

Automated Sign-On for Mainframe Administrator Guide

12.4 SP1

© 2017 Attachmate Corporation, a Micro Focus company. All rights reserved.

No part of the documentation materials accompanying this Attachmate software product may be reproduced, transmitted, transcribed, or translated into any language, in any form by any means, without the written permission of Attachmate Corporation. The content of this document is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Attachmate Corporation. Attachmate Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this document.

Attachmate, the Attachmate logo, and Reflection are registered trademarks of Attachmate Corporation, in the USA. All other trademarks, trade names, or company names referenced herein are used for identification only and are the property of their respective owners.

Contents

Automated Sign-On for Mainframe	5
1 Introduction	7
Overview: How it Works	8
Before you Begin	9
System Requirements and Prerequisites	9
Terms	11
2 Configuration Tasks	13
Summary of Configuration Tasks	13
3 Phase I: Set Up Initial Configuration	15
1.1 Install Management and Security Server and the Automated Sign-On for Mainframe Add-On.	15
1.2 Configure user authentication and authorization.	16
1.3 Install and configure components on z/OS.	17
Configuring DCAS and RACF	17
4 Phase II: Configure a Simple Test Session	19
2.1 Configure the Administrative Server for Automated Sign-On for Mainframe.	19
A. Configure settings to manage certificates and passwords.	19
B. Enter the Automated Sign-On settings.	19
2.2 Create host sessions with the automated sign-on macro.	20
IBM 3270 session for Windows-based emulators	20
IBM 3270 Session for web-based emulators	22
2.3 Assign user access for testing	24
2.4 Run a test.	24
5 Phase III: Set Up and Store Mainframe Username Mappings	25
3.1 Choose a data store option.	25
Option A. An authenticating directory with primary user objects.	25
Option B. An authenticating directory plus a secondary directory.	26
3.2 Implement identity mappings and data storage.	26
Configuration Tasks: Identity Mapping	26
6 Phase IV: Configure Settings for Production	29
4.1 Assign users' access to mainframe sessions.	29
4.2 Deploy automated sign-on to users.	29
A Appendix A. Configuring DCAS and RACF on z/OS	31
Overview of DCAS Configuration and the z/OS Security Server	31
1. Configure RACF so DCAS can run as a system daemon.	32
1.1 Define a user ID as superuser to use OMVS Services.	33

1.2 Provide a user ID with access to MVS.SERVGR.DCAS.	33
1.3 Provide a RACF definition for MVS Start-up.	33
2. Configure TLS for use with DCAS.	33
2.1 Overview of Using System TLS with the DCAS Server	33
2.2 Configure a Client Authentication Level.	34
2.3 Use RACF's Common key ring support to manage keys and certificates.	35
2.4. Create and Connect a Certificate	36
3. Define a PassTicket profile for each application.	38
4. Update the Configuration for the DCAS Server	38
5. Start the DCAS server.	39
Optional Parameters	40
References	41

Automated Sign-On for Mainframe

Automated Sign-On for Mainframe is an add-on to Host Access Management and Security Server and requires a separate license.

When implemented, Automated Sign-On for Mainframe enables an end user to use a terminal emulation client to automatically log on to a host application on a z/OS mainframe.

At a Glance:

- ♦ [Overview: How it Works](#)
- ♦ [Summary of Configuration Tasks:](#)
 - Phase I. Set up initial configuration.
 - Phase II. Configure a simple test session.
 - Phase III. Set up and store mainframe username mappings.
 - Phase IV. Configure settings for production.

1 Introduction

Automated Sign-On for Mainframe enables an end user to use a terminal emulation client to use a terminal emulation client to automatically log on to a host application on a z/OS mainframe.

Automated Sign-On for Mainframe is an add-on product that requires the base installation of Host Access Management and Security Server and its Administrative Server.

To implement Automated Sign-On for Mainframe, configurations are required on:

- ♦ the MSS Administrative Server
- ♦ the terminal emulation client session
- ♦ a data store option
- ♦ z/OS

Configuration must be done by both the Management and Security Server administrator and the mainframe administrator.

About the Administrative Console

Version 12,4 SP1 of Management and Security Server introduces the **Administrative Console** -- the redesigned user interface for the Administrative Server.

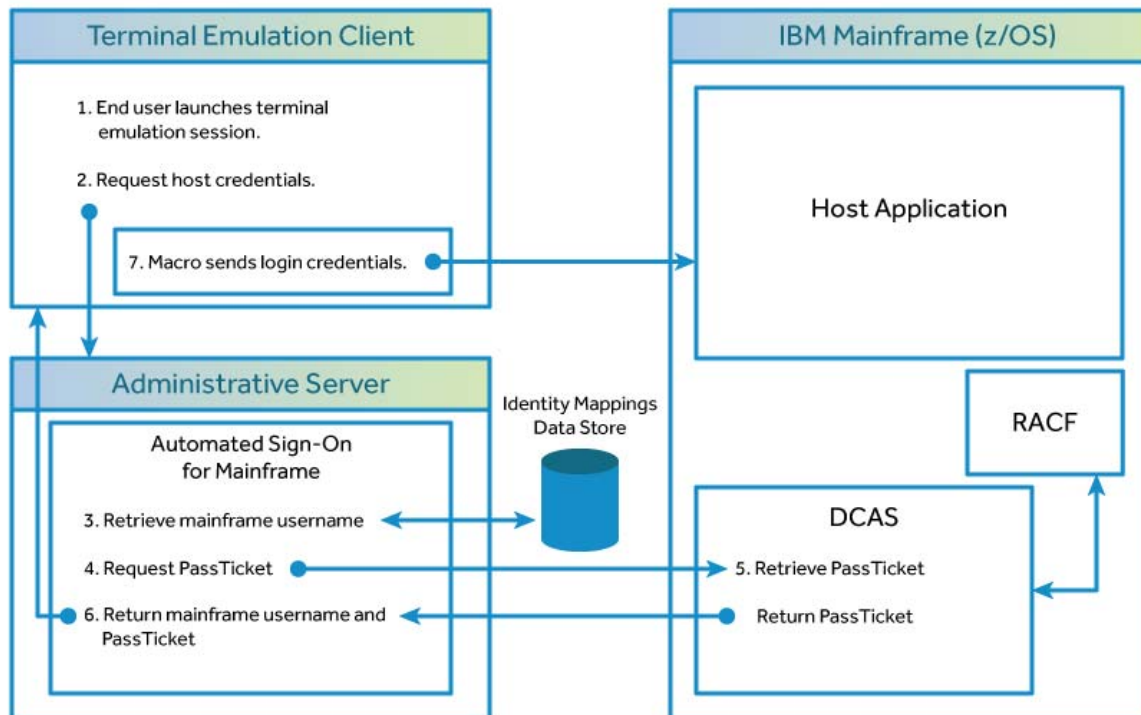
If you are upgrading, you will notice the new navigation:

- ♦ **Manage Sessions** replaces *Session Manager*
- ♦ **Manage Packages** replaces *Package Manager*
- ♦ **Assign Access** replaces *Access Mapper*
- ♦ **Configure Settings** replaces *Settings* and *Security Setup*
- ♦ **Run Reports** replaces *Reports*

Overview: How it Works

Follow the flow of activity from the end user's terminal logon through the automated sign-on to the mainframe application.

