RSA SECURID® ACCESS
Implementation Guide

CA
CA-ACF2 for z/OS

RSA Partner Engineering
Last Modified: February 28, 2018

RSA
READY

Solution Summary

CA-ACF2 (the ACF stands for Access Control Facility) is a set of programs from Computer Associates that enable security on mainframes. ACF2 prevents accidental or deliberate modification, corruption, mutilation, deletion, or viral infection of files. With ACF2, access to a system is denied to unauthorized personnel. Any authorized or unauthorized attempt to gain access is logged. System status can be monitored on a continuous basis, and a permanent usage log can be created. The logging feature, besides helping to identify potential intruders, makes it possible to identify and analyze changes and trends in the use of the system. Settings can be changed on a moment's notice, according to current or anticipated changes in the security or business requirements of the organization using the system.

The CA-ACF2 integration with RSA Authentication Manager introduces an extra level of security by enabling RSA SecurID two-factor authentication for CA-ACF2 users.

RSA SecurID Access Features	
CA-ACF2	
On Premise Methods	
RSA SecurID	✓
On Demand Authentication	✓
Risk-Based Authentication (AM)	-
Cloud Authentication Service Methods	
Authenticate App	-
FIDO Token	-
SSO	
SAML SSO	-
HFED SSO	-

Identity Assurance	
Collect Device Assurance and User Behavior	-

Supported Authentication Methods by Integration Point

This section indicates which authentication methods are supported by integration point. The next section (Configuration Summary) contains links to the appropriate configuration sections for each integration point.

CA-ACF2 Password Manager Pro integration with RSA Cloud Authentication Service

Authentication Methods	REST	IDR SAML	Cloud SAML	HFED	RADIUS
RSA SecurID	-	-	-	-	-
LDAP Password	-	-	-	-	-
Authenticate Approve	-	-	-	-	-
Authenticate Tokencode	-	-	-	-	-
Device Biometrics	-	-	-	-	-
SMS Tokencode	-	-	-	-	
Voice Tokencode	-	-	-	-	
FIDO Token		-	-	-	

CA-ACF2 Password Manager Pro integration with RSA Authentication Manager

Authentication Methods	REST	RADIUS	UDP Agent	TCP Agent
RSA SecurID	-	-	✓	-
AM RBA		-	-	

✓	Supported
-	Not supported
n/t	Not yet tested or documented, but may be possible

Configuration Summary

All of the supported use cases of RSA SecurID Access with CA-ACF2 require both server-side and client-side configuration changes. This section of the guide includes links to the appropriate sections for configuring both sides for each use case.

RSA Authentication Manager – CA-ACF2 can be integrated with RSA Authentication Manager in the following way:

UDP Agent

Authentication Manager UDP Agent Configuration
CA-ACF2 UDP Agent Configuration

RSA SecurID Access Configuration

RSA Authentication Manager Configuration

UDP Agent

To configure your RSA Authentication Manager for use with a UDP-based agent, you must create an agent host record in the Security console of your Authentication Manager and download its configuration file (*sdconf.rec*).

- Hostname: Configure the agent host record name to match the z/OS system hostname.
- IP Address: Configure the agent host record to match the z/OS system IP address.

! **Important**: Authentication Manager must be able to resolve the IP address from the hostname.

Partner Product Configuration

Before You Begin

This section provides instructions for configuring the CA-ACF2 with RSA SecurID Access. This document is not intended to suggest optimum installations or configurations. You should be familiar with RSA Authentication Manager, z/OS and CA-ACF2. You must install and configure Advanced Authentication Mainframe prior to configuring the integration. Perform the necessary tests to confirm it has been configured correctly before proceeding.

! **Important**: Advanced Authentication Mainframe installation and configuration are outside of the scope of this document. See the CA ACF2™ for z/OS - 16.0 *Install and Configure Advanced Authentication Mainframe* document for instructions (https://docops.ca.com/ca-acf2-for-z-os/16-0/en/installing-and-implementing/install-and-configure-advanced-authentication-mainframe)

You will need following files before you continue:

- the RSA Authentication Agent API 8.1 for Java[*] jar files (*authapi.jar* and *cryptoj.jar*) and configuration properties file (*rsa_api.properties*).
- your RSA Authentication Manager server's *sdconf.rec* and *failover.dat* files.

