# Micro Focus® Advanced Authentication Connector for z/OS®

## Installation and Getting Started Guide

Release: 1.2

Publication date: Updated March 2022

# Table of Contents

# Welcome to Micro Focus® Advanced Authentication Connector for z/OS®

Thank you for choosing the Micro Focus® Advanced Authentication Connector for z/OS® (hereafter called AACZ).

The instructions in this manual have been verified for:

- The specified versions of the System Authorization Facility (SAF) and Resource Access Control Facility (RACF®) that IBM has made available to Micro Focus.
- The specified versions of CA ACF2™ (hereafter called ACF2) and CA Top Secret® (hereafter called Top Secret).

These instructions should be read in conjunction with the relevant documentation supplied with your specific z/OS security product (IBM RACF, Broadcom ACF/2, or Broadcom Top Secret).

Consult the documentation for the NetIQ Advanced Authentication Server for information about how to set up and implement the NetIQ Advanced Authentication Server. This documentation is a prerequisite for installing and configuring AACZ. The NetIQ Corporation is a Micro Focus company.

**Audience and Scope**

This manual is intended for system administrators responsible for installing and administering AACZ.

**Running the Installer**

Appendix A, "Uploading the MFAACZ Distribution Files to the Host" on page 45 describes how to move the product distribution files to the mainframe host and expand them into libraries.

**Using this Manual**

The *Micro Focus Advanced Authentication Connector for z/OS Installation and Getting Started Guide* is made available in the Adobe Portable Document Format (PDF). To view PDF files, use Adobe® Reader®, which is freely available from www.adobe.com.

**TIP** Be sure to download the *full version* of Reader. The more basic version does not include the search feature.

This section highlights some of the main Adobe Reader features. For more detailed information, see the Adobe Reader online help system.

This PDF manual includes the following features:

- **Bookmarks.** This manual contains predefined bookmarks that make it easy for you to quickly jump to a specific topic. By default, the bookmarks appear to the left of each page.
- **Links.** Cross-reference links within the manual enable you to jump to other sections within the manual with a single mouse click. These links appear in blue.
- **Printing.** While viewing a manual, you can print the current page, a range of pages, or the entire manual.

# Typographical Conventions

The following typographical conventions are used in this manual. These typographical conventions are used to assist you when using the documentation; they are not meant to contradict or change any standard use of typographical conventions in the various product components or the host operating system.

| Convention | Explanation |
|---|---|
| *italics* | Introduces new terms that you may not be familiar with and occasionally indicates emphasis. |
| **bold** | Emphasizes important information and field names. |
| UPPERCASE | Indicates keys or key combinations that you can use. For example, press the ENTER key. |
| `monospace` | Indicates syntax examples, values that you specify, or results that you receive. |
| *`monospaced italics`* | Indicates names that are placeholders for values you specify; for example, *`filename`*. |
| vertical rule \| | Separates menus and their associated commands. For example, select File \| Copy means to select Copy from the File menu.<br>Also, indicates mutually exclusive choices in a command syntax line. |

# Chapter 1

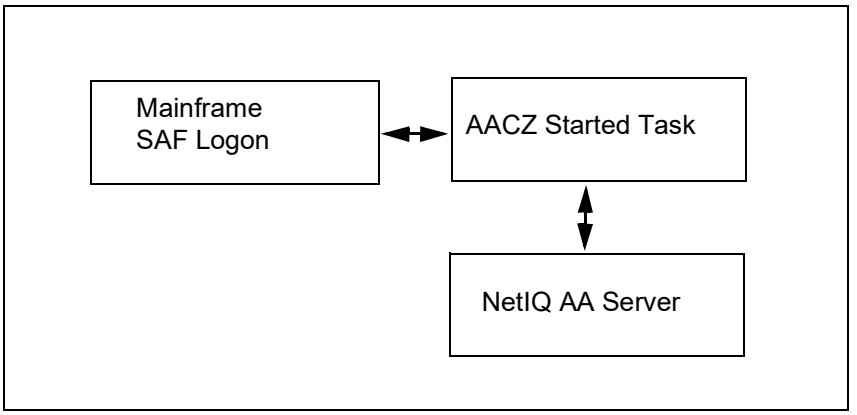
# Overview

The Micro Focus Advanced Authentication Connector for z/OS (AACZ) provides a connection between the IBM® System Authorization Facility (SAF)-managed log-on process and the external, Multi-Factor Authentication (MFA) process provided by the NetIQ Advanced Authentication (AA) Server.

This connection allows the NetIQ AA Server to manage the log-on process for any mainframe applications that interface with SAF, such as TSO, CICS®, z/OSMF, ChangeMan® ZDD, ChangeMan ZMF for Eclipse, and any user applications that require specific log-on authentication.

This processing, for example, enables you to use your general network (LDAP) password rather than the Resource Access Control Facility (RACF®) password to log on to TSO. The multi-factor aspect of this feature means that two or more independent user verification methods can be required during logon, for example, LDAP password followed by input from your registered smartphone.

The following diagram summarizes AACZ processing:




RACF, ACF2, and Top Secret are the SAF facilities that are addressed in this release. If the relevant user segment has been defined in one of these facilities, the facility will hand control of the user verification process to AACZ. AACZ communicates with the NetIQ AA Server to determine which methods are required to authenticate the user who is attempting to log on. Each method must be completed with a positive result if the user log-on process is allowed to proceed; otherwise, the log-on attempt fails and the relevant return code, reason code, and message are returned.

# Chapter 2

# Implementing the Micro Focus Advanced Authentication Connector for z/OS

This chapter describes the actions you must take to implement AACZ.

# Prerequisites

The following products or features need to be in place and operational.

| Product/Feature | Remarks |
|---|---|
| IBM Integrated Cryptographic Service Facility (ICSF) | Needs to be configured. ICSF is part of the z/OS operating system. (See "ICSF Requirements" on page 26.) |
| IBM CP Assist for Cryptographic Function (CPACF) | Needs to be configured. This feature should already be available at most customer sites. |
| IBM z/OS Client Web Enablement Toolkit | This product is part of the z/OS operating system from z/OS 2.2 (available via ptf for z/OS 2.1). |
| RACF passphrase support | Although passphrase support is not required, MFA usability is limited if passphrase support has not been enabled. We recommend that passphrase support be enabled if at all possible to allow the full 100-character passphrase field to be available during the log-on process. (See "Passphrase vs Password Support" on page 26.) However, the use of out-of-band support removes the need for passphrase support. |
| ACF2/TSS | Must support MFA (v16+) |
| Micro Focus Advanced Authentication Server | v6.2+ |

# Uploading the AACZ Product Distribution Files to the Host

Refer to Appendix A, "Uploading the MFAACZ Distribution Files to the Host" on page 45, for instructions on uploading the product distribution files to the host and expanding them into libraries. After you have done this, you can create the AACZ started task.

# Creating the AACZ Started Task

Member STCMFA of the CNTL distribution library contains the JCL to create the AACZ started task. You will need to run a AACZ started task on each LPAR where MFA is to authenticate user logons.

Customize this JCL as appropriate for your installation and environment. For example:

1. Specify a name for this procedure on the PROC statement.
2. Specify the appropriate high-level qualifier for the *somnode* variable in DSN parameters.
3. Specify the appropriate DSN name for *your.apf.linklib* in the STEPLIB DD statement.

The contents of the DDnames that the procedure references are as follows:

| DDname | Description |
|---|---|
| MAALOG | Where successful or unsuccessful attempts to log on are recorded, one line per logged event. This DDname can point to SYSOUT or to a data set with RECFM=FB, LRECL=132. |
| MAADEBUG | Where trace output is written. This DDname can point to SYSOUT or to a data set with RECFM=FB, LRECL=132. |
| MAAENDPT | Where the NetIQ AA Server endpoint id and secret are stored. This information is critical to secure AACZ operation. This data set must be protected with a UACC of NONE and accessible only by the started task userid (UPDATE access is required) and relevant system admin personnel. Data set attributes are RECFM=F, LRECL=32 (contains only 4 records). |
| MAASNAP | Where information about abnormal end situations is written. |
| SYSIN | Contains the start-up parameters. (See Start-Up Parameters on page 12.) |
| SYSPRINT | Where general operational messages are written, usually to SYSOUT. |

Set up the started task security. The started task must be assigned a valid SAF userid which allows access to Unix System Services. For example, you can assign a RACF userid to the started task by adding a profile in the STARTED class. You must also ensure that the RACF userid has an OMVS segment.

**IMPORTANT!** An incompatibility between Top Secret and IBM® HourGlass has been noted. If both Top Secret and HourGlass are active, the AACZ started task issues an S0D7-25 abend the first time the AACZ started task attempts to access OMVS resources. This abend will appear in the job log of the AACZ started task. The abend does not prevent the authorization from working, but its appearance in the job log is confusing.

The current workaround is to include the following DD statement in the JCL for the AACZ started task:

//*xx*BYPASS  DD  DUMMY

where the value of *xx* is determined by the person who installed HourGlass. For example, HG is a typical value for *xx*.

If you want to make use of RACF passticket authentication via the AACZ facility, you need to allow the userid under which the AACZ started task is running to have READ access to a specific PTKTDATA class profile.

What is needed is encapsulated in the following RACF commands:

```
RDEFINE PTKTDATA IRRPTAUTH.*.* OWNER(<yourOwner>) UACC(NONE)

PERMIT IRRPTAUTH.*.* CLASS(PTKTDATA) ACCESS(READ) ID(<aaczStcUserid>)
```

```
SETROPTS RACLIST(PTKTDATA) REFRESH
```

The RDEFINE command sets up a profile in the PTKTDATA class which covers the evaluation of passtickets for all applications and all userids.