Automated Sign-On for Mainframe



- 1 Launch the emulation client (such as Reflection, Rumba, or InfoConnect), and authenticate to the MSS Administrative Server. The client connects to the host, which prompts for the user's credentials.
- 2 The Client requests the user's host credentials from the MSS Administrative Server.
- 3 The MSS Administrative Server retrieves the user's mainframe username from the data store of mapped mainframe usernames.
- 4 The MSS Administrative Server passes the host application ID and the end user's mainframe ID to Digital Certificate Access Server (DCAS) on the z/OS mainframe, and requests a PassTicket.
- 5 DCAS exchanges information with RACF and retrieves a PassTicket, which is then returned to the MSS Administrative Server.
- 6 The MSS Administrative Server returns the user's mainframe username and the PassTicket to the emulation client.
- 7 The terminal emulation client's login macro sends the user's mainframe username and PassTicket to the host application. The user is automatically logged on.

Before you Begin

Look over the [Summary of Configuration Tasks](#) and consider what information to gather before you begin configuring, such as:

- ♦ details about DCAS on the mainframe, which you may need to obtain from your IBM host administrator.
- ♦ which [data store option](#) to use.
- ♦ the [location of the data store](#) where the mainframe usernames are (or will be) mapped to users' authentication credentials.

System Requirements and Prerequisites

The following products and systems must be in place before installing or configuring Automated Sign-On for Mainframe.

- ♦ **Host Access Management and Security Server 12.4 SP1 or higher** on the designated server. Management and Security Server provides the MSS Administrative Server and can be installed on any platform that runs a compatible version of Java.

This version introduces the Administrative Console, and you can use an HTML login that does not require Java.

- ♦ **Micro Focus terminal emulation software** on the client and administrator workstations. Automated Sign-On for Mainframe currently supports:

Reflection Desktop 16 or higher

Reflection 2014 R1 SP1

Reflection ZFE 2.0 or higher

Reflection for the Web 12.3 SP1 or higher (The version must be compatible with Management and Security Server. See the Reflection for the Web [Release Notes](#).)

Rumba 9.4.1 or higher

- ♦ **z/OS** with DCAS installed. (See Section 1.3 for guidance.)
- ♦ **TLS connection** from the MSS Administrative Server to DCAS. The default is **TLSv1.2**.

If the mainframe does not support -- or cannot be updated to -- TLSv1.2, add this MSS server setting to enable TLSv1 or TLSv1.1.

1. In `\MSSData\properties`, open `server.properties` in a text editor.
2. Add a line to enable the TLS protocol versions:

```
server.asm.protocols=TLSv1, TLSv1.1, TLSv1.2
```

NOTE: Syntax matters. Valid property values are `TLSv1`, `TLSv1.1`, and `TLSv1.2`.

(`TLSv1.0` does not work.)

If the property is not specified or is empty, the default value is `TLSv1.2`.

3. Save the file.

- ♦ **LDAP directory** for user authorization.

NOTE: JCE Unlimited Strength Jurisdiction Policy Files are required on the MSS Administrative Server. If you encounter a warning, apply the policy files, as follows.

To apply the policy files:

1. Download the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files from Oracle or IBM. Be sure to download the correct policy file updates for your version of Java:

Java 8: [Java SE Downloads \(http://www.oracle.com/technetwork/java/javase/downloads/index.html\)](http://www.oracle.com/technetwork/java/javase/downloads/index.html)

IBM: [Unrestricted SDK JCE policy files \(https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=jcesdk\)](https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=jcesdk)

2. Uncompress and extract the downloaded file. The download includes a `Readme.txt` and two `.jar` files with the same names as the existing policy files.
3. Locate the two existing policy files:

`local_policy.jar`

`US_export_policy.jar`

On UNIX, look in `<java-home>/lib/security`

On Windows, look in `C:\Program Files\Java\jre<version>\lib\security`

4. Replace the existing policy files with the unlimited strength policy files you extracted.
-

Terms

This list includes brief definitions of terms used in this document.

- ♦ **Administrative Server** (or **MSS Administrative Server**): component installed with Host Access Management and Security Server.
- ♦ **Administrative Console**: the redesigned user interface for the MSS Administrative Server, used to manage and configure terminal sessions.
- ♦ **DCAS** (Digital Certificate Access Server): a TCP/IP server application that interfaces with IBM Resource Access Control Facility (RACF) to return PassTickets, which act as passwords in the automated sign-on process.
- ♦ **MSS**: abbreviation for Host Access Management and Security Server.
- ♦ **PassTicket**: a time-limited, encrypted substitute for a user's password. PassTickets are generated per user for a one-time-only use.
- ♦ **RACF**: IBM Resource Access Control Facility. RACF is a security system that provides access control and auditing functionality for the z/OS and z/VM operating systems.

2 Configuration Tasks

The configuration of Automated Sign-On for Mainframe requires some tasks to be done in the terminal client and some to be done on the mainframe. The configuration tasks are presented in phases to follow the workflow from testing through deployment.

Summary of Configuration Tasks

Use this summary to plan your configuration for testing and production. Detailed steps for each phase are provided in the sections that follow.

Phase I. Set up initial configuration.

First, set up an initial configuration on the MSS Administrative Server and import the **Automated Sign-On for Mainframe Add-On**.

Then, configure the z/OS mainframe to enable support for PassTickets. You may need to consult your IBM host administrator about configuring DCAS and RACF.

Phase II. Configure a simple test session.

Create an IBM 3270 mainframe session and configure it for automated sign-on.

Record and edit a macro, assign user access for testing, and then run a test of Automated Sign-On for Mainframe.

Phase III. Set up and store mainframe username mappings.

Based on your environment, choose a data store option and store the mainframe usernames in a directory so that they are associated with particular users. (This step is not necessary if the mainframe usernames are stored on each user's smartcard.)

Then, create mappings between users and groups in the directory, and map the mainframe usernames for users and groups.

Phase IV. Configure settings for production.

Create and configure the terminal emulation client sessions, and then use **Assign Access** in the MSS Administrative Server to give users and/or groups access to the sessions.

