manager supported features IBM Multi-Factor Authentication for z/OS V1R1	
RSA SecurID Authentication via Native RSA SecurID UDP Protocol	YES
RSA SecurID Authentication via Native RSA SecurID TCP Protocol	NO
RSA SecurID Authentication via RADIUS Protocol	NO
RSA SecurID Authentication via IPv6	NO
On-Demand Authentication via Native SecurID UDP Protocol	YES
Risk-Based Authentication	N/A
RSA Authentication Manager Replica Support	YES
RSA SecurID Software Token Automation	N/A
RSA SecurID SD800 Token Automation	N/A
RSA SecurID Protection of Administrative Interface	N/A



RSA Authentication Manager Configuration

Agent Host Configuration

To facilitate communication between the IBM Multi-Factor Authentication for z/OS and the RSA Authentication Manager / RSA SecurID Appliance, an Agent Host record must be added to the RSA Authentication Manager database. The Agent Host record identifies the IBM Multi-Factor Authentication for z/OS and contains information about communication and encryption.

Include the following information when configuring a UDP-based agent host record.

- Hostname
- IP addresses for network interfaces

> **! Important: The UDP-based authentication agent's hostname must resolve to the IP address specified.**

Set the Agent Type to "Standard Agent" when adding the Authentication Agent. This setting is used by the RSA Authentication Manager to determine how communication with IBM Multi-Factor Authentication for z/OS will occur.

Partner Product Configuration

Before You Begin

All IBM Multi-Factor Authentication for z/OS components must be installed and working prior to the integration. Perform the necessary tests to confirm that this is true before proceeding. For installation instructions see the *IBM Multi-Factor Authentication for z/OS Program Directory*, which is included in the product package.

IBM Multi-Factor Authentication for z/OS Configuration

System Programming Steps

Complete the following customization steps to tailor IBM Multi-Factor Authentication for z/OS (MFA) for the environment:

- Copy S*AZFEXEC* (*AZFEXEC*)
- Customize *AZFEXEC*
- Copy SAZFSAMP (AZF#IN00)
- Customize AZF#IN00
- Authorize the Load Library
- Update SCHEDxx PARMLIB program properties

Copy SAZFEXEC (AZFEXEC)

Copy the S*AZFEXEC* (*AZFEXEC*) member to a data set in your *SYSEXEC* concatenation as follows:

1. Browse the S*AZFEXEC* data set in the target library.
2. Copy the member *AZFEXEC* to a data set in your *SYSEXEC* concatenation. You can use the *TSO ISRDDN* command from *ISPF* to view the current data set allocations, including the *SYSEXEC* concatenation.
3. Verify the change.

Customize *AZFEXEC*

Customize the *azfhlq* parameter of the *AZFEXEC* member of the data set as follows:

1. Edit the *AZFEXEC* member of the data set in your *SYSEXEC* concatenation.
2. Change the *azfhlq* parameter to the high-level qualifier (HLQ) used where you installed IBM MFA.
3. Save the change.

Copy SAZFSAMP (AZF#IN00)

Follow these steps to copy *AZF#IN00* to the *PROCLIB* from which you run started tasks.

1. Copy the *AZF#IN00* member of the *SAZFSAMP* data set in the target library to the *PROCLIB* from which you run started tasks.
2. Browse the *PROCLIB* to ensure the *AZF#IN00* member is there.

Customize AZF#IN00

Customize *AZF#IN00* for the high-level qualifier as follows:

1. Edit *AZF#IN00* and change *azfhlq* to the high-level qualifier of where you installed IBM MFA.
2. Save the change.

Authorize the Load Library

Make sure the load library containing the IBM MFA load modules is *APF* authorized. The *PROGxx parmlib* member contains the names of program libraries that you want the system to define as authorized with the Authorized Program Facility (APF). The *APF* statement defines the format and contents of the *APF* list.

1. Add the following line to the *APF* section of your *PROGxx* parmlib member:

   APF ADD DSNAME(HLQ.SAZFLOAD) SMS

   where *HLQ* is the high-level qualifier used where you installed IBM MFA.

2. Save the change.

Update SCHEDxx PARMLIB program properties

Update the *SCHEDxx parmlib* properties to identify the program, *AZFSTCMN*, that requires special attributes. The PPT statement specifies a list of programs that require special attributes.