CA-ACF2 UDP Agent Configuration

The instructions in this guide use the following configuration values:

%MFASTC%	The OMVS directory containing all OMVS files that are needed to run Advanced Authentication Mainframe. You will choose this directory when you install and configure Advanced Authentication Mainframe.
%RSA_API_HOME%	The directory designated to contain the RSA Authentication Agent API JARS. You can put *%RSA_API_HOME%* under the *%MFASTC%* directory, but you are not required to do so. You may use any OMVS directory as long as the MFASTC started task user ID has access to it.
%RSA_API_HOME%/props	A *%RSA_API_HOME%* sub directory that will contain the *rsa_api.properties* file.
%RSA_AGENT_NAME%	The name of the RSA Authentication Agent you created for the system.

! **Important**: Make sure the *MAINARGS DD* in the Advanced Authentication Mainframe started task JCL also points to *%MFASTC%/props*. See the *CA ACF2™ for z/OS - 16.0 Install and Configure Advanced Authentication Mainframe* document for instructions.

[*] If you don't have a copy of the RSA Authentication Agent API 8.1 for Java, contact your RSA representative.

Configure RSA Securid Authentication Support

Perform these tasks to configure support that allows users to log on to z/OS applications by using RSA SecurID credentials (instead of a CA ACF2 password/password phrase).

1. Log in to the z/OS system and create the %*MFASTC*%, %RSA_API_HOME% *and* %RSA_API_HOME% */props* directories.
2. *Navigate to the %MFASTC%* directory, create a file with the content below and save it as *sdopts.rec.* Replace *<HOST_IP>* with the IP address of the z/OS system.

```
CLIENT_IP=<HOST_IP>
```

3. Transfer the *sdconf.rec* and *failover.dat* files to the z/OS system in binary mode and copy them to %*MFASTC*%.
4. Transfer the *authapi.jar* and *crypto.jar* jar files to the z/OS system in binary mode and copy them to *%RSA_API_HOME%* .
5. Transfer the *rsa_api.properties* file to the z/OS system and copy it to %RSA_API_HOME% */props*
6. Edit the file and set the following properties. (See the **Appendix** for additional suggested values.):

Property	Value
RSA_AGENT_HOST	*%RSA_AGENT_NAME%*
RSA_LOG_LEVEL	*INFO*
RSA_LOG_TO_CONSOLE	*YES*
RSA_LOG_TO_FILE	*NO*
RSA_ENABLE_DEBUG	*NO*
RSA_DEBUG_TO_CONSOLE	*YES*
RSA_DEBUG_TO_FILE	*YES*
RSA_DEBUG_FILE	*%MFASTC%/rsa_api_debug.log*
SDCONF_LOC	*%MFASTC%/sdconf.rec*
SDNDSCRT_LOC	*%MFASTC%/*securid
SDSTATUS_LOC	*%MFASTC%/JAStatus.1*
SDOPTS_LOC	*%MFASTC%/sdopts.rec*

Set Up Control Over RSA SecurID Authentication

Control Whether Users Sign On Through Multi-Factor Authentication

Note: The instructions and examples in this section are from the *Control Whether Users Sign On Through Multi-Factor Authentication* section of the CA ACF2™ for z/OS - 16.0 Administration guide https://docops.ca.com/ca-acf2-for-z-os/16-0/en/administrating/enable-multi-factor-authentication/control-whether-users-sign-on-through-multi-factor-authentication.

Consult this guide for additional configuration options and instructions.

After you have installed and configured Advanced Authentication Mainframe, you can set up a resource rule in the *CASECMFA* class and control whether users must log on by using RSA SecurID credentials.

1. (Optional) Change the default *CAS* type code by inserting a new *GSO CLASMAP* record for the *CASECMFA* class and specifying the desired three-character type code. For example, if you want the code to be *XYZ*, issue the following commands before defining the rule or INFODIR record:

```
SET CONTROL(GSO)
INSERT CLASMAP.CASECMFA ENTITYLN(39) RESOURCE(CASECMFA) RSRCTYPE(XYZ)
F ACF2,REFRESH(CLASMAP)
```

2. Make the rule resident by issuing the following command (Required for checking the rule during system entry validation):

```
SET CONTROL(GSO)
CHANGE INFODIR TYPES(R-RCAS) ADD
F ACF2,REFRESH(INFODIR)
```

3. Create the rule. The resource rule key is *RSA*, and the resource names are of the form *RSA.applid* where *applid* is the application name where the logon is taking place:
   - For CICS, IMS, or APPC/MVS applications, this is the VTAM application ID.
   - For TSO, it is the characters "TSO" suffixed with the SMF system ID as defined in the SMFPRM*xx* member *of SYS1.PARMLIB*.
   - For z/OS batch jobs, it is the characters "MVS" suffixed with the SMF system ID.