Then, specify the source for the mainframe username to be used for automated sign-on when a user or member of a group runs an assigned session.

3 Phase I: Set Up Initial Configuration

Host Access Management and Security Server must be installed as a core product before Automated Sign-On for Mainframe can be added.

NOTE: The user workstations need to use one of the terminal emulator clients (listed in the [System Requirements](#)) that has the API functionality that enables Automated Sign-On for Mainframe.

In brief, you will:

- ◆ Install the products: **Management and Security Server** and the **Automated Sign-On for Mainframe Add-On**.
- ◆ Configure user authentication and authorization.
- ◆ Install and configure components on z/OS.

1.1 Install Management and Security Server and the Automated Sign-On for Mainframe Add-On.

Automated Sign-On for Mainframe Add-On is provided as an activation file, which must be installed in addition to Management and Security Server. Follow these steps.

- 1 Install (or upgrade to) Host Access Management and Security Server (MSS) 12.4 SP1.
Refer to the [Installation Guide](#) for assistance.
- 2 Check to see whether Automated Sign-On for Mainframe is already installed:
 - 2a In the MSS **Administrative Console**, click **Configure Settings > Product Activation**.
 - 2b In the list of Currently Installed products, look for **Automated Sign-On for Mainframe Add-On**.

Configure Settings - Product Activation

Currently Installed

Additional products can be installed by uploading activation files. Click "Activate New" and select an activation file to upload.

Activation file names have this format: activation.<product_name>.jaw

Activate New	Remove	
<input type="checkbox"/>	PRODUCT	VERSION
<input type="checkbox"/>	Automated Sign On for Mainframe Add-On	12.04.10.016
<input type="checkbox"/>	Management and Security Server for Windows Desktop Emulation	00.01.01.302
<input type="checkbox"/>	Host Access Management and Security Server Framework	12.04.10.016

- 2c** If listed, the activation file is installed. Proceed with the steps to [Configure user authentication and authorization](#).
If not listed, continue with step 3.
- 3** Install the Automated Sign-On for Mainframe Add-On activation file.
- 3a** Download the activation file from the same Micro Focus | Attachmate site where MSS was downloaded.
In the list of product entitlements, click **Management and Security Server--Automated Sign-On for the Mainframe Add-On**. The filename is
`activation.automated_signon_for_mainframe<-version>.jaw`
Note the download location.
- 3b** In MSS, open **Configure Settings >Product Activation**. (See steps 2a-2b above.)
- 3c** Click **Browse** or **Choose File** (depending on your browser) to open a list of available files.
- 3d** Browse to the download location. Click the file:
`activation.automated_signon_for_mainframe-12.4.10.jaw`
The list of **Currently Installed** products now includes **Automated Sign-On for Mainframe Add-On**. Once installed, the configuration settings are available in the MSS Administrative Console > **Configure Settings > Automated Sign-On**.
- 4** Restart your browser to ensure that the MSS Administrative Console is fully updated with the new set of activation files. You do not need to restart the MSS Administrative Server.

1.2 Configure user authentication and authorization.

Automated Sign-On for Mainframe requires users to authenticate to the MSS Administrative Server. Users can authenticate to the MSS Administrative Server by using a smart card, username and password, or other credentials.

NOTE: An LDAP directory must be used for user authorization.

To configure user authentication and authorization:

- 1 In Management and Security Server, open the **Administrative Console to Configure Settings > Authentication & Authorization**.
- 2 Select an **Authentication method**. If users will authenticate to the MSS Administrative Server using smart cards for authentication, select **X.509 with LDAP failover**.

Configure Settings - Authentication & Authorization

Choose Authentication Method

Authentication method

- None
- LDAP
- Single sign-on through IIS
- Single sign-on through Windows authentication
- X.509 with LDAP failover

- 3 For Authorization method, select **Use LDAP to restrict access to sessions**.

Choose Authorization Method

Authorization method

- Allow authenticated users to access all published sessions
- Use LDAP to restrict access to sessions

- 4 Enter the required information for your LDAP server. Click **Help** for assistance.
- 5 Click **Apply**. The server is listed under LDAP Servers.

1.3 Install and configure components on z/OS.

To enable Automated Sign-On for Mainframe to IBM host applications, the MSS Administrative Server must exchange information with the Digital Certificate Access Server (DCAS) on z/OS (OS/390 V2R10 and later). DCAS works with RACF to obtain PassTickets, which act as time-limited single-use passwords in the automated sign-on process.

DCAS is included with z/OS Communications Server, but is not installed by default. You may wish to verify whether DCAS has already been enabled on the mainframe.

For example, if you have been using the Express Logon Facility (ELF) feature of z/OS, then DCAS may already be enabled; however, other z/OS components (such as the Telnet server or RACF) may need additional configuration.

Configuring DCAS and RACF

The z/OS administrator must configure DCAS so it can communicate with the MSS Administrative Server. In addition, certain configuration steps are necessary in RACF.

The administrator must also create a TLS key database file that contains both the DCAS client's certificate information and the DCAS server's certificate (public key) information. The MSS Administrative Server and DCAS must exchange public keys and place them in the other's trusted store.

The basic configuration steps include:

1. Configure RACF services for DCAS.
2. Configure DCAS and TLS on the z/OS mainframe.
3. Set up key exchange between the DCAS server and TLS.
4. Manage keys and certificates using RACF's Common key ring support.
5. Define a PassTicket profile for each application.
6. Configure the DCAS server.
7. Start the DCAS server.

For detailed steps, see [Appendix A: Configuring DCAS and RACF on z/OS](#).

4 Phase II: Configure a Simple Test Session

In Management and Security Server, create an IBM 3270 session and test a simple configuration for Automated Sign-On for Mainframe.

In brief, you will:

- ♦ Create a trust relationship (exchange of certificates) between the MSS Administrative Server and the DCAS server.
- ♦ Create an IBM 3270 session.
- ♦ Record and edit the logon macro.
- ♦ Assign user access for a simple test.
- ♦ Run a test.

2.1 Configure the Administrative Server for Automated Sign-On for Mainframe.

These settings are needed for testing, and can also be used in production.

A. Configure settings to manage certificates and passwords.

Use the MSS Administrative Console settings to manage the MSS Administrative Server certificate, the client certificate, and certificate signing requests.

- 1 In Administrative Console, click **Configure Settings > General Security**.
- 2 Scroll to **Smart card settings**. The default parameters specify the certificate attributes associated with the provider, SunPKCS11.
 - ♦ If you use SunPKCS11, you do not need to designate smart card libraries.
 - ♦ If you use a different provider, enter the smart card provider with the certificate attributes and designate the smart card libraries. For assistance, open **Help** and click the link for **Smart card settings**.
- 3 Consider entering passwords for the keystore and the keychain. See **Help** for details.
- 4 Accept or change the default settings.
- 5 Click **Apply**.