Follow these steps to update the properties:

1. Edit the *SYS1.PARMLIB(SCHEDxx)* member that defines program properties.
2. Add the following entry:

```
PPT PGMNAME (AZFSTCMN)     /* MULTI-FACTOR AUTH */
KEY(2)                      /* PROTECTION KEY */
NOSWAP                      /* NON-SWAPABLE */
CANCEL                     /* CANCELABLE */
```

3. Save the changes.

RACF administration steps

After you complete the system programming steps, perform the following RACF administration steps.

- Define a user for AZF started task
- Define entry in STARTED Class
- Activate MFADEF class
- Define factors in MFADEF class
- Define factors in FACILITY class
- Authorize access to IRR.RFACTOR.MFADEF.AZFSIDP1 profile

Define a user for AZF started task

1. Define a user for the *AZF* started task with the following properties:
   - No passphrase or password
   - Owned by a suitable started task group
   - PROTECTED
   - No TSO segment
   - An OMVS segment with a unique user ID

For example:

```
USER=AZFSTC NAME=STCFORMFA OWNER=STCGROUPCREATED=15.257
DEFAULT-GROUP=STCGROUP PASSDATE=N/A PASS-INTERVAL=N/A
PHRASEDATE=N/A
ATTRIBUTES=PROTECTED
REVOKEDATE=NONE RESUMEDATE=NONE
LAST-ACCESS=15.282/13:36:54
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED (DAYS) (TIME)
---------------------------------------------
ANYDAY ANYTIME
GROUP=STCGROUP AUTH=USE CONNECT-OWNER=STCGROUP CONNECT-DATE=15.257
CONNECTS=123 UACC=NONE LAST-CONNECT=15.282/13:36:54
CONNECT ATTRIBUTES=GRPACC
REVOKE DATE=NONE RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
NO TSO INFORMATION
OMVS INFORMATION
----------
UID= 0000015100
CPUTIMEMAX= NONE
ASSIZEMAX= NONE
FILEPROCMAX= NONE
PROCUSERMAX= NONE
THREADSMAX= NONE
MMAPAREAMAX= NONE
```

2. Save the change.
3. To verify the user information, you can use a command such as the following:

   LU AZFSTC OMVS

Define entry in STARTED Class

Follow these steps to define an entry in the RACF STARTED class to ensure that the IBM MFA address space has the proper level of authority:

1. Define an entry in the STARTED class. For example:

CLASS	NAME	LEVEL	OWNER	UNIVERSAL ACCESS	YOUR ACCESS	WARNING	INSTALLATION DATA
STARTED	AZF*.** (G)	00	STCGROUP	NONE	NONE	NO	NONE

APPLICATION DATA	AUDITING	NOTIFY	STDATA INFORMATION
NONE	FAILURES(READ)	NO USER TO BE NOTIFIED	USER= AZFSTC GROUP= STCGROUP TRUSTED= NO PRIVILEGED= NO TRACE= NO

2. Verify the change.

Activate MFADEF class

Follow the steps below to activate the *MFADEF* class if it hasn't been activated. The *MFADEF* class must be active before a user can log on with IBM MFA.

1. Activate the *MFADEF* Class:

   SETROPTS CLASSACT(MFADEF)

2. Verify the change.

Define factors in MFADEF class

Use *RDEFINE* to define the authentication factors in the *MFADEF* class. You define MFA factors by creating an *MFADEF* class profile named *FACTOR.AZFSIDP1*.

1. Define the factors in the MFADEF class:

   RDEF MFADEF FACTOR.AZFSIDP1

2. Verify the change. For example:

   RLIST MFADEF FACTOR.AZFSIDP1 MFA

Define factors in FACILITY class

Use *RDEFINE* to define the authentication factors in the class. You define MFA factors by creating a *FACILITY* class profile named *IRR.RFACTOR.MFADEF.AZFSIDP1*.

1. Define the factors in the *FACILITY* class:

   RDEF FACILITY IRR.RFACTOR.MFADEF.AZFSIDP1

2. Verify the change. For example:

   RLIST FACILITY IRR.RFACTOR.MFADEF.AZFSIDP1

Authorize access to IRR.RFACTOR.MFADEF.AZFSIDP1 profile

Authorize the RSA Authentication Manager administrators who execute the panels to the *IRR.RFACTOR.MFADEF.AZFSIDP1* profile.

1. Allow the following access:

Permission	Access
READ	Able to view configuration options, but may not update, create, or delete SecurID parameters.
UPDATE	Able to view and update configuration options, but may not create or delete SecurID parameters.
CONTROL	Able to view and update configuration options, but may not create or delete SecurID parameters.
ALTER	Able to create, update, delete, and view configuration options.