The PERMIT command gives the appropriate access to the AACZ stc userid (you can either use the actual userid or a group to which it is connected).

The SETROPTS command refreshes centrally cached copies of PTKTDATA profiles.

Similar steps are needed if you are using ACF2 or Top Secret products.

## Additional Notes About Creating the AACZ Started Task

The AACZ software must run in a Language Environment (LE) enclave that has POSIX(ON) set. If you are not sure that POSIX(ON) is your default setting, you can ensure it is set by adding a CEEOPTS DD statement to the started task JCL. For example:

```
//CEEOPTS   DD DISP=SHR,DSN=somnode.MFA.PARMLIB(LEOPTS)
```

with the LEOPTS member containing the following statement:

```
POSIX(ON)
```

# Start-Up Parameters

Sample start-up parameters are provided in member MFACNTL of the SAMPLES distribution library. The following table includes the full list of available parameters:

| Parameter | Description |
|---|---|
| AT-TLS | Use this parameter if you want AACZ to use AT-TLS policies to secure the connection to the AA-server.<br>In this case it is the responsibility of those policies to specify the appropriate keyring, etc., to support secure communications.<br>If you specify this option, any other SSL-related option (such as keyring) is ignored. |
| AuthRequestTimeOut=nnn | Optional parameter that specifies a time-out value for the AACZ started task for all authentication requests.<br>▪ nnn is the number of seconds that you want to keep trying to logon. If this parameter is omitted, the default value is 15 seconds less than that specified for a TSO logon process time out, or 600 seconds if no value is available for TSO logon.<br>See Note 2 for additional information. |

| Parameter | Description |
|---|---|
| Compound=Standard | Optional parameter to invoke compound in-band processing. This is where the normal AA server authentication process is augmented by the local SAF security mechanism (e.g. RACF etc. password checking). More details can be found in "Using the compound In-Band mechanism" on page 33.<br>This option requires pass phrases to be active. The response passed to the MFA process consists of the RACF password/phrase followed by a colon and then the response required by the AA server authentication process. |
| ConcurrentAuth=nnn | Optional parameter which can be used to restrict the number of authentication requests that are processed simultaneously. The default is the same as the MAXUSERPROC setting for the LPAR on which the stc is running.<br>If a setting higher than the MAXUSERPROC value is requested, it will be ignored and the MAXUSERPROC value will be used. The values for both the MAXUSERPROC and the actual concurrent authentication limit in use is shown in sysprint during start up.<br>See "Concurrency" on page 19. |
| Delimiter=*value* | Optional parameter that specifies the character that delimits multi-response passphrases if they are being used. Valid values are:<br>■ *x*, where *x* is the single-byte character that is used as a delimiter. The comma is the default if the Delimiter parameter is omitted.<br>■ **NONE** - Specifies that multi-response passphrases are not in use.<br>See Note 3 for additional information. |
| KeyDbFile=/*path*/*file_name*.kdb | Is one of two options that you can specify to secure communications with the NetIQ AA Server. Specifies the zFS path, filename, and type (kdb) of the KeyDbFile where the SSL certificate is stored. If this option is used, the KeyStashFile parameter must also be specified.<br>Mutually exclusive with KeyRing parameter. |
| KeyRing=*userid*/*ringname* | Is one of two options that you can specify to secure communications with the NetIQ AA Server. Specifies where the Secure Sockets Layer (SSL) certificates can be found within the RACF database. The format is *userid*/*keyringname*, where:<br>■ *userid* is the RACF userid under which the AACZ started task is running.<br>■ *keyringname* is the name of the keyring.<br>Mutually exclusive with KeyDbFile parameter. |
| KeyStashFile=/*path*/*file_name*.sth | Allows access to the SSL certificate (as generated by the gskkyman utility). This parameter must be specified if the KeyDbFile parameter is specified. |
| MixedCase=YES/NO | If using compound-in-band this parameter specifies whether SAF passwords should be treated as mixed case by the SAF software. If the value of NO is used, or the parameter is omitted, then passwords will be folded to upper case before they are passed to the SAF software. |

| Parameter | Description |
|---|---|
| OobTimeOut=*nnn* | Deprecated—has same meaning as the AuthRequestTimeOut parameter |
| Port= | The port on which the NetIQ server is listening at the server address. This parameter is optional, the default port will be used if missing. |
| Recovery=YES/NO | Optional parameter which can be used to suspend recovery processing (Recovery=NO) in order to get a full dump for an error scenario. The default is Recovery=YES. |
| Revoke=*value* | Optional parameter that specifies if an authentication failure results in the RACF revoke count being incremented by 1. Valid values are:<br>■ YES (or Y) - The RACF revoke count is incremented by 1 for each authentication failure. YES is the default if this parameter is omitted.<br>■ NO (or N) - The RACF revoke count is not incremented by 1 for each authentication failure.<br>You can use the following z/OS modify command to turn this parameter on or off while the AACZ started task is running:<br>`/F stcname,REVOKE=value`<br>See Note 1 for additional information. |
| SafOnly=YES/NO | If using compound-in-band this parameter specifies whether a SAF-only authentication is permitted in the case where the NetIQ process has failed for some reason. If the value of NO is used, or the parameter is omitted, then failure in the NetIQ process will result in authentication being denied. |
| Server=*name* | The IP address of the NetIQ AA server that is used to authenticate log-on requests. This parameter is required and can be specified as a DNS name, an IPV4 address or an IPV6 address.<br>Examples:<br>`Server=www.my.netIQ.server`<br>`Server=192.168.0.1`<br>`Server=[2001:1890:1112:1::20]` |

| Parameter | Description |
|---|---|
| Trace=*value* | Optional parameter that specifies if and how tracing is to be used. Valid values are:<br>■ **NO** - The default if this parameter is omitted.<br>■ **YES** - Turns on standard tracing for all userids. All permanent passwords are overlaid with asterisks. Any information about the Endpoint id and Secret is missing from trace entries. Trace entries are written to the MAADEBUG DDname.<br>■ **YES,USERID=***userid* - Turns on standard tracing for the specified userid.<br>Valid variations to these values are:<br>■ YES or Y<br>■ NO or N<br>■ USERID or USER or U<br>You can use the following z/OS modify command to turn tracing on or off while the AACZ started task is running:<br>`/F stcname,TRACE=value` |

Note 1: **Revoke=** is an optional parameter with a default of YES. The default setting means that each authentication failure that the AACZ started task generates will result in the RACF revoke count being incremented by 1, eventually resulting in the userid's being revoked. With this default setting, each time such a failure occurs the following RACF message will be seen in syslog:

```
ICH408I USER(userid) GROUP(group) NAME(user_name) 595
LOGON/JOB INITIATION - MULTIFACTOR AUTHENTICATION FAILURE
```

For testing purposes, you can specify **Revoke=NO** to avoid having failures increase the RACF revoke count, and thus avoid having userids continually revoked. A consequence of this setting is that no ICH408I messages are issued for authentication failures. However, the AACZ started task writes a line to the MAALOG DDname for each such success and/or failure. For example:

```
20170919 08454175 USER123 MAA3001I Authorization was successful using chain: TOTP
20170919 18012900 USER123 MAA3002I Authorization denied: TOTP_PASSWORD_WRONG
```

if the RACF MFA option NOPWFALLBACK is specified for the userid in question, all attempts of that user to log on will fail if the AACZ started task is not available. In this case, the following message is written to syslog:

```
ICH408I USER(userid) GROUP(group) NAME(user_name) 595
LOGON/JOB INITIATION - MULTIFACTOR AUTHENTICATION UNAVAILABLE
```

If option PWFALLBACK is used, a user will be able to log on using his or her RACF password if the AACZ started task is down.

**CAUTION!**: Using option PWFALLBACK can lead to confusion and revoked userids, because most users will not be able to remember their RACF passwords once external authentication has been in use for more than a few days. However, if you want to keep passwords current and, in extremis, as a backup method to using MFA then you should consider using the Compound in-band mechanism (see "Using the compound In-Band mechanism" on page 33.)

| Parameter | Description |
|---|---|
| Note 2: The TSO logon process has a built-in default timeout value of 300 seconds. (This can be changed using the IKJTSOxx system parmlib member). If any authentication mechanism causes a delay longer than this during a TSO logon attempt, you may find the TSO user becomes unusable until it is removed from the system via an operator command. You may also find that recycling the AACZ stc can free up such a 'stuck' user.<br>It is recommended that this AACZ timeout value be set at less than the TSO logon timeout value to avoid such issues. | |
| Note 3: The mainframe log-on process typically has a single input, that is, a password or passphrase (up to 100 bytes). There is no possibility of a conversation between the NetIQ AA Server and the mainframe log-on process. To allow multiple methods that require some kind of input pass code to be used, AACZ supports the interpreting of the single passphrase into multiple responses.<br>Thus, for example, if the authentication chain implements two methods, for example, LDAP password and TOTP (time limited one-time password), the user can enter *ldap-pwd,totpcd* in the passphrase. In this case, AACZ will present *ldap-pwd* to the first method and *totpcd* to the second.<br>Furthermore, the NetIQ AA administrator has the option to implement up to nine different chains per user. The default chain is number 1.<br>The user can select a different chain by prefacing his or her response with a number *n*, where *n* is between 1 and 9. For example, *n*,*ldap-pwd*,*totpcd*<br>Using the preceding example for the default chain (chain 1):<br>*ldap-pwd*,*totpcd* and 1,*ldap-pwd*,*totpcd* are equivalent. | |

# Notes on the Use of Signed Certificates

The self-signed certificate that comes with the NetIQ Advanced Authentication Server or a certificate that has been signed by a Certificate Authority such as Verisign can be used with the AA Server.