B. Enter the Automated Sign-On settings.

These settings require information from your IBM host administrator.

- 1 In the Administrative Console, click **Configure Settings > Automated Sign-On**.
- 2 Check **Enable automated sign-on to mainframe sessions**.

- 3 Enter the **DCAS server** name and port number.
- 4 If you choose to use a **DCAS keystore filename** and **password**, see **Help** for further instruction.
- 5 If you are using a secondary LDAP directory (described in Phase III Option B), click **Enable secondary LDAP server**.
 - 5a Enter the server-specific information for this LDAP server: Server type, Security options, Server name, Server port, Username, and Password.
 - 5b Enter details for the **Directory search base**. See **Help** for more information
- 6 Click **Apply**.

2.2 Create host sessions with the automated sign-on macro.

These settings are needed for testing, and can also be used in production.

In brief, you will

- ♦ Create an IBM 3270 session using the Administrative Console.
- ♦ Record and edit a logon macro in the host session.

For **Automated Sign-On for Mainframe**, the macro must

- ♦ Send a host application ID to the MSS Administrative Server so that the Administrative Server can request a PassTicket from DCAS.
- ♦ Insert the user's RACF credentials (PassTicket and mainframe user ID) that are returned from the MSS Administrative Server (to the client) into the data that is transmitted to the host. This action logs the user on to the mainframe application.

Follow the steps to create and configure an IBM 3270 session for either a Windows-based or web-based emulator.

- ♦ [IBM 3270 session for Windows-based emulators](#)
- ♦ [IBM 3270 Session for web-based emulators](#)

IBM 3270 session for Windows-based emulators

Create a session, and then record and edit a logon macro using Reflection/InfoConnect Desktop or Rumba. These steps use Reflection/InfoConnect Desktop as an example.

A. Create a IBM 3270 session for Reflection/InfoConnect Workspace

- 1 Open the Administrative Console to **Manage Sessions**, and click **+Add**.
- 2 Select **Reflection/InfoConnect Desktop** as the Product.
- 3 Select **Workspace** as the Session type.
- 4 Enter a **Session name**.
- 5 Accept the defaults, and click **Launch** to open the session.
- 6 Create a new **3270 terminal session**. In the **Create New Document** dialog, 3270 terminal should be selected. Click **Create**.
- 7 If the session will connect through the **Security Proxy Server**, continue with steps 8-11 to configure security.

- Otherwise, enter the name or IP address of the host computer, click **OK**, and proceed to step 12.
- 8 In the **Create New 3270 Terminal Document** dialog, check **Configure additional settings** (at the bottom of the dialog), and click **OK**.
 - 9 On the **Settings** dialog, under Host Connection, select **Set Up Connection Security** and click the **Security Settings** button.
 - 10 On the **SSL/TLS** tab in the Security Properties dialog, check both **Use SSL/TLS** security and **Use Security Proxy**. Configure the Security Proxy settings and Configure PKI, if necessary. Click **OK**. (The Security Proxy server name and port are listed on the Administrative Console > **Security Proxy** panel.)
 - 11 Accept the connection security settings and click **OK**. Continue to configure the features you want users to be able to access or edit. Click **Help** for guidance.
 - 12 Keep the session open and connected to the host to record and edit a macro (next set of steps).

B. Record and edit a macro in a Reflection Desktop session.

The logon macro is initiated when an authenticated user launches the session to connect or reconnect to the host.

NOTE: These instructions are guidelines to enable Automated Sign-On for Mainframe. Although error-checking is omitted for brevity and clarity, the macro author should check for errors as required by the application.

- 1 In the 3270 session you just created, start the macro recorder (Tools > Macros > **Record Macro**).
- 2 Connect to the host and log on to the appropriate host application using a valid user name and password.
You will edit the macro to remove specific user information and replace it with values that support logon by any authenticated user.
- 3 Stop the macro recorder (Tools > Macros > **Stop Recording**).
- 4 In the Recording Complete dialog, name the macro (for example TSO_logon). Click **OK**.
- 5 **Save** the macro with the current document (session) or in the common project.
By saving the macro with the current document, it will be transferred to the MSS Administrative Server when the session is saved in the Administrative Console, and then distributed to users who run this session from the list of session links.
- 6 Open the Visual Basic Editor (Macros > **Visual Basic**). Locate your macro: open Project > Modules, and double-click **Recorded** (or right-click > **View Code**).
- 7 After retrieving the `ibmCurrentTerminal` object, add the following line:

```
ibmCurrentTerminal.GetDASOPassTicket("APPID")
```


where "APPID" is replaced with the appropriate host application ID.
- 8 Edit the statement that sends your user name. Remove your user name and replace it with the mainframe username that was retrieved by the `GetDASOPassTicket` function call.
The edited line should look like this:

```
ibmCurrentScreen.SendKeys(ibmCurrentTerminal.DASOUserID)
```
- 9 a. Comment out or delete the line that uses the `PasswordBox` function to prompt the user for the password.

```

hiddenTextEntry = ibmCurrentTerminal.Macro.PasswordBox("", "")
If (hiddenTextEntry = "") Then
    Err.Raise 5002, "Hidden TextEntry", "No Value Provided.", "VBAHelp.chm",
    "5002"
End If

```

b. Replace it with a line that looks like this:

```

ibmCurrentScreen.SendKeys(ibmCurrentTerminal.DASOPassTicket)

```

- 10 **Save** the macro. Click **Yes** to send settings to the MSS Administrative Server.
Close the Visual Basic editor, and keep the session open.
- 11 Click the **Document Settings** button in the toolbar. On the Settings dialog, under Host Connections, select **Configure Advanced Connection Settings**.
- 12 In Configure Advanced Connection Settings, under Connection Action, check the boxes to
 - ♦ **Run a macro** or other action after the initial connection.
Select the logon macro and click **Ok**.
 - ♦ **Run when reconnecting**.
Select the logon macro and click **Ok**. The macro will be initiated when a user connects to a mainframe session.
- 13 **Save** the session. (Click **Save** or **Exit**.) Click **Yes** to send the settings to the MSS Administrative Server.