Note: Any special characters in the SMF system ID are ignored (for example, "SY*6" becomes "SY6"). Consult your product documentation for details about the application name that is used by your product.

Use the following specification as an example for creating the rule:

```
t r(cas)
RESOURCE
compile
ACF70010 ACF COMPILER ENTERED
. $KEY(RSA) TYPE(CAS)
. CICS01 UID(ABC) ALLOW
. MVSSYSA UID(DEF) ALLOW
. TSOSYSA UID(GHI) ALLOW
. - UID(JKL) ALLOW
. END
STORE
F ACF,REBUILD(CAS)
```

Based on the sample rule above:

- All logonids with UID strings that start with *ABC* must supply their RSA passcode when logging on to the CICS01 application.
- All logonids with UID strings that start with *DEF* must supply their RSA passcode during batch system entry validation on system SYSA (unless the logonid is inherited from the submitter).
- All logonids with UID strings that start with *GHI* must supply their RSA passcode when logging on to TSO on system SYSA.
- All logonids whose UID strings start with *JKL* must supply their RSA passcode for any type of system entry validation.

Note: You have the option to map an RSA user ID to a CA ACF2 logonid if they are different. For more information, consult the CA ACF2™ for z/OS - 16.0 Administration guide https://docops.ca.com/ca-acf2-for-z-os/16-0/en/administrating/enable-multi-factor-authentication/control-whether-users-sign-on-through-multi-factor-authentication.

4. Issue the command below to start the Advanced Authentication Mainframe started task.

```
START MFASTC
```

! **Important**: The Advanced Authentication Mainframe started task must start *after* TCP/IP is available. If TCP/IP is stopped, the address space terminates and cannot be restarted until TCP/IP is restarted.

At startup, Advanced Authentication Mainframe issues the following messages:

- MFA00100 MFA Initialization in progress
- MFA00521 JVMNAME IS: JVMLDM76
- MFA00101 MFA Initialization complete

After initialization is complete, users can sign on with their RSA SecurID credentials.

Stop the Advanced Authentication Mainframe Started Task

To stop the started task, you can issue the following command from the console:

```
STOP MFASTC
```

At shutdown, Advanced Authentication Mainframe issues the following messages:

- JZOS - MVS STOP command received
- MFA00102 MFA Shutdown in progress
- MFA00103 MFA Shutdown is complete

After Advanced Authentication Mainframe shuts down, RSA users need to use their regular CA ACF2 password or password phrase to log on.

Login Screenshots

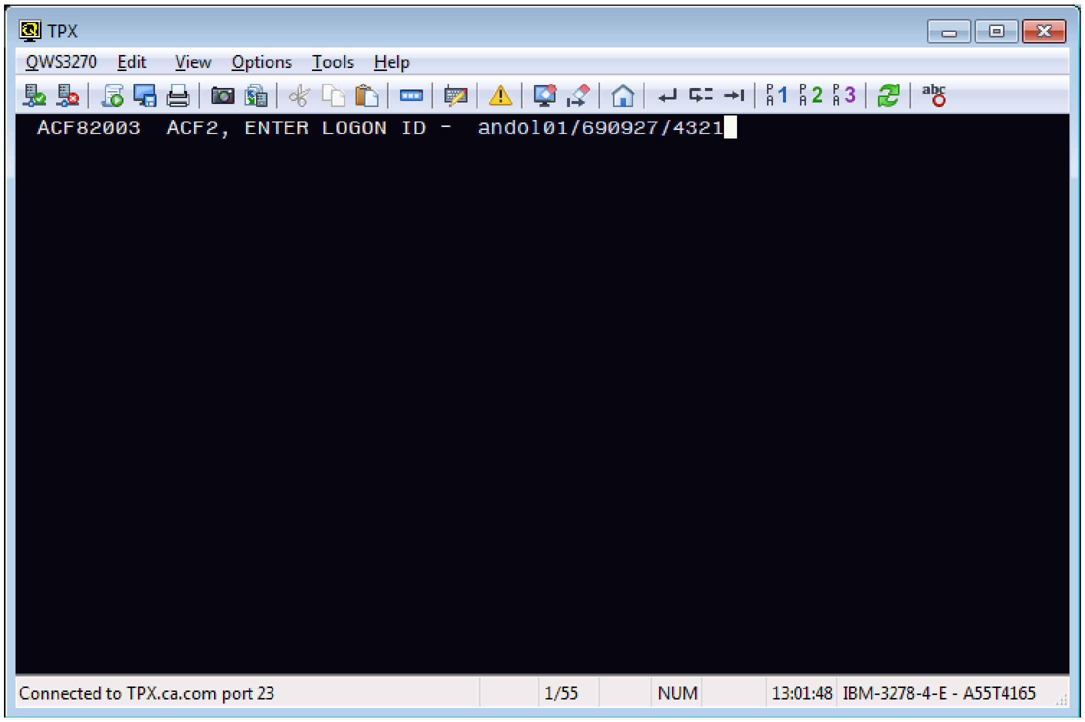
! **Important**: The integration uses the standard z/OS password login command line prompts and repurposes z/OS's password reset format (*logind/password/new password*) for submitting RSA SecurID credentials.