2. Save the change.

Additional system programming steps

Perform the following additional system programming tasks:

- Allocate SDCONF.REC data set
- Allocate node secret data set
- Copy sdconf.rec to SDCONF.REC data set
- Create SDOPTS.REC file (Optional)
- Define SecurID parameters for data sets
- Start the started task

Allocate SDCONF.REC data set

Allocate the *SDCONF.REC* data set as follows

1. Allocate the *SDCONF.REC* data set with the following attributes. The User ID under which the started task runs must have read access to this data set.

   DCB:
   - *RECFM FB*
   - *LRECL 3072*
   - *BLKSIZE 3072*

   SPACE:
   - *BLKS*
   - *Primary 1*
   - *Secondary 1*

2. Verify the change.

Allocate node secret data set

You must allocate the node secret data set. The RSA node secret is a shared secret known to IBM MFA and the RSA Authentication Manager.

1. Allocate the node secret data set with the following attributes. The User ID under which the started task runs must have read access to this data set.

   DCB:
   - *RECFM FB*
   - *LRECL 72*
   - *BLKSIZE 72*

   SPACE:
   - *TRKS*
   - *Primary 1*
   - *Secondary 1*

2. Verify the change.

Copy sdconf.rec to SDCONF.REC data set

1. Download the *sdconf.rec* file from the RSA Authentication Manager Security Console.
2. Use your tool of choice to copy *sdconf.rec* in to the *SDCONF.REC* data set on the z/OS system. Copy the file in binary mode.

Create SDOPTS.REC file (Optional)

In some environments, the host IP the *AZFSIDP1* plug-in detects won't match the actual IP address used for outgoing traffic. In such cases, use the *sdopts.rec* file's *CLIENT_IP* override parameter to manually specify the IP Address that *AZFSIDP1* should use. (Currently, *sdopts.rec* only supports IPV4 addresses.)

The *sdopts.rec* file adheres to the following syntax:

- *LPAR_NAME=<SYSTEM/LPAR NAME>*
- *CLIENT_IP=<IP v4 Address Override>*
- *LPAR_NAME=<SYSTEM/LPAR NAME of another system in SYSPLEX>*
- *CLIENT_IP=<IP v4 Address Override for second LPAR>*

Follow the steps below if you want to allocate the SDOPTS.REC data set.

1. Allocate the SDOPTS.REC data set with the following attributes. The User ID under which the started task runs must have read access to this data set.

   DCB:
   - *RECFM FB*
   - *LRECL 72*
   - *BLKSIZE 72*

   SPACE:
   - *TRKS*
   - *Primary 1*
   - *Secondary 1*

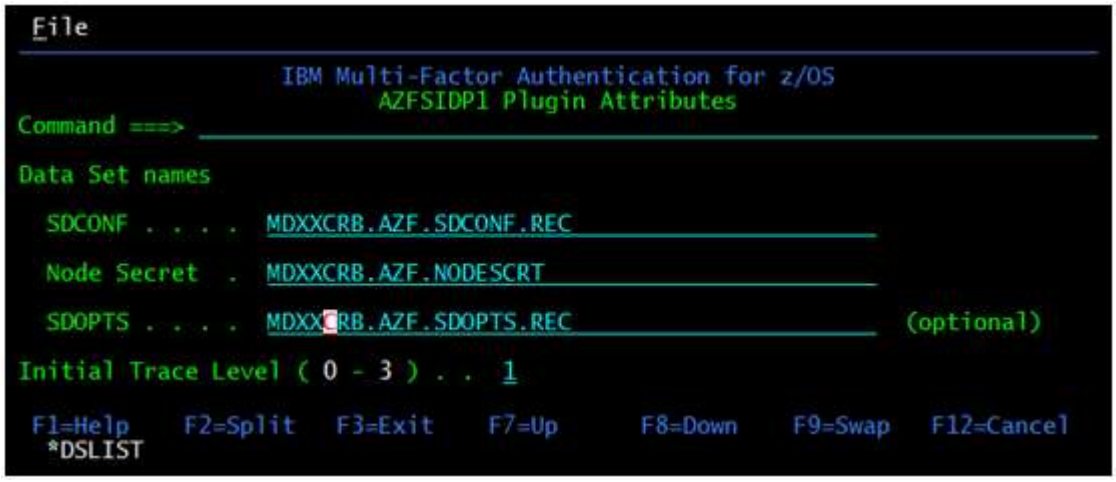
2. Verify the change.
3. Create the *sdopts.rec* file with the required parameters.
4. Save your changes.