IBM 3270 Session for web-based emulators

Create a session, and then record and edit a logon macro using Reflection ZFE or Reflection for the Web. These steps use Reflection for the Web as an example

A. Create a Web-Based IBM 3270 session.

- 1 Open the Administrative Console to **Manage Sessions**, and click **+Add**.
- 2 Select **Reflection for the Web** as the Product.
- 3 Select **IBM 3270** as the Session type.
- 4 Enter a **Session name**.
- 5 Accept the defaults and click **Launch** to open the session.
- 6 In **Connection Setup**, enter the name or IP address of the host computer. Click **OK**.
- 7 Continue to configure the features you want users to be able to access or edit. Click **Help** for guidance.
- 8 Verify that the session connects to the host.
- 9 Click File > **Save and Exit**, and then **Save/Exit** to send the settings to the MSS Administrative Server.

B. Record and edit a logon macro in a Web-Based session.

The logon macro is initiated when an authenticated user launches the session to connect or reconnect to the host.

NOTE: These instructions are guidelines to enable Automated Sign-On for Mainframe. Although error-checking is omitted for brevity and clarity, the macro author should check for errors as required by the application.

- 1 Open the session you created from **Manage Sessions**.
- 2 Start the macro recorder.(Click Macro > **Start recording**.)
- 3 Connect to the host and log on to the appropriate host application using a valid user name and password.
You will edit the macro to replace specific user information with values that support logon by any authenticated user.
- 4 Click Macro > **Stop Recording...**
- 5 Enter the macro name, such as TSO Logon.
- 6 **Save** the macro and click **OK** to acknowledge the alert message, which tells you that the macro will be saved to the MSS Administrative Server only after you save and exit the session.
- 7 Click Macro > **Macros...** Select the macro you just created and click **Edit**.The macro opens for editing in a Macro Editor window.
- 8 Below the variable definitions at the top of the recorded macro, add the following line:

```
var credentials = eclcredentials.getDASOPassTicket( "APPID" );
```

where "APPID" is replaced with the appropriate host application ID.
- 9 In the macro's "performAction" function, edit the statement that sends your user name.
Remove your user name and replace it with the mainframe username that was retrieved by the `getDASOPassTicket` method. The edited line should look like this:

```
ps.SendDASOUserID( credentials );
```
- 10 In the macro's "performAction" function, comment out or delete the lines that use the `SendCredential` method to transmit the mainframe password. Replace it with a new line that transmits the `passticket` retrieved from the MSS Administrative Server.
The modified line should look like this:

```
ps.SendDASOPassTicket( credentials );
```
- 11 **Save** the macro and close the Macro Editor window.
- 12 To configure the macro to run on session startup, or to run on each connection, click Macros > **Macros...**
- 13 In the Macros dialog box, select "**Run at startup**" if you want the automated sign-on macro to run after the session launches.
Click the **Events** button, then assign the macro as the "On connect macro" if you want the automated sign-on macro to run every time the session connects.
- 14 Close the Macros dialog box, then **Save and Exit** the session to send the settings to the MSS Administrative Server.

2.3 Assign user access for testing

Map a setting to be used *only* for testing. For production, the assigned access is based on your data store of identity mappings, detailed in Phase III.

For testing, look up a sample of usernames by searching for a “literal value.”

- 1 In Administrative Console, click **Assign Access**.
- 2 Select a sample user or group by using **Search** in the left pane.
- 3 For the selected user or group, check the sessions that they are entitled to access. You should see the session that you created with the logon macro.
- 4 Next to the selected session, click **Edit**. This dialog asks for the method to use to derive the mainframe username for the selected user. For testing, choose **Literal value**.
Enter a literal value, such as an email address. Click **Help** for information about the other options.
- 5 Click **OK**.
- 6 To confirm your entries, click **Currently Assigned** on the left nav.

2.4 Run a test.

Using the smart card or credentials of one of the test users (assigned in section 2.3), log on to Management and Security Server (the MSS Administrative Server).

The expected result: The mainframe session opens without requiring a log on.

If the user is entitled to more than one session, a list of links appears. Clicking a link will automatically log the user on to that session.

5 Phase III: Set Up and Store Mainframe Username Mappings

Mainframe usernames can be stored in different ways. Determine which data store option fits your environment and whether you need to change the existing schema.

In brief, you will:

- ◆ Choose a data store option.
- ◆ Implement identity mappings and data storage.

3.1 Choose a data store option.

To decide, read through the conditions and scenarios described in each option:

- ◆ [Option A. An authenticating directory with primary user objects.](#)
- ◆ [Option B. An authenticating directory plus a secondary directory.](#)

Option A. An authenticating directory with primary user objects.

Conditions:

- ◆ Mainframe usernames are stored on the same LDAP directory that is used to authenticate your users.
- ◆ Every user has a single unique object.
- ◆ Each object has multiple attributes.
- ◆ An attribute is needed to search for mainframe usernames.

Implementation scenarios:

1. Add an attribute to an object.

Advantages:

- ◆ The LDAP schema is similar to a template.
- ◆ One user can have multiple mainframe usernames (attributes).

Disadvantage: Requires a change in schema.

2. Re-purpose an unused attribute.

Advantage: No change in schema is required.

Option B. An authenticating directory plus a secondary directory.

Conditions:

- ◆ An LDAP directory is used to authenticate users.
- ◆ Mainframe usernames are stored on a separate LDAP directory that is not used for authentication.

Implementation scenario:

1. **Set up a separate LDAP server and create a new set of objects – one per user – in the second directory.**

The LDAP search filter would: (1) Find the user's object with the attribute and

(2) Find the attribute within the object that has the mainframe username.

Advantages:

- ◆ The object is stable over time.
- ◆ Using **Assign Access** (in MSS), several options are available for searching the second LDAP directory and authorizing users to use automated sign-on:
 - ◆ Select UPN as the key to a secondary LDAP search filter.
 - ◆ Specify the LDAP attribute in the authenticating directory from which the UPN is obtained.
 - ◆ Select an LDAP attribute value in the authenticating directory as the key to a secondary LDAP search filter.
 - ◆ Select a literal value

Disadvantage:

- ◆ This scenario requires two LDAP directories.

3.2 Implement identity mappings and data storage.

The administrator must create a data store of identity mappings. The mapped data relates a user's enterprise identity (such as a smart card) to his or her mainframe username identity. Users may have more than one mainframe identity based on the applications they are entitled to access.

The text of the mappings must be provided in a format (such as CSV) that can be uploaded and searched. The administrator may choose to work with Consulting Services to prepare the identity mappings.

Configuration Tasks: Identity Mapping

- 1 Identify the data store option that you selected above, either
 - ◆ an authenticating directory with primary user objects -- or --
 - ◆ an authenticating directory plus a secondary directory
- 2 Gather the data for the identity mappings:
 - ◆ **Enterprise (authenticating) IDs**, recognized by the MSS Administrative Server.
 - ◆ **Mainframe Usernames** (RACF IDs), recognized by RACF.

For example, a user might have the following identities.