## Using a Certificate that Has Been Signed by a Certificate Authority

If you are using a certificate that has been signed by a Certificate Authority such as Verisign, you only need to install that certificate on the AA Server. To do this, follow the instructions provided in the NetIQ AA Server documentation.

## Using a Self-Signed Certificate

If you are not using a certificate that has been signed by a Certificate Authority, you need to extract the self-signed certificate that comes with the AA Server and save it in a RACF Key Ring (or the ACF2 or Top Secret equivalent) that is then defined in the KeyRing= parameter in the start-up parameter file before you attempt to start up the z/OS Connector. The KeyRing= parameter is described in the section titled "Start-Up Parameters" on page 12.

With the Chrome browser, for example, the easiest method to extract that self-signed certificate is as follows:

**1** Access the URL or TCP/IP address of the AA Server.

2 Press CTRL-SHIFT-I to toggle the developer tools pane on/off.

3 Select the Security tab.

4 Select the View Certificate button.

5 Select the Details tab in the certificate viewer, then use the **Save to file** option to export the certificate.

6 Use the base-64 encoded option as the file format.

# RACF, ACF2, and Top Secret Requirements and Commands

This section provides an overview of the RACF, ACF2, and Top Secret requirements and commands that are needed to enable MFA. See the RACF, ACF2, or Top Secret user documentation for the full details of enabling MFA.

## Defining an MFA Profile

### *For RACF:*

1 To implement MFA, your RACF administrator must first activate the MFADEF class in RACF. For example:

```
SETROPTS RACLIST(MFADEF) CLASSACT(MFADEF) GENERIC(MFADEF)
```

2 Then, define a profile in this class like this:

```
RDEFINE MFADEF FACTOR.AACZ
```

where FACTOR.AACZ should be specified exactly as shown. The Micro Focus Advanced Authentication Connector for z/OS expects this value.

3 Then, refresh the RACLISTed class:

```
SETROPTS RACLIST(MFADEF) REFRESH
```

### *For ACF2:*

Equivalent commands for ACF2 are:

```
SET CONTROL(FACTOR)
INSERT AACZ ACTIVE
F ACF2,REFRESH(FAC),TYPE(FAC)
```

### *For Top Secret:*

Equivalent command for Top Secret is:

```
TSS MODIFY MFA(IBMRSA(YES,NOFALLBACK))
```

## Defining Users to the NetIQ AA Server

Unless the TSO userid and the userid for the NetIQ AA Server are exactly the same (in which case the TAGS/MFADATA sub-parameters may be omitted from the following examples), your security administrator will need to correlate the TSO userid for each user with the userid by which the user is known to the NetIQ AA Server. For example, if the NetIQ AA Server is validating userids by your company's LDAP directory, the following commands associate TSO userid DJACKSO with the NetIQ AA Server userid of DJACKSON:

***For RACF:***

```
ALTUSER DJACKSO MFA(FACTOR(AACZ) ACTIVE NOPWFALLBACK
TAGS(AAUSERID:DJACKSON))
```

***For ACF2:***

```
SET LID
CHANGE DJACKSO NOFALLBACK
SET P(USER),DIV(MFA)
INSERT DJACKSO.AACZ ACTIVE TAGS(AAUSERID:DJACKSON)
F ACF2,REBUILD(USR),CLASS(P)
```

***For Top Secret:***

```
TSS PER(Qyyyyyy) CASECAUT(TSSCMD.ADMIN.) ACC(UPD)
```

where:

*Qyyyyyy* is the userid of the administrator (the person who is attempting to add the MFA segment to the userid specified in the ADD parameter of the following command:

```
TSS ADD(DJACKSO) MFACTOR(AACZ) MFADATA(AAUSERID:DJACKSON)
MFACTIVE(YES) NOFALLBACK
```

**NOTE** Both the NetIQ AA Server and AACZ must be running for these commands to be issued successfully.

## Listing the Details About a Specific User

You can use the following commands to list the details of the MFA segment for a user profile whose TSO userid is DJACKSO:

***For RACF:***

```
LISTUSER DJACKSO MFA
```

***For ACF2:***

```
SET LID
LIST DJACKSO PROF(MFA)
```

If you want to check if FALLBACK is set:

```
LIST DJACKSO
```

***For Top Secret:***

```
TSS ADMIN(Qyyyyyy) DATA(MFA)
```

where:

*Qyyyyyy* is the userid of the administrator.

```
TSS LIST(DJACKSO) DATA(MFA)
```

# Concurrency

There are a number of environmental settings that limit the amount of concurrency (that is, the number of logons that can be processed simultaneously) available to the AACZ started task.

## Virtual storage

Each authentication subtask requires between 5 and 5.5 Mb of virtual storage. This storage is transient and is given up when the subtask ends (the whole process usually taking of the order of a second depending on the method chain). However, if 100 users are trying to logon at the same time, the AACZ started task will need at least 500Mb of virtual storage to process these. The sample started task JCL specifies REGION=0M to allow as much storage as necessary to be used by the AACZ started task. If you want to regulate this storage use more closely, you need to pay attention to the maximum concurrency rate (see below) in order to avoid S878 abends.

## USS processes

Each authentication subtask generates a unix process in its own right. Subsequent facilities called by the subtask may fork yet further (transient) processes. There is a system limit (set by systems programmers) on the number of processes that a single address space can fork, known as the MAXPROCUSER limit. The AACZ started task reports on this limit during start up:

```
Note: MAXPROCUSER value is   00000100
```

This limit is used as the maximum concurrency setting by the AACZ started task. If you want to restrict this further, then a sysin parameter is available "ConcurrentAuth=nnnn" and the AACZ started task will use whichever is the lower of the system MAXPROCUSER value and the sysin specified ConcurrentAuth value. (If the sysin value is missing, we just use MAXPROCUSER). This is reported during startup:

```
Input parameters:
-----------------

ConcurrentAuth=50

   Note: MAXPROCUSER value is   00000100
         Max concurrent auths   00000050
```

If the number of concurrent authentication requests exceeds this maximum value, the AACZ started task introduces delays and retry loops to attempt to allow the backlog to be addressed before proceeding with the current request. Note that massive concurrency is typically only seen when driven by automated processes (e.g. many asynchronous tasks requiring a mainframe logon driven by an off-platform script). These delays and retries (every 0.5 seconds) will be attempted for 30 seconds before the authentication attempt is given up and control is returned to the user with the following message:

```
ICH70008I IBM MFA Message:
          MAA1010E AACZ task too busy (MAXPROCUSER), please try later.
```

The AACZ started task reports on any 15 minute period when the number of requested concurrent authentication requests exceeds the maximum (and, therefore, results in delays to the authentication process). If this is the case, the following message is produced in SYSPRINT:

```
20210119 042514 - Demand for concurrent authentication processes
    exceeded maximum in the last period
(15 mins). HWM: 00004093
```

And the following WTO is issued (to allow for automation processes):

```
04.25.14 S0096523  MAA0016W Concurrent authentication demand exceeded
    maximum,
HWM: 00004093
```

Every hour the AACZ started task outputs (in SYSPRINT) values for the concurrency high water mark (HWM) and number of authentication requests in the last hour:

```
20210119 042559 - Interval values - Concurrency HWM: 00004096;
    Authentication requests: 00006734
```

When the AACZ started task is stopped, it reports overall totals:

```
20210119 042707 - Overall totals  - Concurrency HWM: 00004096;
    Authentication requests: 00167456
```

While the AACZ started task is able to manage these large rates of concurrency, you may well run into the situation where the AA server itself is a bottleneck and unable to respond to the barrage of requests. You may see http errors (reported in SYSPRINT of the AACZ started task):

```
20210120 03273217 SDOWNE2  Http error at: Cses4000
20210120 03273217 SDOWNE2          HwtRC: 00000106
20210120 03273218 SDOWNE2    HwtDiag_rsn: 000F0000
20210120 03273218 SDOWNE2    HwtDiag_sno: 000001B5
20210120 03273218 SDOWNE2    HwtDiag_txt: Connection closed
20210120 03273218 SDOWNE2         Reason: Http failed to send request
    for: Create Endpoint Session.
20210120 03273218 SDOWNE2  MAA2003E Unable to establish endpoint
    session
```

The AACZ started task gives up at this point and returns to the logging on user denying access and indicating something went wrong with the MFA process. The way to address this problem is to configure a load balancing cluster of AA servers in order to supply more processing power at that end of the path (see the NetIQ documentation for more details).

# Recovery

By default, if an unexpected error occurs in the authentication processing, recovery routines write diagnostic information (to be reviewed by Micro Focus support) to the MAAPRINT dd statement. The logging on user is not authenticated and is sent a message indicating something went wrong. For example, a TSO user would receive the following message:

```
ICH70008I IBM MFA Message:
   MAA1011E abnormal end, please review output in the AACZ stc.
```

The following message is also displayed in the AACZ started task SYSPRINT dd:

```
AACZ stc has abended, please review output for cause. If a dump is
    required use RECOVERY=NO.
```

As indicated in this message, if you want to avoid recovery you can start the AACZ started task with the sysin parameter RECOVERY=NO. You may need to do this (in order to get a

full dump for the error situation) if Micro Focus support determines that the standard diagnostics are not sufficient to work out the problem.