Credentials are entered as a single string that is divided into three sections, which are delimited by a forward slash. The standard z/OS *password* and *new password* sections are used for RSA SecurID tokencodes and PINs, respectively. Note that the PIN is entered **after** the tokencode. Thus, RSA SecurID users must enter their credentials as follows.

loginid/tokencode/PIN

Standard Passcode Login:

The user enters a username in the *loginid* section, a token code (690927) in the *password* section, and a PIN (4321) in the *new password section*.

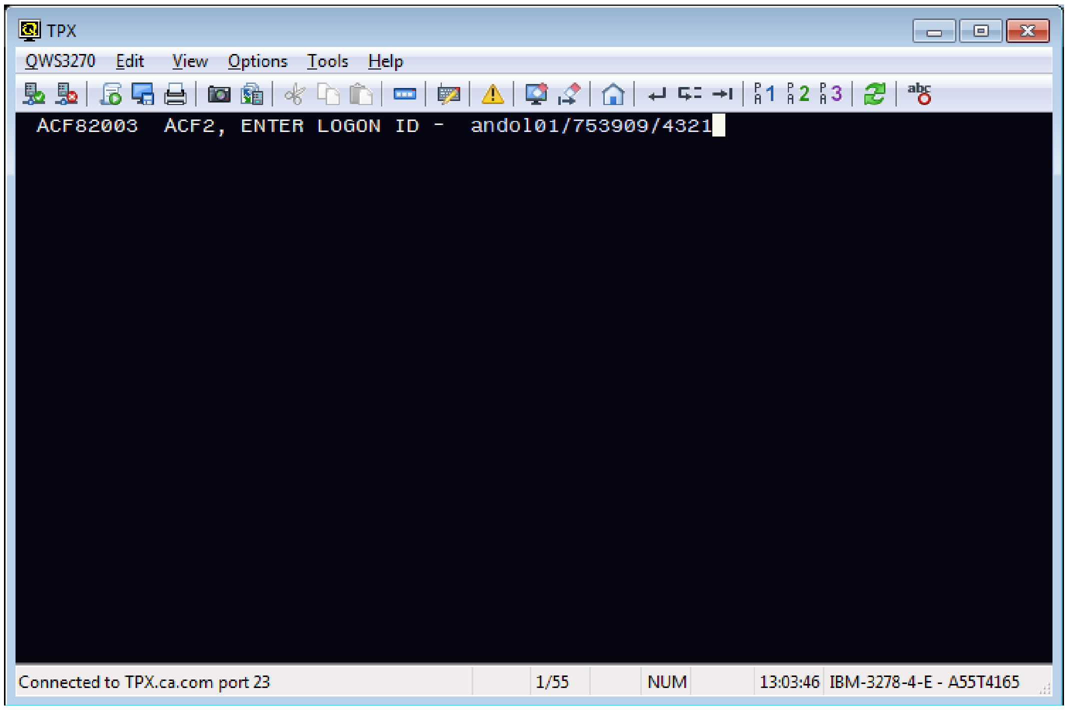


```
 ACF82003  ACF2, ENTER LOGON ID -  andol01/690927/4321
```

Connected to TPX.ca.com port 23 1/55 NUM 13:01:48 IBM-3278-4-E - A55T4165

User-defined New PIN:

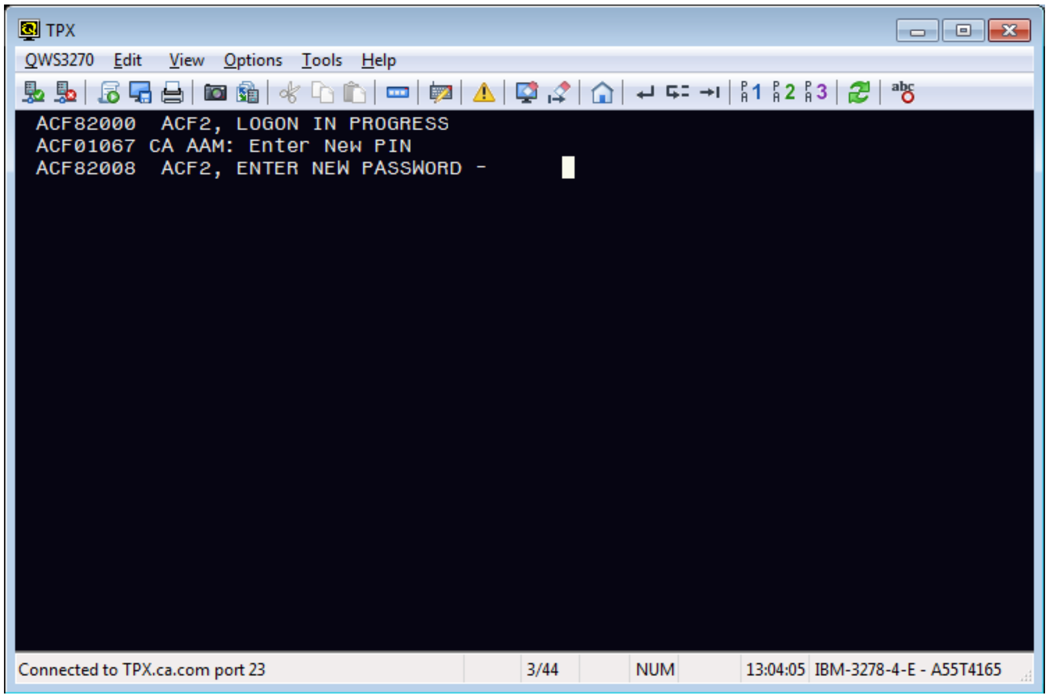
1. The user enters a tokencode (753909) in the *password* section and the current PIN (4321) in the *new password* section.


TPX
QWS3270 Edit View Options Tools Help

```
ACF82003  ACF2, ENTER LOGON ID -  andol01/753909/4321
```

Connected to TPX.ca.com port 23
1/55
NUM
13:03:46 IBM-3278-4-E - A55T4165

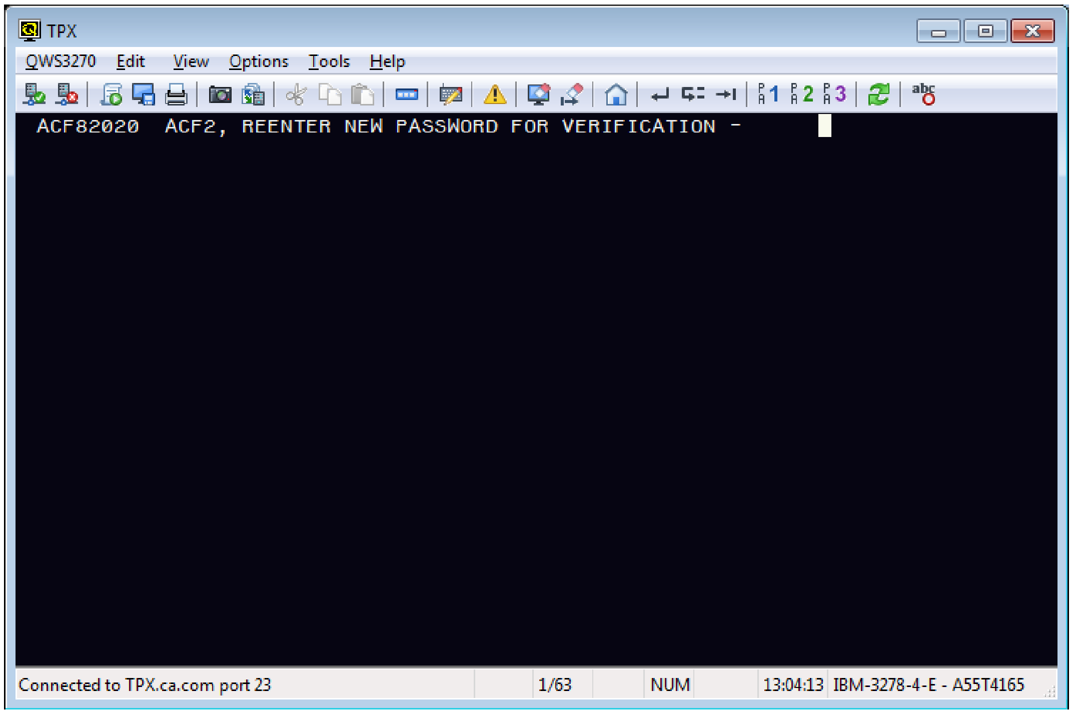
2. The system displays a message to enter a new PIN followed by the standard z/OS new password prompt. The user enters a new PIN (4444 – not visible).