Enterprise ID	Mainframe Username (RACF ID)
CN=Joe User,OU=Users,DC=my-org,DC=com	TSOS2W3

NOTE: A user can have multiple mainframe usernames, based on their roles (such as end user or admin) and on the applications they are entitled to access.

- 3 Populate the data store with the mappings.

6 Phase IV: Configure Settings for Production

After the initial configuration has been tested, set user authorization by mapping access for all of your users and groups to their entitled sessions.

In brief, you will:

- ◆ Assign users' access to mainframe sessions.
- ◆ Deploy automated sign-on to users.

4.1 Assign users' access to mainframe sessions.

Assigning users' access to the sessions you created specifies the session URLs that are available to each user. You can assign access for individuals or groups.

- 1 In Management and Security Server's Administrative Console, click **Assign Access**.
- 2 Verify the Domain. Enter the name of a user or a group in the Search box. Click **Search**.
- 3 For the selected user or group, check the sessions in the right pane that they are entitled to access. You should see the sessions that you created with the logon macro.
- 4 Next to the selected session, click **Edit**. Choose the method you configured for determining the user's or group's mainframe username:
 - ◆ Derive from UPN
 - ◆ Get LDAP attribute value from authenticating directory
 - ◆ Get LDAP attribute value from secondary directory using search filter

For production, use a method *other* than "Literal value." Click **Help** for more information.

- 5 Click **Apply**.
- 6 To confirm your entries, click **Currently Assigned** on the left nav.

4.2 Deploy automated sign-on to users.

Provide the URL to the MSS Administrative Server for users to log on. Once authenticated, they will be automatically logged on to their mainframe session.

If a smart card is not available or does not work, the user will be prompted to enter their login credentials.

A Appendix A. Configuring DCAS and RACF on z/OS

Read through the overview and then follow the steps for configuring DCAS and RACF on z/OS.

NOTE: The procedures for using TopSecret and ACF2 are similar to those for RACF, but are not presented in this guide. For details, refer to the TopSecret or ACF2 documentation.

In Appendix A:

[Overview of DCAS Configuration and the z/OS Security Server](#)

1. [Configure RACF so that DCAS can run as a system daemon.](#)
2. [Configure TLS for use with DCAS.](#)
3. [Define a PassTicket profile for each application.](#)
4. [Update the Configuration for the DCAS server.](#)
5. [Start the DCAS server.](#)

[References](#)

Overview of DCAS Configuration and the z/OS Security Server

Automated Sign-On for Mainframe (ASM) works with IBM's Digital Certificate Access Server (DCAS), which is a component of the z/OS Communications Server. ASM requires that DCAS and the z/OS security server be configured to support PassTickets.

Security servers, such as RACF (Resource Access Control Facility), Top Secret, and ACF2, support PassTickets for use with z/OS. For simplicity, procedures are presented for configuring RACF; however, with minor modifications, the concepts and procedures also apply to Top Secret and ACF2.

To enable DCAS and RACF to support PassTicket services, the following conditions must be met.

- 1 RACF must be configured so that DCAS can run as a system daemon.
- 2 TLS must be configured for use with DCAS, including these items:
 - ♦ RACF key ring support must be enabled.
 - ♦ A TLS client authentication level must be configured.
 - ♦ A TLS server certificate for DCAS must be created or obtained.
 - ♦ A TLS client certificate must be created or obtained for use by the Automated Sign-On for Mainframe system to authenticate to DCAS.
- 3 A PassTicket profile must be defined for each host application that will support automated sign-on.
- 4 The DCAS server configuration must be updated with values that match those used with your deployment.
- 5 The DCAS server must be started.

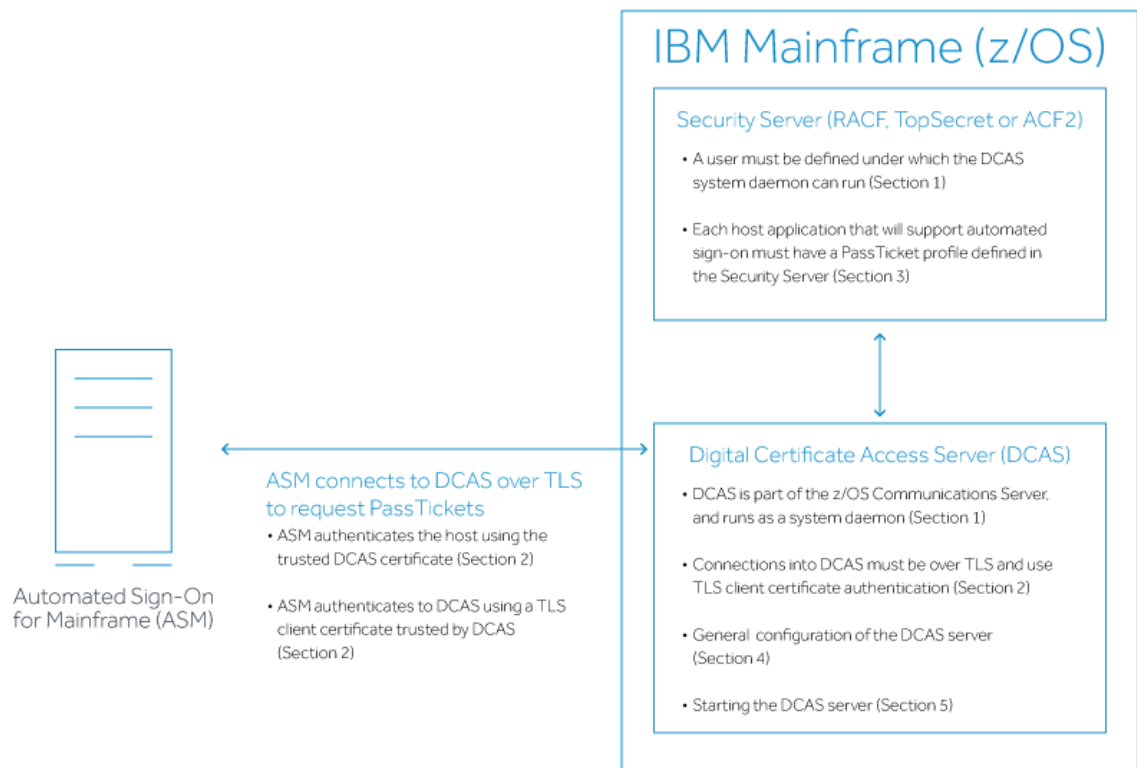
Detailed steps are provided in the sections that follow.

For more information, see these [References](#): *IBM Redbooks* and *Examples of Using CA ACF2 , CA Top Secret, or IBM RACF to Configure Passtickets*.