If you need to cancel or force the AACZ started task, you will leave the connection between your SAF software and our AACZ software in place. This connection is normally removed as part of the started task shutdown process. If this connection is left in place after the AACZ stc has disappeared, you are likely to see SAF software abends during user authentication. Sample JCL member JCLRESET executes program MAARESET and is provided to allow you to unilaterally remove this connection in such situations.

You may also want to include this as an extra step in the AACZ stc JCL. The COND=EVEN parameter setting ensures that the connection is removed should the AACZ software abend (although if you cancel or force the stc, you will need to run JCLRESET manually).

- If the AACZ step terminates normally, the extra step simply reports that it has nothing to do.
- If MAARESET finds that AACZ is still "connected" to RACF on this LPAR, it removes the connection and issues the following WTO:

```
MAA0012I Micro Focus(R) Advanced Authentication Connector for
    z/OS(R) is disabled.
```

- If AACZ is not still connected, it does nothing and issues this WTO:

```
MAA0011E Micro Focus MFA facility is no longer installed, terminating
    without deleting token.
```

# NetIQ AA Server Concepts and Requirements

This section introduces some NetIQ AA Server concepts that you need to understand.

Endpoint

Each instance of the AACZ Server (one per LPAR where logons are processed) establishes an **endpoint** with the designated NetIQ AA Server. The names of these endpoints are **Mainframe *smfid***, for example Mainframe D001. This endpoint is the entry point for any ensuing authentication conversation. AACZ creates the endpoint automatically.

Event

An authentication **event** is triggered by an external device or application that needs to perform authentication.

**IMPORTANT!** The name of the event for AACZ processing is **Mainframe logon**.

One or more chains are associated with an event.

Method

A **method** is an authentication method (for example, an LDAP password, time-limited one-time password (TOTP), smartphone push button, and so on). An authentication method verifies the identity of an individual who wants to access data, resources, or applications.

Refer to the NetIQ Advanced Authentication documentation for complete details on the supported methods.

Chain

An authentication **chain** is a combination of authentication methods. A user must pass all methods in the chain to be successfully authenticated.

**NOTE** The NetIQ AA Server processes the various methods in the chain. AACZ only reacts to the responses that the NetIQ AA Server returns.

A user must first enroll in each method that the user will be required to use. The user will have to log on to the designated AA Server using the Self Service portal in order to enroll in these methods. At the end of this authentication process, AACZ passes a return code, reason code, and message back to RACF.

Refer to the NetIQ AA Server documentation for further information about how to implement the NetIQ AA features that you want to use with AACZ.

## The User Enrollment Process

A user must enroll in the various methods that your NetIQ AA administrator has configured. Refer to the NetIQ AA Server documentation, which fully documents the enrollment process.

The user will need to download and install the NetIQ Advanced Authentication app on his or her mobile phone, and enroll in the appropriate methods to use the phone-based authentication methods.

In the following example, the user brings up the NetIQ Advanced Authentication app that installed on the user's mobile phone. The resulting display on the mobile phone shows that the user has enrolled in one method: TOTP


# Example: Logging On to the Mainframe

In this example, we assume that **Chain A** and **Chain B** have been defined for event **Mainframe logon**.

| Event Name | Chain Name | Method 1 | Method 2 |
|---|---|---|---|
| Mainframe logon | Chain A | LDAP Password | TOTP |
| | Chain B | LDAP Password | Smartphone |

The user must enroll in two methods if Chain A is selected:

- LDAP Password
- TOTP

The user must enroll in two methods if Chain B is selected:

- LDAP Password
- Smartphone

If user DJACKSO2 logs on to TSO using Chain A, he must specify the following two method values in the Password field, separated by a comma (the default separator):

*LDAP_password,TOTP_code*

A sample log-on panel for TSO user DJACKSO2 follows. (Passphrase has been enabled for this example, and thus the user can specify up to 100 characters in the Password field.)

In the Password field, user DJACKSO2 specifies his LDAP password **abcdefghij** followed by a comma followed by the TOTP code **956924**, which is displayed on his mobile phone:

```
------------------------------ TSO/E LOGON ----------------------------------


   Enter LOGON parameters below:                  RACF LOGON parameters:

   Userid    ===> DJACKSO2

   Password  ===> abcdefghij,956924

   Procedure ===> ISPFD001                        Group Ident  ===>

   Acct Nmbr ===> ACCT#

   Size      ===>

   Perform   ===>

   Command   ===> ispf

   Enter an 'S' before each option desired below:
   -New Password    -Nomail    -Nonotice  S -Reconnect    -OIDcard

PF1/PF13 ==> Help    PF3/PF15 ==> Logoff    PA1 ==> Attention    PA2 ==> Reshow
You may request specific help information by entering a '?' in any entry field
```

In this example, the methods for Chain A are correctly specified, and the following messages are displayed:

```
ICH70008I IBM MFA Message:
          MAA1000I Logon successfully authenticated by Micro Focus MFA
ICH70001I DJACKSO2    LAST ACCESS AT 11:20:24 ON MONDAY, OCTOBER 30, 2017
IKJ56455I DJACKSO2 LOGON IN PROGRESS AT 14:05:15 ON NOVEMBER 20, 2017
```

If user DJACKSO2 makes an error, for example, specifies the LDAP password incorrectly, the following messages are displayed:

```
ICH70008I IBM MFA Message:
          MAA1003E Micro Focus MFA authentication was unsuccessful...
          LDAP_PASSWORD_WRONG
```

Because **Chain A** is the first chain that is defined for event **Mainframe logon**, the user can either specify or omit the chain sequence number in the Password field. (The sequence number defaults to 1 if not specified.) Thus, both the following are valid password specifications for Chain A:

- *LDAP_password,TOTP_code*
- 1,*LDAP_password,TOTP_code*

If the user has enrolled in the methods for Chain A and wants to log on with the methods for Chain B (the second chain that is defined for event **Mainframe logon**), the user must

specify the sequence number for the second chain that is defined for the event. For example:

2,*LDAP_password,smartphone_code*

If the user has not enrolled in the methods for Chain A, Chain A would not be presented as available. Thus, Chain B would become the first available chain for that user, and the user would not have to specify the sequence number, which would be 1 in this instance. This scenario is likely to be the case at most customer sites, where only one chain may be in use.

# Passphrase vs Password Support

Passphrase support has been enabled in the examples given in the preceding section. If possible, you should enable passphrase support in the LPARs in which you intend to install AACZ.

Passphrase support extends the length of the Password field of the TSO log-on screen to 100 characters. If passphrase support is not enabled, the Password field can have a maximum number of 8 characters, along with the New Password field that provides another 8 characters.

If passphrase support is not enabled, the log-on process is cumbersome and may be error prone. In this case, the first 8 characters that the user enters are placed in the Password field on the TSO log-on screen. The next characters (up to 8) are placed in the New Password field. AACZ concatenates the contents of the two fields and treats the two entries as if a single passphrase had been entered.

# SSL Setup Requirements

Actions must be taken to enable secure communications between the AACZ started task and the NetIQ AA Server. For example, if using SSL then you need to set up the certificate and make it known to the RACF database in a way that the AACZ started task can find it. Consult the relevant communications documentation for further information.

In addition, for SSL, the AACZ started task userid needs access to the profile that secures the SSL digital certificate. This profile is in the FACILITY class and is IRR.DIGTCERT.LISTRING.

# ICSF Requirements

ICSF is generally available on all mainframes but may not have been configured. If or when it is enabled, you will need to permit the AACZ started task userid access to the relevant profiles to allow it to use the ICSF facilities. These profiles are in the CSFSERV class. The AACZ started task userid needs READ access to the CSFOWH and CSFIQA profiles.

# Tracing Considerations

All communications (which include the passing of passwords) with the NetIQ AA Server will take place using SSL. Standard tracing causes permanent passwords to be redacted in the trace output.

# Out-of-Band Support

NetIQ AA Server provides out-of-band (OOB) support for AACZ. Out-of-band support provides the z/OS Connector with the capability to authenticate either using a browser web application (as of v6.3.5) or via the desktop Windows application running the NetIQ Authentication Agent at Windows startup.

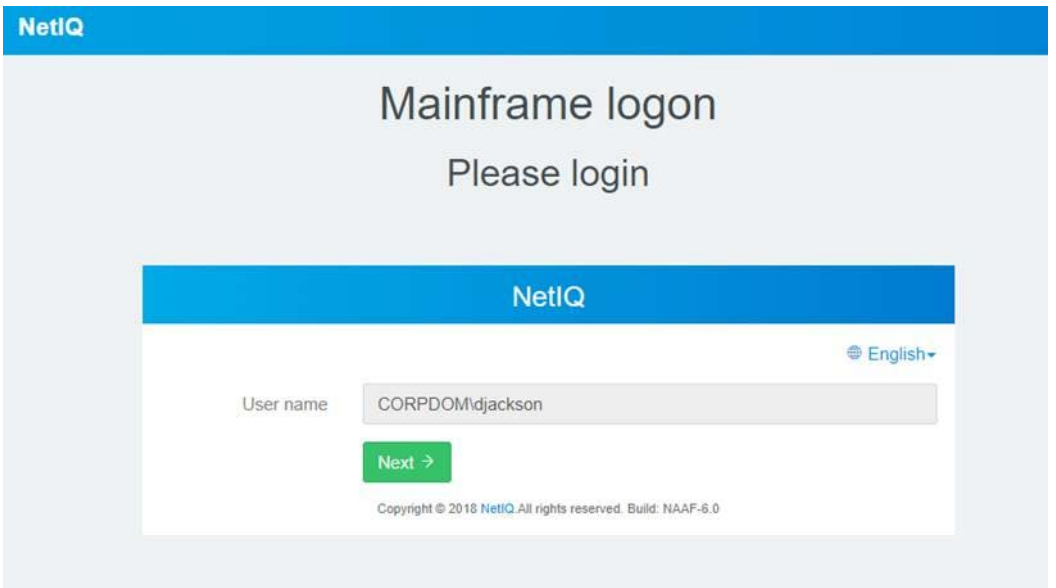
Two separate mechanisms are available, either by using the OOB method (recommended), which is administered as any other authentication method (see the NetIQ AA Server documentation for more details), or by using the legacy dedicated windows agent process.