Define SecurID parameters for data sets

Execute *AZFEXEC* to define the RSA SecurID parameters for the data set names of the *SDCONF.REC*, node secret, and optional *SDOPTS.REC* data sets.

Note: You must have *ALTER* access to *the FACTOR.AZFSIDP1* profile in the *MFADEF* class to create or delete factor-wide settings for the *AZFSIDP1* Authenticator. Make sure you performed the Copy SAZFEXEC (AZFEXEC) and Customize AZFEXEC system programming steps before proceeding.

1. Execute *AZFEXEC*. Output similar to the following is displayed:



2. Verify the changes.

Start the started task

After you have defined the SecurID parameters for the data sets, start the started task.

1. To start the started task, enter the following operator command:

   S <STC Job Name>

   For example, you might use

   s AZF#IN00

2. Verify that the task started.

Administration and operation steps

Follow the steps in this section to provision users and start up and administer IBM MFA. You need to configure an RSA Authentication Agent for each z/OS system or LPAR that is running IBM Multi-Factor Authentication for z/OS. See your Authentication Manager documentation for details.

- Before You Begin
- Activate and deactivate users for IBM MFA
- Start the started task
- Shut down the started task
- Clear the node secret
- Modify component trace levels

Before you begin

Before you can activate users for IBM MFA, you must first create accounts for the users in RSA Authentication Manager and assign RSA tokens. When you activate a user for IBM Multi-Factor Authentication for z/OS, that user is no longer able to use the z/OS password to log in. Therefore, the user must first have a valid token and credentials for RSA Authentication Manager.

To defer activation to a later time, omit the *ACTIVE* keyword from the *ALTUSER* command, or supply the *NOACTIVE* keyword to deactivate the authenticator for the user ID.

Activate and deactivate users for IBM MFA

Use the *ALTUSER* or *ALU* command to activate users for IBM MFA as follows:

1. Enter the following command to activate a user for IBM MFA:

   ALU [Login ID] MFA (FACTOR (AZFSIDP1)
   ACTIVE PWFALLBACK TAGS (SIDUSERID:[RSA User ID]))

   Where:

   - *[Login ID]* is the z/OS user name.
   - *ACTIVE* activates the AZFSIDP1 authenticator for the user ID.
   - *PWFALLBACK* configures password fallback for the user. If you configure user accounts with the password fallback parameter, users can log in with their z/OS password if the RSA Authentication Manager or IBM MFA server are down. The password fallback mechanism is provided as a fail safe authentication method. If you omit this parameter, the default is *NOPWFALLBACK*.
   - *RSA User ID* is the associated RSA user ID.

2. Enter the following command to defer activating a user for IBM MFA:

   ALU [Login ID] MFA (FACTOR (AZFSIDP1)
   TAGS (SIDUSERID:[RSA User ID]))

   Then, at a later time, enter an *ALTUSER* or *ALU* command of the following form to activate the *AZFSIDP1* authenticator for the user ID:

   ALU <USERID> MFA (FACTOR (AZFSIDP1) ACTIVE)

3. Enter the following command to deactivate a user for IBM MFA:

   ALU [Login ID] MFA (FACTOR (AZFSIDP1) NOACTIVE TAGS (SIDUSERID:[RSA User ID]))

4. Enter the following command to display IBM MFA information for a user profile:

```
LISTUSER [Login ID] MFA
MULTIFACTOR AUTHENTICATION INFORMATION:
---------------------------------------
PASSWORD FALLBACK IS NOT ALLOWED
FACTOR = AZFSIDP1
STATUS = ACTIVE
FACTOR TAGS =
SIDUSERID:user
```

Start the started task

The IBM MFA started task supports the start operation at runtime.

1. To start the started task, enter the following operator command:

   S <STC Job Name>

   For example, you might use

   S AZF#IN00

2. Verify that the task started.

Shut down the started task

The IBM MFA started task supports the stop operation at runtime.

1. To stop the started task, enter the following operator command:

   P <STC Job Name>

   For example, you might use

   P AZF#IN00

2. Verify that the task stopped.

Clear the node secret

The RSA node secret is a shared secret known to each *AZFSIDP1* Authenticator instance (one per host or LPAR where IBM MFA is installed) and the RSA Authentication Manager. If this secret must be established (or re-established), your RSA Authentication Manager administrator will request that the node secret be cleared from each z/OS client host.