Information Exchange between ASM, DCAS, and RACF

In the Introduction of this Administrator Guide, an [overview diagram](#) depicts how the terminal client emulator, Administrative Server, and Automated Sign-On for Mainframe use PassTickets to provide automated log-on for the end user.

The following diagram shows further detail about how DCAS and the z/OS security server provide PassTicket services for use by Automated Sign-On for Mainframe. This diagram refers to the sections in Appendix A for configuring each item.



1. Configure RACF so DCAS can run as a system daemon.

To configure RACF so that DCAS can run as a system daemon, follow the steps and examples in this section.

In the sample RACF commands below, *italicized* items should be replaced with values appropriate for your environment.

NOTE: For information on RACF commands, refer to these [References](#): *OS/390 SecureWay® Security Server RACF Security Administrator's Guide* and *OS/390 SecureWay® Security Server RACF Command Language Reference*.

1.1 Define a user ID as superuser to use OMVS Services.

The DCAS server runs as a system daemon and must be started under a controlled user ID that has superuser authority (that is, not an end-user or system programmer user ID). To define the user ID to use OMVS services, use the following command:

```
ADDUSER dcasid DFLTGRP(OMVSGRP) OMVS(UID(0) HOME('/'))
```

where *dcasid* is the name of the user ID.

1.2 Provide a user ID with access to MVS.SERVMMGR.DCAS.

Starting DCAS from an MVS procedure requires that the user ID from which it is started have access to the MVS.SERVMMGR.DCAS resource in the OPERCMDS class. To provide this access, use the following commands:

```
RDEFINE OPERCMDS (MVS.SERVMMGR.DCAS) UACC(NONE)  
PERMIT MVS.SERVMMGR.DCAS CLASS(OPERCMDS) ACCESS(CONTROL) ID
```

1.3 Provide a RACF definition for MVS Start-up.

If DCAS is started as an MVS procedure, you will need the following RACF definition:

```
RDEFINE STARTED DCAS.* STDATA(USER(dcasid))  
SETR RACLIST(STARTED) REFRESH
```

where *dcasid* is the name of the user ID.

If CLIENTAUTH LOCAL2 is coded in the DCAS configuration file, at a minimum, you must use RACF to associate the certificate with a valid user ID. You can do this using the RACDCERT ADD command. The user ID could be the one associated with DCAS itself or it could be any valid user ID. If you want additional checking, you must activate the SERVAUTH class and define an EZA.DCAS.cvtsysname profile with the user ID associated with the client certificate to access the profile.

2. Configure TLS for use with DCAS.

This section provides overviews and detailed steps for configuring TLS for use with DCAS on z/OS.

2.1 Overview of Using System TLS with the DCAS Server

The DCAS server and the DCAS client use TLS to communicate. The TLS protocol uses a handshake in which the DCAS client and DCAS server authenticate each other, and they agree on how to encrypt/decrypt the data.

The cipher level used for encryption can be specified at the time DCAS is configured, using the V3CIPHER configuration keyword. The cipher level can also be set dynamically when DCAS starts, based on the level of cipher installed on the system. To set the cipher level dynamically, do not specify the V3CIPHER keyword.

TLS uses X.509 certificates and public/private keys (PKI). These keys are generated and stored in key databases, known as key rings.

The X.509 certificates can be created or obtained from a Certificate Authority. In either case, the certificate becomes part of a key ring. Various services are available for creating and managing key rings and certificates.

The RACDCERT command

The RACDCERT command in RACF can be used to create, register, store, and administer keys and certificates. If you use RACDCERT, you should specify the key ring to the DCAS server in the configuration file using the SAFKEYRING keyword. A key ring created this way does not have a password file associated with it.

For more information on digital certificates, refer to these [References](#): *z/OS SecureWay Security Server RACF Security Administrator's Guide* and *z/OS SecureWay Security Server RACF Command Language Reference*.

2.2 Configure a Client Authentication Level.

DCAS and RACF support several levels of authentication.

Authenticating the DCAS server: The DCAS client always authenticates the DCAS server. This requires that the DCAS client sent the client's z/OS application ID and user ID to the DCAS server.

Authenticating the DCAS client: The DCAS client uses a key pair and certificate to authenticate to the DCAS server. Different authentication level (levels of strictness) can be configured for the DCAS client's authentication to the DCAS server.

Authentication Levels

Choose client authentication Level 1, 2, or 3.

Level 1

With Level 1 authentication, the DCAS server verifies the client's identity using the TLS key database file. This file must contain the following certificates:

- ◆ the DCAS server certificate
- ◆ the DCAS client certificate

To use Level 1 authentication: Specify the CLIENTAUTH LOCAL1 keyword and value in the DCAS configuration file. The KEYRING or the SAFKEYRING keywords in the DCAS configuration file are used to specify the key ring used by the DCAS server.

Level 2

Level 2 includes Level 1 authentication plus an additional step -- verification that the DCAS client certificate has been associated in RACF with a valid user ID, which must be the user ID that DCAS is running under.

To configure DCAS for Level 2 authentication:

- 1 Specify the CLIENTAUTH LOCAL2 keyword and value in the DCAS server configuration file.
- 2 Use FTP (with the BINARY send option) to send the DCAS client's DER certificate to an MVS dataset.
- 3 Use the RACDCERT ADD command to add the certificate to RACF and associate it with a user ID, as shown in this example:

```
RACDCERT ID(dcasid) ADD('DCAS.HOSTPUB.CERT') TRUST
```

where *dcasid* is the name of the user ID.

Level 3

Level 3 includes Level 2 authentication plus verification that the DCAS client has been granted access in RACF to the DCAS server. The user ID derived from the certificate used with Level 2 RACF checks is defined as having access to the SERVAUTH RACF class and the EZA.DCAS.cvtsysname resource in the SERVAUTH class.

NOTE

- ◆ If the SERVAUTH class is not active or the EZA.DCAS.cvtsysname profile is not defined, or both, it is assumed this enhanced level is not requested.
- ◆ If the SERVAUTH class is active and the EZA.DCAS.cvtsysname profile is defined (but not for the user associated with the certificate) the requester's connection is terminated. For example:

```
RDEFINE SERVAUTH EZA.DCAS.cvtsysname UACC(NONE)  
PERMIT EZA.DCAS.cvtsysname CLASS(SERVAUTH) ACCESS(CONTROL) ID(dcasid)
```

where *dcasid* is the name of the user ID.

To configure DCAS for Level 3 authentication:

- 1 Specify the CLIENTAUTH LOCAL2 keyword and value in the DCAS configuration file.
- 2 Activate the SERVAUTH RACF class.
- 3 Define a profile for the EZA.DCAS.cvtsysname resource and associate the profile with the user ID associated with the certificate.