For completeness, the legacy process is described here but it is recommended that you use the OOB method.

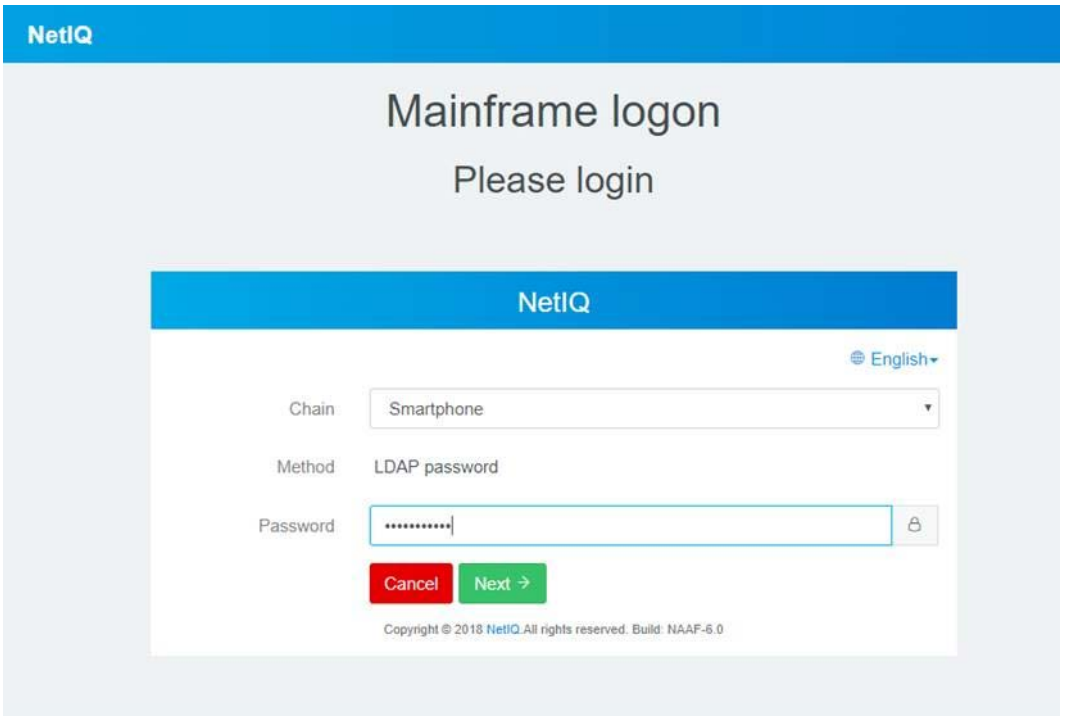
This facility makes a number of additional authentication methods available to the mainframe user. Some of these additional methods are SmartCards, Bluetooth, Fingerprint or Iris scanners, and others.

## Example

Once the Authentication Agent is enabled, the user signs on to mainframe as normal but enters **oob** for the password. AACZ will then authenticate with the NETIQ AA Server and the desktop Authentication Agent Client. The desktop Authentication Agent Client then displays the log-on mainframe image on the user's desktop.



The user enters the User name and presses Next. In this example, the Smartphone Chain and LDAP Password Method are in effect, and the following image is displayed on the user's desktop:




However, the user can select from the drop-down list any other chains that are available.

In this example, the user enters the LDAP password and presses Next. If the LDAP password is accepted, the second method is invoked and the user's smartphone will receive a push message to accept or deny the logon. The Windows Authentication screen will then clear and the Mainframe logon will proceed.

## AACZ Started Task Timeout

You can set the amount of time that you want the AACZ started task to keep trying the OOB logon before the task times out with the OobTimeOut start-up parameter. See the OobTimeOut parameter in "Start-Up Parameters" on page 12 for details.

If the OOB request times out, you will see the same message as if the OOB authorization request is rejected during the log-on process:

ICH70008I IBM MFA Message:

> MAA1003E Micro Focus MFA authentication was unsuccessful...
>
> Out-Of-Band authentication failed or timed out.

# Chapter 3

# Bypassing Multi-Factor Authentication

Normal MFA processing may be bypassed either by using a pass ticket or, using RACF profiles, specifically allowing one or more users to bypass the process.

Pass tickets are generated by the application using services provided by the security software (e.g. RACF). They are used to allow the application to confirm that the user has already been authenticated and the MFA process is not needed again. For example, when logging on to a multi-session manager the full MFA process is employed to authenticate the user. When the manager subsequently logs on to one or more dependent applications it can use a pass ticket to allow those logons to proceed without going through the MFA process all over again.

Beyond pass ticket support, AACZ also provides a mechanism, using RACF profiles, for selectively bypassing the MFA process. If MFA is bypassed, the standard RACF password is used to validate the log-on attempt.

For illustration purposes, this chapter looks at two specific scenarios:

- An attempt to access z/OS resources with IBM Explorer for z/OS.
- An attempt to log on to TSO.

# Accessing z/OS Resources with IBM Explorer for z/OS

IBM Explorer for z/OS is a desktop application that connects to the z/OS started task that hosts the IBM® Rational® Developer for System z® Remote System Explorer (RSE) Daemon. This started task then contacts RACF to authenticate the user's log-on attempt. This started task, like any other z/OS started task, runs with a RACF userid of its own. It also identifies itself to RACF with the application name of FEKAPPL.

AACZ provides a trace facility that displays the following attributes for any MFA-enabled user log-on attempt. In the example shown below, MFA-enabled userid SDOWNE2 is attempting to use IBM Explorer for z/OS to access z/OS resources.

The userid assigned to the RSE Daemon started task is RSED900. The following trace command was used to ask AACZ to provide trace information for any log-on attempt made by SDOWNE2:

```
MODIFY aaczstc,TRACE=YES,USER=SDOWNE2
```

where *aaczstc* is the AACZ started task name.

When the attempt to log on is made, the MAADEBUG ddname output shows the following (extraneous trace information has been omitted for brevity):

```
20171102 09465983 SDOWNE2 MfaUser : SDOWNE2
20171102 09465983 SDOWNE2 Fn1Appl : FEKAPPL
20171102 09465983 SDOWNE2 Fn1User : RSED900
```

where:

- *MfaUser* is the userid that is attempting to log on.
- *Fn1Appl* is the application name that the RSE Daemon passes to RACF.
- *Fn1User* is the userid of the started task hosting the RSE Daemon.

The logic that AACZ uses to decide whether to proceed with the full multi-factor authentication or hand the request back to RACF is as follows.

- If the application (*Fn1Appl*) is not-null/blank, AACZ checks the log-on userid (*MfaUser*=SDOWNE2 in this case) for access to a profile in the RACF class MFADEF with the name MFABYPASS.APPL.FEKAPPL.
- If the userid has READ (or greater) access to this profile, the MFA process is bypassed and control is passed back to RACF for standard RACF password validation. If access to this profile is NONE, the MFA process proceeds.

In general, the profile name checked is:

```
MFABYPASS.APPL.applname
```

where *applname* is the name that the application in question (*Fn1Appl*) passes to RACF.

Generic profiles can be used to manage one or more applications. For example, MFABYPASS.APPL.FEK* would cover all applications starting with FEK.

The application in question may pass the userid under which it is running (RSED900, in this example) instead of an application name. In this case, AACZ can still provide for selective MFA bypass for that application by using its userid. If (and only if) the application name is null/blank, AACZ checks to see if a userid has been passed (*Fn1User*).

If it is non-null/blank, we check the log-on userid for access to a profile in the RACF class MFADEF with the name MFABYPASS.USERID.RSED900. Again, if the userid has READ or higher access to this profile, the MFA process is bypassed.

In this case the general profile checked is:

```
MFABYPASS.USERID.stcuserid
```

where *stcuserid* is the userid of the started task.

Finally, if neither application nor userid is passed by the process requesting authentication, AACZ checks a backstop profile called:

```
MFABYPASS.DEFAULT
```

# Accessing z/OS Resources through TSO

In the case of a user's attempting to log on to TSO, neither the application name nor userid is passed to RACF. The trace items look like this for a TSO log-on attempt:

```
20171102 09540980 SDOWNE2 MfaUser : SDOWNE2
20171102 09540980 SDOWNE2 Fn1Appl :
20171102 09540980 SDOWNE2 Fn1User :
```

In this case AACZ checks the MFABYPASS.DEFAULT profile. If SDOWNE2 has READ access to this profile, MFA is bypassed and control is passed to RACF for password validation.

MFA can be bypassed for everybody attempting to log on to a specific application by making the UACC on the relevant profile READ.