Follow these steps to clear the node secret:

1. Enter the following a Modify command:

   F <STC Job Name>,AZFSIDP1 CLEAR NODE SECRET

   For example:

   F AZF#IN00,AZFSIDP1 CLEAR NODE SECRET

2. Repeat step1 on each host or LPAR where IBM MFA is installed
3. Verify that the RSA Authentication Manager generates a new node secret on the first successful login after the previous secret has been cleared.

Modify component trace levels

The IBM MFA started task supports modifying trace levels on a per-component basis at runtime. The available trace levels are as follows:

0	Only standard or unconditional message are output.
1	All output from level 0 plus major items of interest.
2	All output from level 1 plus lesser items of interest.
3	All trace information (Verbose).

Follow these steps to change trace levels on a per-component basis:

1. Issue a **Modify** command of the following form:

   F <STC Job Name>,<Component> SET TRACE LEVEL <Trace Level>

   where *component* can be one of the following two literal values:

   - *STC, which* represents the started task (excluding PC routine.)
   - *AZFSIDP1*, which represents the AZFSIDP1 authenticator that supports RSA SecurID.

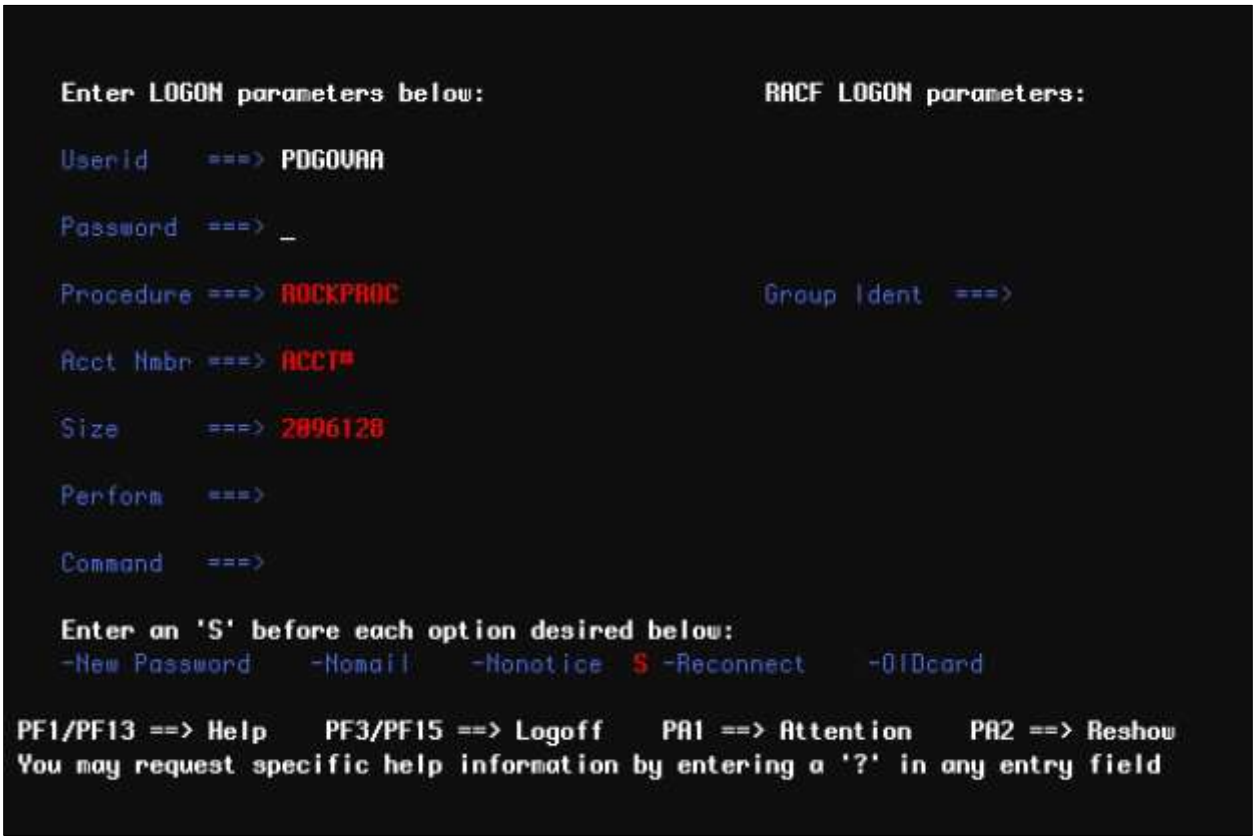
   For example:

   F AZF#IN00, AZFSIDP1 SET TRACE LEVEL 1

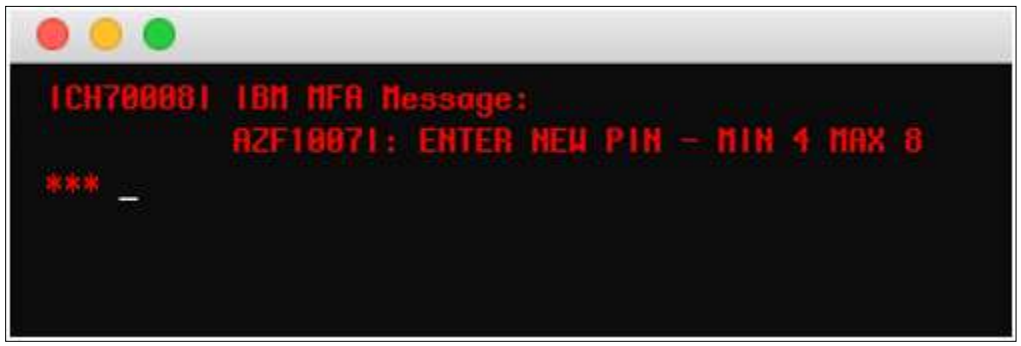
2. Verify that the log has the expected output.

RSA SecurID Login Screens

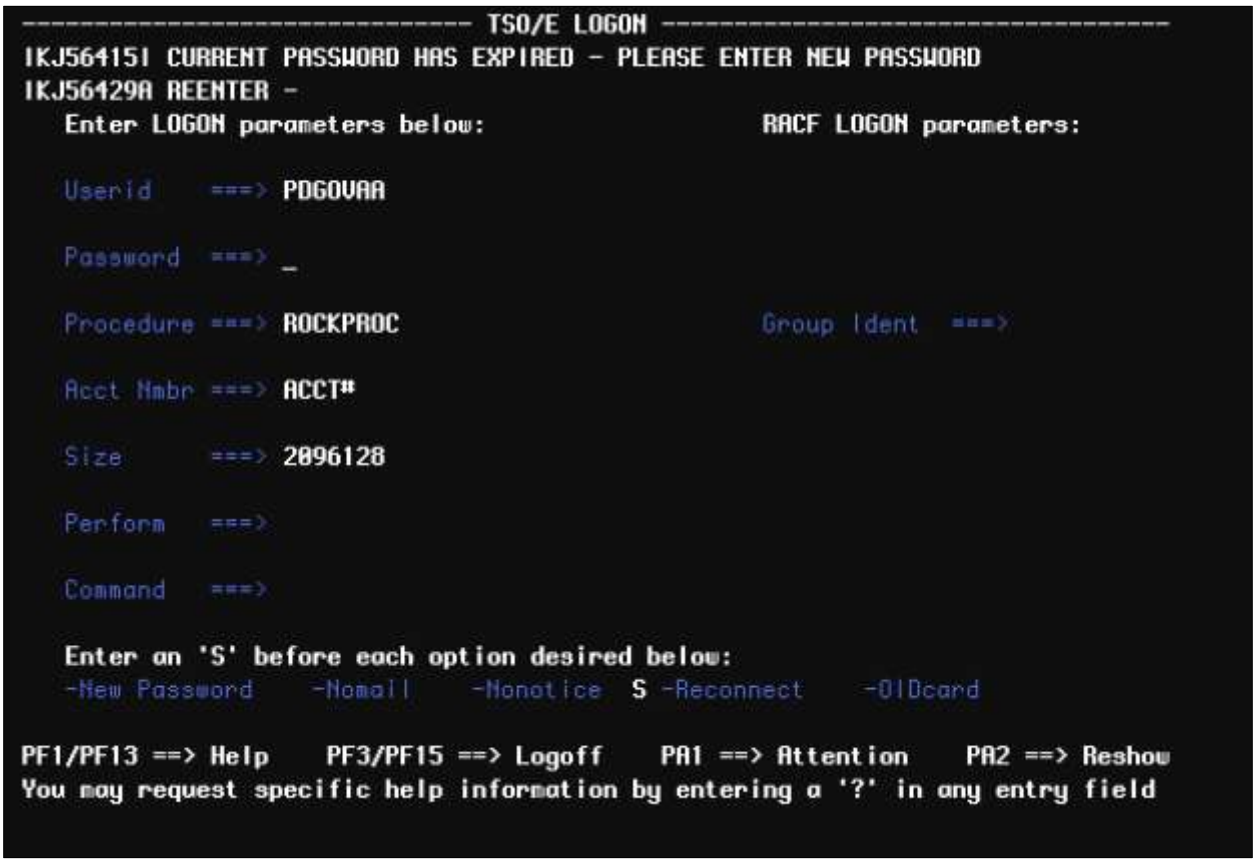
Login screen:



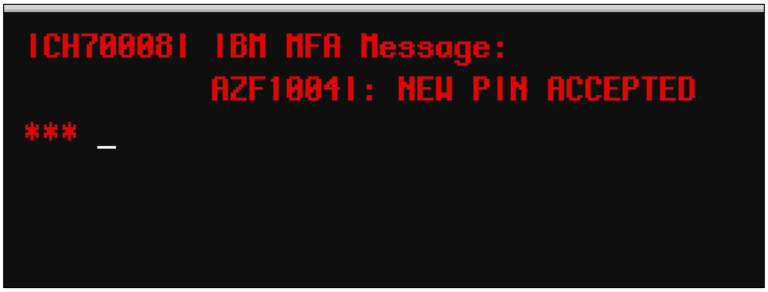
User-defined New PIN Screen 1:



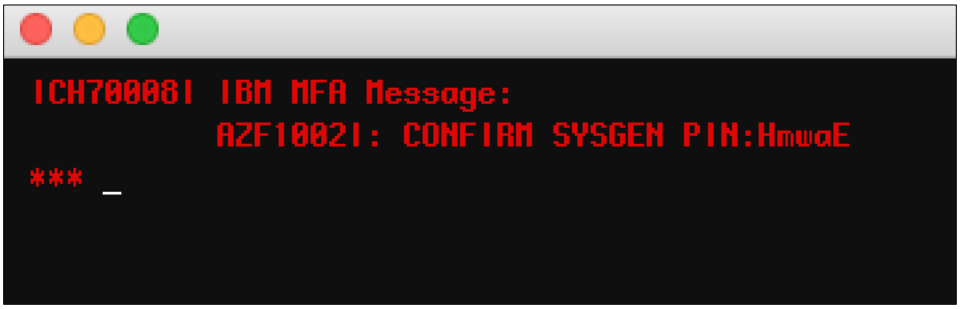
User-defined New PIN Screen 2:



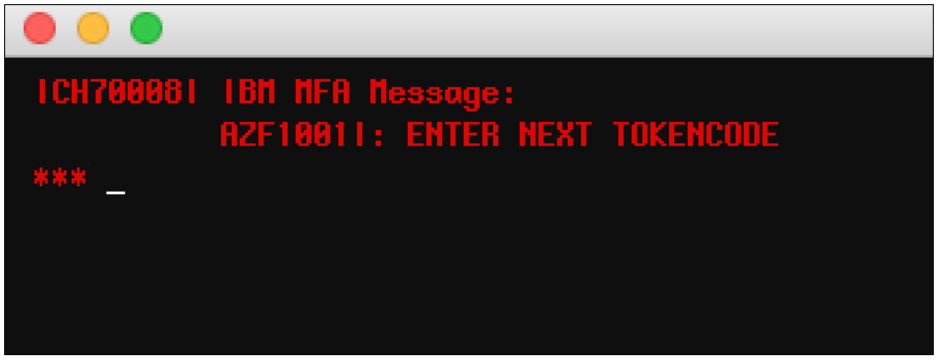
User-defined New PIN Screen 3:



System-generated New PIN Screen:



Next Tokencode Screen:



Certification Checklist for RSA Authentication Manager

Date Tested: March 16, 2016

Certification Environment		
Product Name	**Version Information**	**Operating System**
RSA Authentication Manager	8.1.1	Virtual Appliance
RSA Authentication Agent	N/A	N/A
IBM Multi-Factor Authentication for z/OS	V1R1	V1R1