TPX
QWS3270 Edit View Options Tools Help

```
ACF82000  ACF2, LOGON IN PROGRESS
ACF01067 CA AAM: Enter New PIN
ACF82008  ACF2, ENTER NEW PASSWORD -
```

Connected to TPX.ca.com port 23
3/44
NUM
13:04:05 IBM-3278-4-E - A55T4165

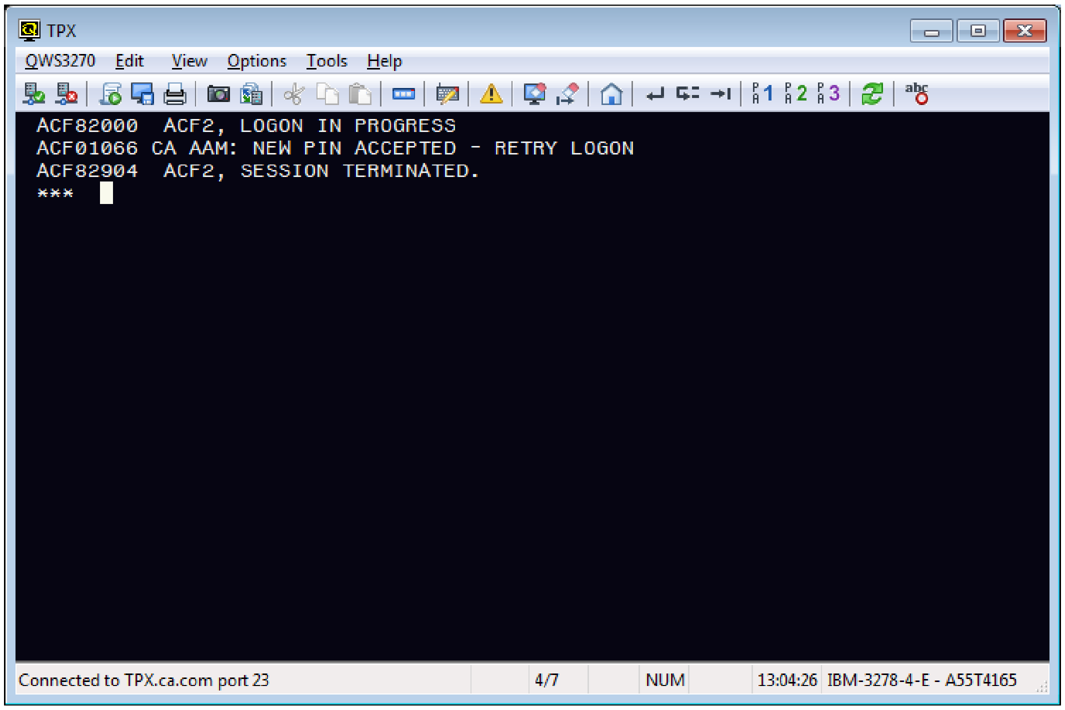
3. The user is asked to reenter the new PIN (4444 – not visible).



```
ACF82020  ACF2, REENTER NEW PASSWORD FOR VERIFICATION -
```

Connected to TPX.ca.com port 23 1/63 NUM 13:04:13 IBM-3278-4-E - A55T4165

4. The system informs the user that the new PIN was accepted. The user must now start the login process over again, using the new PIN. See the Standard Passcode Login section.