MFA can be bypassed for certain specific users by making the UACC on the relevant profile NONE, but then permitting READ access to that list of specific userids (or the group to which they belong).

Here is the list of sample RACF commands that can be used to set up this facility within RACF:

**NOTE** The MFADEF class must be active and allowed to use generic profiles for the following commands to be effective.

```
RALTER MFADEF GENERIC(ALLOWED)
SETROPTS GENERIC(MFADEF)
```

Then, the class is RACLISTed to make sharing profile updates easier and faster:

```
SETROPTS RACLIST(MFADEF)
```

Then, a set of backstop profiles is set up to ensure that any request not covered by a more specific profile is required to follow the MFA process:

```
RDEFINE MFADEF MFABYPASS.APPL.* UACC(NONE)
RDEFINE MFADEF MFABYPASS.USERID.* UACC(NONE)
RDEFINE MFADEF MFABYPASS.DEFAULT UACC(NONE)
```

Refreshing the RACLISTed class publishes these profiles:

```
SETROPTS RACLIST(MFADEF) REFRESH
```

Then, as required, more specific profiles can be defined. For example, you would issue the following commands to make all RSE Daemon log-on attempts bypass MFA:

```
RDEFINE MFADEF MFABYPASS.APPL.FEKAPPL UACC(READ)
SETROPTS RACLIST(MFADEF) REFRESH
```

As another example, if the default profile MFABYPASS.DEFAULT has a UACC of NONE, all TSO log-on attempts will use MFA. To allow a single userid (for example, SDOWNES) to use RACF password for log-on instead, you would permit that userid access to the default profile:

```
PERMIT MFABYPASS.DEFAULT CLASS(MFADEF) ID(SDOWNES)
    ACCESS(READ)
```

# Chapter 4

# Using your SAF Password in Conjunction with MFA

In normal operations, user authentication using MFA devolves all responsibility for that authentication to the MFA server, which is the NetIQ Advanced Authentication server in this case. The local security software (RACF, etc.) password is no longer used in the authentication process.

However, you may wish to *add* the MFA process to the local security software authentication, instead of replacing it. This may be useful if you ever envisage having to fallback to using only the local software authentication process (that is, no MFA), as keeping the password checking in the loop means that the local password is kept current and *known* by the user. Thus use of the PWFALLBACK MFA option makes sense for this configuration.

This variation in the MFA authentication process is known as "compound In-Band" and can be implemented by the AACZ started task by using the `Compound=Standard` start-up parameter in conjunction with the `SafOnly` start up parameter.



## Using the compound In-Band mechanism

Use the Compound=Standard start-up parameter to have the AACZ started task implement the compound In-Band mechanism. This start-up parameter works in conjunction with the `SafOnly=` parameter as follows:

```
SafOnly=Yes
```

With this parameter specified, any failure in the NetIQ AA server authentication process will result in the local SAF authentication process being allowed to proceed as if the AA server authentication had been passed.

Failure, in this context, means a process failure of some kind (an abend in the MFA stc software, failure to contact the AA server, etc.). It does not apply to the AA server positively denying the authentication request.

Messages will be written to sysprint of the MFA stc showing where the failure occurred and for which userid (which may need to be passed on to Micro Focus support). A WTO will be written to the job log. For example:

```
03.45.05 S0281663  MAA0098E Failure in NetIQ AA server
authentication process - reverting to SAF authentication only.
```

If this parameter is not active, then any process failure will result in authentication failure.

This mechanism requires the use of passphrases in order to allow the user to enter both the SAF password/phrase and the response required by the NetIQ AA server authentication chain methods.

The SAF password/phrase is entered first followed by a colon delimiter and then the AA server response, i.e.

```
<password>:<mfa response#1>,<mfa response#2>, etc
```

For example, if the local SAF password is MYPASS99 and the default MFA authentication chain requires an LDAP password of WiderNetPassword followed by a generated TOTP of 123456, the user would enter:

```
MYPASS99:WiderNetPassword,123456
```

Using the same example except this time supposing that the MFA response chain we wish to use is number 2 on the list of alternatives (instead of the default), the same response would look like this:

```
MYPASS99:2,WiderNetPassword,123456
```

Be careful when using this mechanism if the MFA authentication process does not require users to enter a response (for example, you are using the OOB method). In this case, the user must make sure that their response qualifies as a passphrase (that is, it must be longer than 8 bytes).

For example, say the SAF password is PASS (this is not a good password!). Then the user would have to enter something like:

```
PASS:::::
```

Only the first colon is required; the other characters could be anything as long as they pad the response out to at least 9 bytes. For example:

```
PASS:6789
```

The underlying process works out how long the password is (that is, all characters from the beginning of the passphrase to the first colon). Then it ignores the first colon and uses the rest of the passphrase as the response to the NetIQ AA server authentication process. The AA server process happens first. If that authentication fails, the logon request is stopped with an appropriate message. If it passes, control is passed back to SAF with an indication of the password length (n bytes). SAF then uses the first n bytes of the passphrase as its own password/phrase and proceeds to validate or fail this password or phrase as normal. It also processes requests to change this password or phrase.

## Examples of the process in action

The following examples have been processed against a userid which has the following MFA segment:

```
MULTIFACTOR AUTHENTICATION INFORMATION:
--------------------------------------
 PASSWORD FALLBACK IS ALLOWED
```

```
  FACTOR = AACZ
    STATUS = ACTIVE
```

**NOTE** In this case the SAF userid is the same as the userid registered with the NetIQ AA server. If this were not the case then the MFA segment would have to look like this:

```
MULTIFACTOR AUTHENTICATION INFORMATION:
---------------------------------------
  PASSWORD FALLBACK IS ALLOWED
  FACTOR = AACZ
    STATUS = ACTIVE
    FACTOR TAGS =
```

AAUSERID:aauserid

If the AACZ MFA stc is not up and running, the `PASSWORD FALLBACK` attribute in the MFA segment allows logon authentication to be performed by the local SAF software only. In this case, if you enter a compound response (that is, a password and MFA response), then your logon will still fail. However, if you enter just your SAF password or phrase, you will be allowed to logon.

If all is working well with the AA server and the MFA stc, logon processing proceeds as expected.

If, however, there is some problem in the authentication process (in this case we specified an invalid server address) and `SafOnly=Yes` has not been specified, the user will see something like this example:

```
ICH70008I IBM MFA Message:
          MAA1013E Failure in authentication process...
          Http connection to advanced authentication server failed
***
```

When we start up the AACZ stc with an invalid server address and `SafOnly=Yes`, this is what we see in SYSPRINT:

```
Micro Focus(R) Advanced Authentication Connector for z/OS(R) - version
    1.2

Input parameters:
-----------------
Server=xxx.xxxxx.xxx
KeyRing=CAAD/AARING
Trace=NO
Revoke=NO
Delimiter=,
ConcurrentAuth=50
Compound=Standard
SafOnly=Yes

Options in effect for this instance:
------------------------------------

Authentication server name is   https://xxx.xxxxx.xx
                      on port   Default
                 using protocol   SSL   KeyRing=CAAD/AARING
      Authentication event is   Mainframe logon
   Out-of-Band time out value   00240
  Multi-response delimiter is   ,

   Note: MAXPROCUSER value is   00000100
      Max concurrent auths   00000050

Tracing is not active
Authentication failures will not update the revoke count.
Standard compound in-band process is being used.
AA server will authenticate first and then ask SAF to validate the
    password.
Failure in AA process will revert to SAF-only authentication.
```

When a user attempts to logon, the AA server authentication process will fail because of the invalid server name. We see this in SYSPRINT in the AACZ stc:

```
20210728 03450515 SDOWNES  Http error at: Auth1200
20210728 03450515 SDOWNES          HwtRC: 00000106
20210728 03450515 SDOWNES    HwtDiag_rsn: 001C0001
20210728 03450515 SDOWNES    HwtDiag_sno: 00000001
20210728 03450515 SDOWNES    HwtDiag_txt: EDC9501I The name does not
    resolve for the supplied parameters.
20210728 03450515 SDOWNES         Reason: Http connection to advanced
    authentication server failed
```

In the joblog of the AACZ stc we also see this:

```
03.45.05 S0281663  MAA0098E Failure in NetIQ AA server
authentication process - reverting to SAF authentication only.
```

As long as the SAF password/phrase is entered correctly as part of the compound response (that is, the characters up to the first colon in the response), the logon proceeds successfully, as validated by the local SAF software only.

## Examples of messages from TSO logons:

### *Compound=Standard and SafOnly=Yes*

When deliberately setting the AA server address incorrectly, with the user entering a correct RACF password and AA server response, the user logging on sees this message:

```
ICH70008I IBM MFA Message:
          MAA1015I The NetIQ AA server process failed.
          However, your password/phrase will now be passed to SAF.
ICH70001I SDOWNES  LAST ACCESS AT 07:19:14 ON WEDNESDAY, JULY 28, 2021
***
```

The MAA1015I message is from the AACZ software and the ICH70001I message was produced by requesting RACF to validate the users password.

### *Compound=Standard with AA server working correctly*

When the user has supplied an incorrect RACF password but sees a correct AA server response, this message is displayed.

```
ICH70008I IBM MFA Message:
          MAA1012I First stage of compound logon authenticated by
          Micro Focus(R) Advanced Authentication Connector for z/OS(R)
***
```

The MAA1012I message is from the AACZ software and indicates that we have successfully processed the AA server authentication. We are now going to pass the (invalid) password to RACF.

The next panel is the TSO logon panel with the usual message given for an invalid RACf password.