RSA SecurID Authentication

Date Tested: March 16, 2016

Mandatory Functionality	Native UDP	Native TCP	RADIUS Client
New PIN Mode			
Force Authentication After New PIN	✓	N/A	N/A
System Generated PIN	✓	N/A	N/A
User Defined (4-8 Alphanumeric)	✓	N/A	N/A
User Defined (5-7 Numeric)	✓	N/A	N/A
Deny 4 and 8 Digit PIN	✓	N/A	N/A
Deny Alphanumeric PIN	✓	N/A	N/A
Deny PIN Reuse	✓	N/A	N/A
Passcode			
16 Digit Passcode	✓	N/A	N/A
4 Digit Fixed Passcode	✓	N/A	N/A
Next Tokencode Mode			
Next Tokencode Mode	✓	N/A	N/A
On-Demand Authentication			
On-Demand Authentication	✓	N/A	N/A
On-Demand New PIN	✓	N/A	N/A
Load Balancing / Reliability Testing			
Failover (3-10 Replicas)	✓	N/A	N/A
No RSA Authentication Manager	✓	N/A	N/A

✓ = Pass ✗ = Fail N/A = Non-Available Function

Appendix

RSA SecurID Authentication Files

RSA SecurID Authentication Files	
UDP Agent Files	**Location**
sdconf.rec	Location must be **specified in configuration panels**
sdopts.rec	Optional. Location must be **specified in configuration panels**
Node secret	Location must be **specified in configuration panels**
sdstatus.12 / jastatus.12	N/A
TCP Agent Files	**Location**
rsa_api.properties	N/A
sdconf.rec	N/A
sdopts.rec	N/A
Node secret	N/A

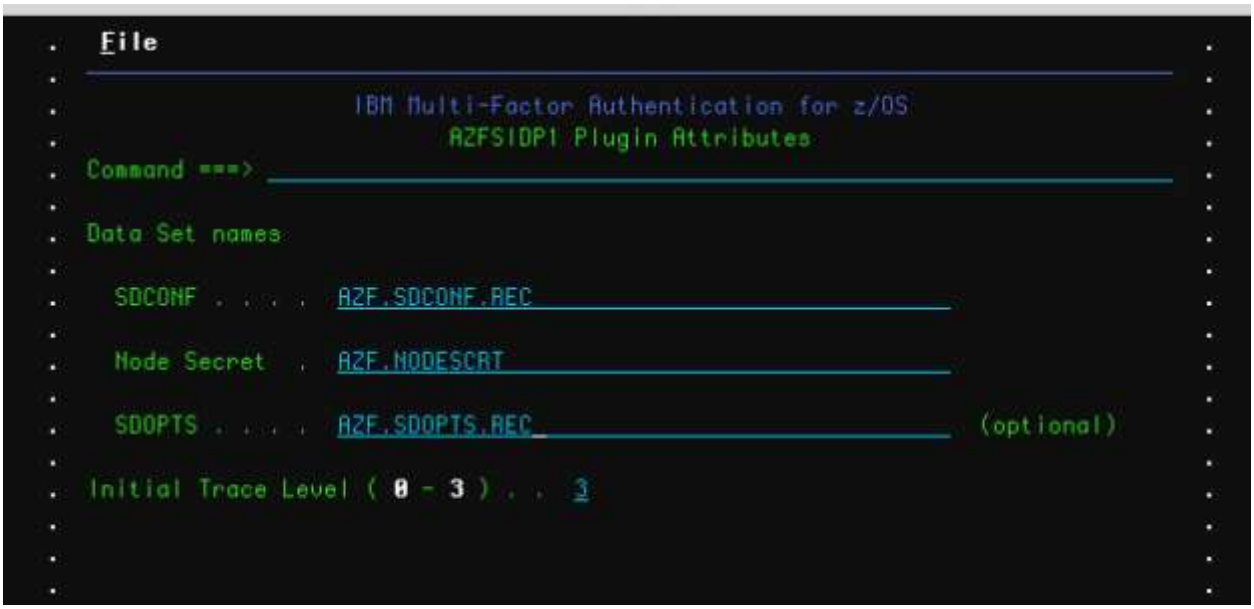
Partner Integration Details

Partner Integration Details	
RSA SecurID UDP API	8.1.3
RSA Authentication Agent Type	Standard Agent
RSA SecurID User Specification	Designated Users
Display RSA Server Info	No
Perform Test Authentication	No
Agent Tracing	Yes

Agent Tracing:

Tracing can be set by issuing the following console command to the AZF started task:

F AZFSTC,AZFSIDP1 SET TRACE LEVEL 0/1/2/3



See the Modify component trace levels section for more information.