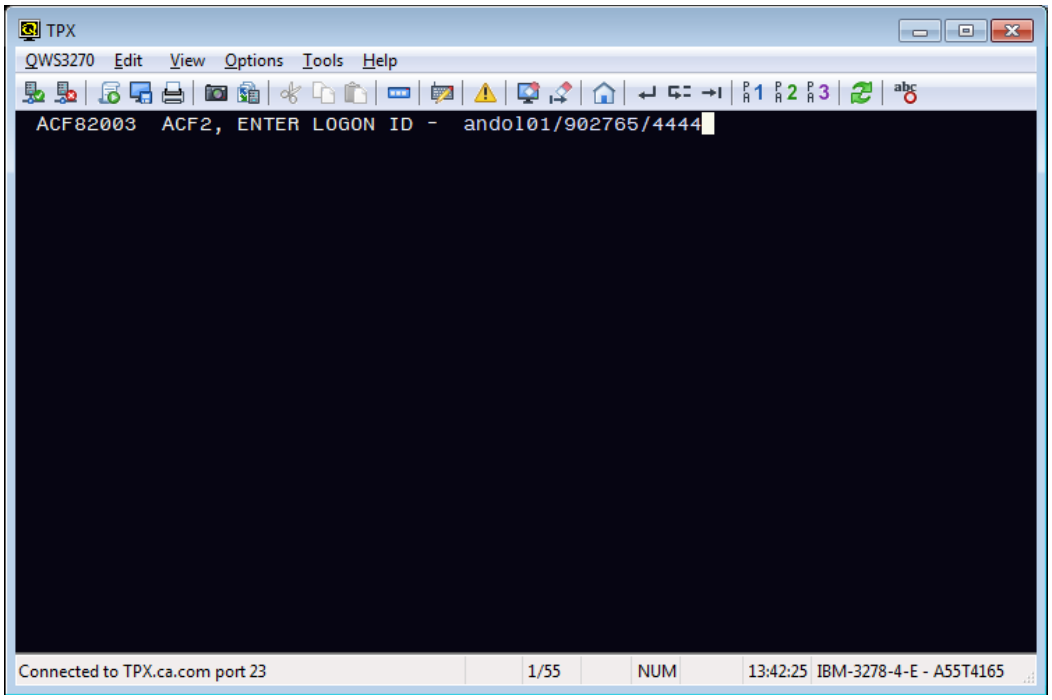


```
ACF82000  ACF2, LOGON IN PROGRESS
ACF01066 CA AAM: NEW PIN ACCEPTED - RETRY LOGON
ACF82904  ACF2, SESSION TERMINATED.
***
```

Connected to TPX.ca.com port 23 4/7 NUM 13:04:26 IBM-3278-4-E - A55T4165

Next Tokencode:

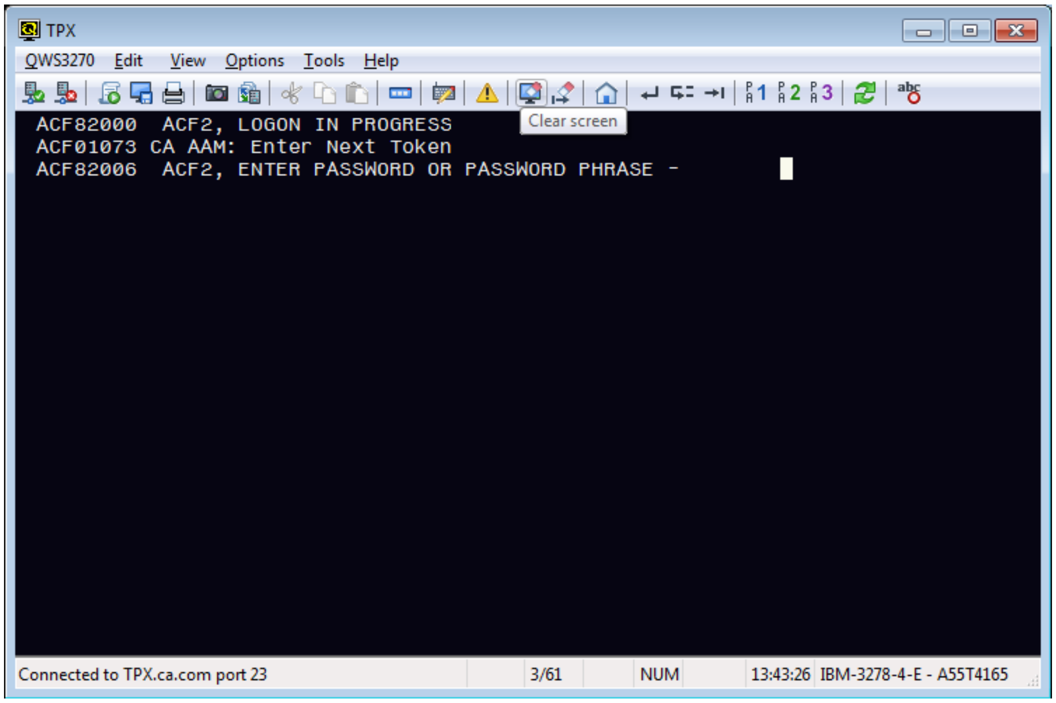
1. The user enters a username in the *loginid* section, a token code (902765) in the *password* section, and a PIN (4444) in the *new password* section.



```
ACF82003  ACF2, ENTER LOGON ID -  andol01/902765/4444
```

```
Connected to TPX.ca.com port 23    1/55    NUM    13:42:25  IBM-3278-4-E - A55T4165
```