### *Compound=Standard with the AA server working correctly and both the RACF password and the AA server response are valid.*

When the user has supplied the correct RACF password and sees a correct AA server response, this message is displayed.

```
ICH70008I IBM MFA Message:
          MAA1012I First stage of compound logon authenticated by
          Micro Focus(R) Advanced Authentication Connector for z/OS(R)
ICH70001I SDOWNES  LAST ACCESS AT 07:28:39 ON WEDNESDAY, JULY 28, 2021
***
```

The MAA1012I message is from the AACZ software. We then pass the (valid) password to RACF, which issues the ICH70001I message and the logon proceeds.

# Chapter 5

# Messages

The AACZ can issue the following messages. They will appear in both the JOBLOG of the AACZ started task and in SYSLOG.

**MAA0001E** `Not APF authorized - terminating`

**Explanation:** The started task procedure has a steplib which is not APF-authorized, execution cannot continue.

**MAA0002E** `Non-Micro Focus MFA PC-routine is already established, terminating.`

**Explanation:** The method by which RACF communicates with the MFA process is already in use by some other started task or job. Execution cannot continue.

**MAA0003E** `MAAMAIN has been unable to open/process SYSIN parameters, terminating.`

**Explanation:** Either the SYSIN ddname has not been specified in the started task procedure or the file has not been opened successfully. Execution cannot continue.

**MAA0004E** `EndPoint dataset not allocated (MAAENDPT) - terminating`

**Explanation:** The endpoint id and secret are held in the file allocated to ddname MAAENDPT in the started task procedure. We have been unable to find this ddname in the current started task execution. We cannot continue.

**MAA0005I** `Micro Focus® Advanced Authentication Connector for z/OS® version` *`v.r`* `enabled`

**Explanation:** Initialization of MFA support was completed successfully for the specified release (v=version, r= release within version).

**MAA0006E** `Unable to open Endpoint dataset (MAAENDPT) - terminating.`

**Explanation:** MAAENDPT has been allocated but we have been unable to open it. Execution cannot continue.

**MAA0009E** *`xxxxxxxx`* `of token` *`xxxxxxxxxxxxxxxx`* `failed, R15=`*`xxx`*

**Explanation:** Token services failure. The first *xxxxxxxxx* will be replaced by the action being attempted (Create, Retrieve, Delete). The second by the name of the token being acted on. The MFA initialization process cannot complete and execution is terminated.

**MAA0011E** `Micro Focus MFA facility is no longer installed. Terminating without deleting token.`

**Explanation:** During AACZ started task termination we attempt to uninstall our interface but have found that it is no longer in place. This is informational. We carry on to terminate anyway.

**MAA0012I** `Micro Focus® Advanced Authentication Connector for z/OS® is disabled`

**Explanation:** We have disabled our interface as part of normal termination.

**MAA0013I** `Waiting for MFA subtasks to complete`

**Explanation:** During normal termination we have found active log-on requests in progress. No new log-on requests will be processed but we wait while those in flight complete. This message may be issued up to 10 times with a 1 second wait between each scan for active log-on processes. When no active processes are found termination will complete.

**MAA0014I** `Unilateral termination proceeding, subtasks still active.`

**Explanation:** After 10 iterations of waiting for active processes to complete, one or more are still in flight. We will terminate anyway. In-flight logons may end abnormally.

**MAA0015E** `Unable to set DUBPROCESS rc/rsn is: xxxxxxxx / xxxxxxxx - terminating.`

**Explanation:** This is an indication that the Unix System Services environment is not correct or not available.

**MAA0016W** `Concurrent authentication demand exceeded maximum, HWM: nnnnnnnn`

**Explanation:** The rate at which requests for authentication reached the AACZ started task would have required more subtasks active concurrently than can be supported. This has resulted in some authentication requests being delayed. This is not a problem if this is an infrequent occurrence, but if it is issued regularly you may want to review your MAXPROCUSER setting. Alternatively, if you are deliberately throttling the concurrency rate (using the ConcurrentAuth sysin parameter), then this message is to be expected and can be ignored.

**MAA0099E** `Recovery requested but no MAASNAP DD statement was available.`

**Solution:** In order to provide diagnostic information for an abnormal end, the MAASNAP DD statement is required. Add this to the AACZ started task JCL ( //MAASNAP DD SYSOUT=* ).

The following messages may be issued to the user who is logging on via MFA implemented by this product. IBM provides the first message. Our messages will follow on from this message:

`ICH70008I`

IBM MFA Message: If the logon authentication has been successful, the user will see:

**MAA1001I** `Logon successfully authenticated by Micro Focus® Advanced Authentication Connector for z/OS®`

**MAA1002E** `Logon attempt was rejected by Micro Focus® Advanced Authentication Connector for z/OS®...`

**MAA1003E** `Logon attempt was rejected by Micro Focus® Advanced Authentication Connector for z/OS®...`

**Explanation:** There are a number of different reasons why authentication may have failed. If it is a genuine authentication failure (for example, wrong password), you will see message MAA1002E or MAA1003E.

The difference between the MAA1002E and MAA1003E message is that the first case results in the revoke count being incremented for this attempt, while the second does not (see the REVOKE= sysin parameter, Chapter 2, "Start-Up Parameters" on page 12). This message will be followed by the reason for the failure as supplied by the NetIQ AA Server. For example:

```
ICH70008I IBM MFA Message:
          MAA1003E Logon attempt was rejected by
          Micro Focus® Advanced Authentication Connector for z/OS®
          LDAP_PASSWORD_WRONG
***
```

Messages MAA1004E through MAA1007E indicate different reasons for the failure in the MFA authentication process.

**MAA1004E** `Failure in authentication process...`

**Explanation:** A session could not be established with the AA Server.

**MAA1006E** `Failure in authentication process...`

**Explanation:** The attempt to perform the AA Server log-on process failed.

**MAA1007E** `Failure in authentication process...`

**Explanation:** Unable to extract method information (probably because the user is not enrolled in one or more methods for the chosen chain or has specified an invalid chain number).

**MAA1008I** `MFA authentication has been bypassed.`

**Explanation:** The application is which you are logging on allows MFA to be bypassed if your userid has access to the relevant security profile. You have such access and the authentication process has thus been bypassed.

**MAA1009I** `Authentication successful using a passTicket`

**Solution:** A pass ticket, generated by a previously fully authenticated process, has been used to grant access to the application. The normal MFA process has been bypassed.

**MAA1010E** `AACZ task too busy (MAXPROCUSER), please try later.`

**Solution:** There is too much concurrent authentication activity for the local environment. This may mean that the lpar MAXPROCUSER setting is too low (see your systems programmers). The AACZ started task will attempt to overcome bursts in concurrent activity by delaying authentication requests and retrying automatically. However, if this has not been possible within a 30 second timeframe, the authentication request is denied with this message.

**MAA1011E** **abnormal end, please review output in the AACZ stc.**

**Solution:** The authentication process has abended. Diagnostic information will be written by the AACZ started task. Storage will be snapped to MAAPRINT, messages may be written to SYSPRINT and to the JobLog. If IBM components such as LE have abended then they will also produce output (e.g. CEEDUMP). This information should be gathered and reported to Micro Focus support.

**MAA1012I** `First stage of compound logon authenticated by Micro Focus® Advanced Authentication Connector for z/OS®`

**Solution:** Compound in-band standard process has resulted in the AA server successfully authenticating your MFA credentials. The local SAF software (e.g. RACF) will now be asked to validate your password or phrase.

**MAA1013E** `Failure in authentication process…`

**Solution:** The AA server has not been contactable. See output in the AACZ stc for more details.

MAA1015I The NetIQ AA server process failed. However, your password or phrase will now be passed to SAF.

**Solution:** Compound in-band standard process has encountered a failure in the AA server process (see output in the AACZ stc). However, SafOnly=Yes is active so the password/phrase is being passed to SAF (e.g. RACF) for validation anyway.

MAA1099E Error in Micro Focus MFA validation process at point: *xxxxxxxx*

**Explanation:** This is the catch-all message that will be issued if the process goes wrong in an unexpected fashion. It will be issued as a WTO (all the other messages so far have been TPUTs, actually issued by RACF. We pass the message to RACF which is why we get the ICH7008I message first). This will be output to the user who is logging on and written to SYSLOG. The *xxxxxxxx* will contain program location information of use to the developer looking into the cause of the failure.

The following messages may be written to the SYSPRINT ddname of the AACZ started task:

MAA2001E Unable to extract endpoint data

**Explanation:** The log-on process cannot continue and the authentication request is failed. The user will see message MAA1004E.

MAA2002I Endpoint not found. We will attempt to re-establish it.

**Explanation:** We have successfully contacted the NetIQ AA Server but the requested endpoint has not been found. The AACZ started task will now request it be created and will store the resulting id and secret in the MAAENDPT data set. This may happen if the NetIQ AA Server administrator deletes the endpoint for some reason, usually because the endpoint id and secret have become known. Authentication processes will re-commence once the new endpoint is established.

MAA2003E Unable to establish endpoint session

**Explanation:** We have failed to establish a new endpoint at the NetIQ AA Server. The authentication request is denied and the user will see message MAA1004E.

MAA2004E Unable to extract method for user

**Explanation:** The attempt to extract information about an authentication method for a user has failed (usually, user not enrolled in one or more methods in the chain). The authentication request is denied and the user will see message MAA1007E.

MAA2005E Logon failed

**Explanation:** The attempt to process the logon for the current method has failed. The authentication request is denied and the user will see message MAA1006E. Tracing facilities are available if it proves difficult to discern why a failure in process has happened. (See Chapter 2, "Start-Up Parameters" on page 12.)

MAA3001I Authorization successful using chain: *chain*

**Explanation:** We have date/time stamp followed by the userid attempting to logon, the message number, and text. For a successful logon the message is ended with the name of the chain used to authenticate the logon.

MAA3002I Authorization denied: *reason*

**Explanation:** We have date/time stamp followed by the userid attempting to logon, the message number and text. For an unsuccessful attempt we echo the reason for the failure as provided by the NetIQ AA Server.

MAA3003I MFA bypassed (returned to SAF) for: <applname/userid>

**Explanation:** The user and application qualified for MFA to be bypassed - control is returned to SAF (e.g. RACF) with an indication that this has happened.

MAA3004I Authentication successful using a passTicket

**Explanation:** The application has generated a pass ticket, which has been accepted by the security software (e.g. RACF) for this logon. The normal MFA process has been bypassed.

MAA3005I Out-Of-Band authorization successful.

**Explanation:** The authentication process was successful using an out-of-band mechanism.

# Appendix A

# Uploading the MFAACZ Distribution Files to the Host

This appendix describes how to upload the compressed MFAACZ distribution files to the host and decompress the uploaded files into libraries.

# Unloading the Product Media

Installation Summary

Whether you install the Micro Focus Advanced Authentication Connector for z/OS from a physical CD-ROM or from a downloaded installation program, the process of moving files from the product media to the mainframe host is essentially the same.

**NOTE** The userid under which you are signed on to your workstation must have administrator privileges to run the installer program.

First, you will run the installer program (AACZ*vrm*Setup.exe) to unload the product binaries to an uploadable format on a PC that has TCP/IP connectivity to the host. Copying the files to your local PC decreases the transfer time to the host and decompresses the files from their distributed format.

Next, you will edit and execute a generated batch file with the FTP commands needed to upload the product files to the host.

Finally, you will run the included JCL **RECEIVE** job on the mainframe to decompress the uploaded product files and install them in a library.

Prerequisites

This process has the following prerequisites:

- TCP/IP FTP connectivity from your PC to the host computer.
- A TSO userid and password on the host.
- Security authorization to allocate files on the host.
- The IP address of the host.

Customization and Setup

Subsequent code customization and rollout to your production libraries are discussed in earlier chapters of this manual.

## Copy the files to your computer

PC Setup Procedure

1. Run the setup program (AACZ*vrm*Setup.exe) software on your host-connected PC to unload the product distribution media. Depending on the source of the media, do one of the following:
   a. Insert the Micro Focus Advanced Authentication Connector for z/OS CD-ROM in your CD drive. The setup program will start automatically. Or:
   b. Download the self-extracting media file (AACZ*vrm*Setup.exe) to a local hard disk on your PC, then run it. Take the default when prompted for an extraction location. The setup program then starts automatically.

**IMPORTANT! Do not run the setup program from a network drive.** Problems can occur if the installer needs to access files from the network drive and the network connection is not available or if access permissions are not set up correctly.

2. If you previously installed the Micro Focus Advanced Authentication Connector for z/OS on the installation PC and did not remove the old product files, the first dialog box that appears allows you to remove them. If this occurs:
   a. Click **Remove** to remove the old product files, then click **Finish**.

**b** Restart the installer again from Step 1.

**3** When the End User License Agreement displays, select "**I accept the terms in the License Agreement**" and click **Next**.

**4** When prompted for an operating system, select the desired target operating system on the host and click **Next**.

**5** When the Destination Folder dialog appears, do one or more of the following:

- To change the default location where the product media will be unloaded on the PC, click **Change**. Or:
- To accept the target install location on the PC, click **Next**.

Data Set HLQs

**6** The next dialog box requests **Host FTP and Transmission Information**. The information you provide is used to generate automated scripts that transfer the product files from the PC to the host. Fill in the dialog to use the automated file transfer process. The following values are required:

**Host Name or IP** — DNS Network name or an IPV4 dotted decimal address or IPV6 address enclosed in square brackets — for example:

192.168.0.1

or

[2001:1890:1112:1::20]

**Host Logon ID** — Your TSO userid.

**Transmitted File HLQ** – Specify the high-level qualifier to be added to the host XMIT data set names (that is, the sequential data sets transmitted from the PC to the host).

**PDS Library HLQ** – Specify the high-level qualifier to be added to the host RECEIVE data sets (that is, the PDS libraries created when the RECEIVE command expands the XMIT files uploaded from the PC).

**CAUTION!** The PDS Library HLQ *must be different* from the Transmitted File HLQ.

Rules for HLQs

Follow these rules when entering the high-level qualifiers:

- Do not choose high-level qualifiers that create data set names that already exist on the host. The XMIT and RECEIVE processes overlay existing data sets.
- Your host userid must have authority to allocate files with these names.
- Do not add leading or trailing periods.
- Do not use parentheses or quotes (single or double).
- The maximum length of the total data set name (high-level qualifier plus Micro Focus Advanced Authentication Connector for z/OS library name) is 44 characters.
- Case is not significant. All entries are normalized to upper case.

When you are satisfied with your entries, click **Next**.

**7** To change any of your previous entries, click **Back**. To execute your set-up instructions, click **Install**.

8 If you would like to view the Readme file at this time, click the checkbox to the left of the Open Readme instruction.

Click **Finish**.

The set-up process is complete. You are now ready to transfer files from the PC to the mainframe.

# Automated FTP File Transfer to the Host

Automated Upload Scripts

If you provided the requested information in the **Host FTP and Transmission Information** dialog (Step 6 on page 47), follow the instructions in this section to perform an automated transfer the product files to the host.

## Edit the FTP Input File

1 Go to **Start | All Programs | Micro Focus| Advanced Authentication Connector for z/OS | Edit FTP Input**, which displays the `FTP.bat` control file that FTP uses to transfer the compressed Micro Focus Advanced Authentication Connector for z/OS XMIT files to the host. (Open this file with Notepad.)

2 Edit the third line in the FTP control file, replacing the text `<PASSWORD>` with your own password on the host.

**IMPORTANT!** An incorrect password value in the FTP control file will trigger a "login error" message.

3 Save the file under the same name.

## Edit the RECEIVE Job

The installation creates a *somnode*`.RECEIVE.AACZ`*vrm*`.JCL` file on the host using the *somnode* high level qualifier that you specified for Transmitted File HLQ in Step 6 on page 47. This file contains JCL to receive the files on the host.

You must add your JOB statement to this file in order for the `RECEIVE` job to run. You may edit the file now to add your JOB statement or through your TSO session after the files have been uploaded to the host.

If you want to edit the file now, do the following:

1 Go to **Start | All Programs | Micro Focus| Advanced Authentication Connector for z/OS | Edit Receive JCL**. (Open the file with Notepad.)

2 Add your JOB statement at the top of the file.

3 Save the file under the same name.

## Transfer Files to the Host

After you edit the FTP input file by adding your password, you are ready to transfer files to the host. Note that your local system must be connected to the network for this transfer to work.

To transfer the files:

1 Select **Start | All Programs | Micro Focus | Advanced Authentication Connector for z/OS | FTP Files To Host**.

2 A window displays with the message "FTP process has begun". This process can take several minutes to complete, depending on the size of the files that are transmitted.

3 After the transfer completes, view the FTP log file to verify that the files were transferred to the host. To do this, select **Start | All Programs | Micro Focus| Advanced Authentication Connector for z/OS | View FTP Log.**

   If any of the following errors appear, you must make corrections in the FTP Input File and repeat Step 1 above.

| Message | Corrective Action |
|---|---|
| Invalid Command | Verify that you entered the correct user name and password in the FTP input file. |
| Login Error | Verify that you entered the correct user name and password in the FTP input file. |
| Not Connected | The FTP connection was lost so these files did not get uploaded. Resubmit the program by choosing FTP Files To Host. |
| Unknown Host | Check the IP address of your host and edit the FTP Input file. |

## Deleting Your Password

Delete your password, which you added in Edit the FTP Input File on page 48, from the `FTP.bat` file.

# Submitting the Host RECEIVE Job

After you complete a valid FTP transfer, you need to issue a `RECEIVE` command for the XMIT files on the host. To do this:

1 Log on to your TSO session.

2 Find the *somnode*`.RECEIVE.AACZ`*vrm*`.JCL` file.

3 If you did not edit the *somnode*`.RECEIVE.AACZ`*vrm*`.JCL` file on the PC ("Edit the RECEIVE Job" on page 48), edit the file now and add JOB statement information.

4 Submit the job contained in the file.

5 Examine the libraries created with the host high level qualifier (see Step 6 on page 47) to ensure that they are PDS libraries. They may not have been created

successfully despite a zero return code or a job SYSOUT message that says, "Restore successful to dataset."

# Post-Upload Cleanup

After the product files have successfully been uploaded to the host, perform the following clean-up steps on the installation PC.

## Deleting Your Password

To prevent a mainframe security breach, delete your password from the FTP Input file.

1. Go to **Start | All Programs | Micro Focus| Advanced Authentication Connector for ZOS | Edit FTP Input**
2. Delete your password from the source code.
3. Re-save the file.

## Removing Product Files

If you wish to remove the product files from the installation PC when you are finished, do the following:

1. Restart the installer software from Step 1 on page 46. The first dialog box to display prompts you to remove the product files.
2. Choose the **Remove** option to remove the product files, then click **Finish** to exit.

This step is optional.

# Index