2. The system displays a message to enter the next token code followed by the standard z/OS password/passphrase prompt. The user enters the next tokencode (not visible).



```
ACF82000  ACF2, LOGON IN PROGRESS
ACF01073 CA AAM: Enter Next Token
ACF82006  ACF2, ENTER PASSWORD OR PASSWORD PHRASE -
```

```
Connected to TPX.ca.com port 23    3/61    NUM    13:43:26  IBM-3278-4-E - A55T4165
```

Certification Checklist for RSA SecurID Access

Certification Environment Details:

RSA Authentication Manager 8.2, Virtual Appliance

RSA Authentication Agent 8.1 API for Java

CA-ACF2 for z/OS - 16.0

RSA Authentication Manager

Date Tested: February 23, 2018

Authentication Method	REST Client	UDP Agent	TCP Agent	RADIUS Client
RSA SecurID	-	✓	-	-
RSA SecurID Software Token Automation	-	-	-	-
On Demand Authentication	-	-	-	-
Risk-Based Authentication		-		-

✔ = Passed, **X** = Failed, **-** = N/A

Known Issues

The integration does not support system-generated PINs.

SMF records are not normally cut for the RSA rule, even when the rule line says *LOG*. To generate SMF logging records for the RSA rule, set TRACE in the user's logonid record. Use the *ACFRPTRV* report to view the loggings. Depending on the logon application, it might not be clear what will be used for the applid in the RSA rule. The *ACFRPTRV* report will show the applid in the resource name.

Appendix

RSA SecurID Access Integration Details

Partner Integration Details	
RSA Authentication Agent API (UDP)	Java 8.1

RSA Authentication Agent Files

RSA SecurID Authentication Files	
UDP Agent Files	**Location**
sdconf.rec	The *%MFASTC%* directory
sdopts.rec	The *%MFASTC%* directory
Node secret (*securid*)	The *%MFASTC%* directory
sdstatus.12	The *%MFASTC%* directory

! **Important**: If you have already authenticated to the z/OS system using RSA SecurID credentials, copy the existing Node secret file (*securid*) file to the *%MFASTC%* directory. If you clear the node secret on the RSA Security Console, you must manually remove the file from the *%MFASTC%* directory. Upon the next successful logon, the new *securid* file will be generated.

RSA Authentication Agent API Properties

```
#   rsa_api.properties file

#  Override Host IP Address
RSA_AGENT_HOST=%RSA_AGENT_NAME%
#  Type of the Server configuration.
SDCONF_TYPE=FILE
#  Path of the Server configuration.
SDCONF_LOC=%MFASTC%/sdconf.rec
#  Type of the Server statuses.
SDSTATUS_TYPE=FILE
#  Path of the Server statuses.
SDSTATUS_LOC=%MFASTC%/JAStatus.1
#  Type of the Server options.
SDOPTS_TYPE=FILE
#  Path of the Server options.
SDOPTS_LOC=%MFASTC%/sdopts.rec
#  Type of the Node Secret.
SDNDSCRT_TYPE=FILE
#  Path of the Node Secret.
SDNDSCRT_LOC=%MFASTC%/securid
#  Logs event messages to the console.
RSA_LOG_TO_CONSOLE=YES
#  Logs event messages to a file.
RSA_LOG_TO_FILE=NO
#  Name of the log file.
RSA_LOG_FILE=rsa_api.log
#  Minimum severity level allowed to log.
RSA_LOG_LEVEL=INFO
#  Enables debug tracing.
RSA_ENABLE_DEBUG=NO
#  Sends tracing to the console.
RSA_DEBUG_TO_CONSOLE=YES
#  Sends tracing to a file.
RSA_DEBUG_TO_FILE=YES
#  Name of the trace file.
RSA_DEBUG_FILE=%MFASTC%/rsa_api_debug.log
#  Allows function entry tracing.
RSA_DEBUG_ENTRY=YES
#  Allows function exit tracing.
RSA_DEBUG_EXIT=YES
#  Allows control flow tracing.
RSA_DEBUG_FLOW=YES
#  Allows regular tracing.
RSA_DEBUG_NORMAL=YES
#  Traces the location.
RSA_DEBUG_LOCATION=YES
```