The ID associated with the certificate and the EZA.DCAS.cvtsysname can be any valid user ID.

2.3 Use RACF's Common key ring support to manage keys and certificates.

2.3 a. Set up initial RACF key ring support.

Before using RACF to store your key database information, ensure that:

- ◆ the digital certificate and digital key ring (DIGTCERT and DIGTRING) classes are active before defining certificates or key rings to RACF. For example:

```
SETROPTS CLASSACT(DIGTCERT DIGTRING)
```

- ◆ a refresh is performed after each update or change. For example:

```
SETROPTS RACLIST (DIGTRING DIGTCERT) REFRESH
```

- ◆ the RACDCERT command is defined as an authorized TSO command in the IKJTSoxx member.

To issue the RACDCERT command, you must have access to the FACILITY class IRR.DIGTCERT.function with UPDATE or CONTROL access.

If the DCAS server is started as an MVS started procedure, you must permit the RACF user ID to IRR.DIGTCERT.LIST.

If the DCAS server is started from a TSO user ID under the OS/390 UNIX shell, you must also permit that ID. For example:

```
DEFINE FACILITY (IRR.DIGTCERT.function)
UACC(NONE)
PERMIT IRR.DIGTCERT.LIST
CLASS(FACILITY) ID(dcasid)
ACCESS(control)
```

where *dcasid* is the name of the user ID.

2.3 b. Create a key ring.

Create a key ring for your DCAS server. For example:

```
RACDCERT ID(dcasid) ADDRING(SERVERKeyring)
```

where *dcasid* is the name of the user ID.

2.4. Create and Connect a Certificate

You can use RACF to create self-signed certificates (see section 3.4 a) or request a well-known certificate from a Certificate Authority and add it to RACF (see section 3.4 b).

For information on RACF commands, see *z/OS Security Server RACF Security Administrator's Guide* and *z/OS Security Server RACF Command Language Reference* in [References](#).

2.4 a. Create and connect self-signed certificates on the host.

Because the clients will not know about the issuer of the self-signed certificate, in most cases you must add the server's self-signed certificate to the client's signer certificates. This process requires the following high-level steps:

- ♦ Generate the DCAS server self-signed certificate on the host.
- ♦ Transfer the DCAS server's certificate to the DCAS client machine.

Following are detailed steps describing the process. DCAS server self-signed certificates can be created using RACF.

If using RACF, follow these steps:

- 1 Generate the DCAS server self-signed certificate on the host and transfer to the DCAS client.
 - 1a Create a self-signed server certificate using RACDCERT gencert:

```
RACDCERT ID(dcasid)
SUBJECTSDN(CN('DCASCERT'))
OU('TEST')
C('US'))
TRUST
SIZE(512)
WITHLABEL('DCASCERT')
```

where *dcasid* is the name of the user ID.

- 1b Use RACDCERT Connect to connect the certificate to a key ring and make it default. This example assumes a key ring called SERVERKeyring already has been created.

```
RACDCERT ID(dcasid)
CONNECT(ID(dcasid)
LABEL('DCASCERT')
RING(SERVERKeyring)
USAGE(PERSONAL) DEFAULT)
```

where *dcasid* is the name of the user ID.

- 1c Use RACDCERT EXPORT to export the DCAS server self-signed certificate in ".DER" format to an MVS file.

```
RACDCERT ID(dcasid) EXPORT(LABEL('DCASCERT'))
DSN('dcasid.SAFCERT.DER')
FORMAT(CERTDER)
```

where *dcasid* is the name of the user ID.

- 2 FTP the exported DCAS server certificate to the DCAS client using the FTP binary option.

2.4 b. Create and connect well-known certificates on the host.

Use the following steps to add a Certificate Authority Root and Personal Certificates to the Host.

- 1 Create a self-signed certificate and key pair for the DCAS server:

```
RACDCERT ID(dcasid)
GENCERT SUBJECTSDN(CN('labelname') C('us'))
WITHLABEL('labelname')
```

where *dcasid* is the name of the user ID.

- 2 Create a certificate request for a Certificate Authority (CA) by issuing RACDCERT GENREQ against the self-signed certificate:

```
RACDCERT ID(dcasid)
GENREQ(LABEL('labelname'))
DSN(labelname.certreqname)
```

where *dcasid* is the name of the user ID.

- 3 Send the certificate request to a Certificate Authority.
- 4 When you receive the DCAS server certificate from the Certificate Authority, transfer the file to the DCAS host.
- 5 If RACF does not already have the root certificate for the Certificate Authority, then you need to get it in .DER format, and add it to RACF using this command:

```
RACDCERT CERTAUTH ADD(caroot.der)
TRUST WITHLABEL('caroot')
```

- 6 Add the DCAS server certificate from the Certificate Authority back into RACF:

```
RACDCERT ID(dcasid) ADD(certname) WITHLABEL('certname')
```

where *dcasid* is the name of the user ID.

- 7 Connect the CA root certificate to the key ring with usage CERTAUTH:

```
RACDCERT ID(dcasid)
CONNECT(CERTAUTH LABEL('caroot'))
RING(SERVERKeyring)
USAGE(CERTAUTH) DEFAULT)
```

where *dcasid* is the name of the user ID.

- 8 Connect the DCAS server certificate to the key ring with usage PERSONAL:

```
RACDCERT ID(dcasid)
CONNECT(ID(dcasid) LABEL('certname')
RING(SERVERKeyring)
USAGE(PERSONAL) DEFAULT)
```

where *dcasid* is the name of the user ID.

3. Define a PassTicket profile for each application.

A RACF PTKTDATA (PassTicket data class profile) must be created for each application ID that will support PassTickets. This profile enables the DCAS server to obtain a PassTicket for the application and user ID, and to pass it back to the client that requested the PassTicket from DCAS. This profile name must match the RACF PTKTDATA application name that is configured on the host. This name could be the same as the application name that the user is logging onto (for example, the name on USSMSG10).

When creating PTKTDATA profiles for applications such as TSO, the application name portion of the profile will most likely not be the same. For example, RACF requires that the application ID portion of the profile name be TSO+SID. Refer to *z/OS Security Server RACF Security Administrator's Guide* (in [References](#)) to determine the correct profile naming.

You must create these profiles on each separate RACF system (the system where the users will be logging on to) that contains target applications for Automated Sign-On for Mainframe. The PTKTDATA class profile defined in the "target" RACF system must match the PTKTDATA class profile in the system where the PassTicket is created, which is the system where the DCAS server executes. These PTKTDATA class profiles need to have corresponding profile names and identical secret keys (defined using the KEYMASKED parameter).

Here is an example of a PassTicket data class profile for the application TSORUS (the KEYMASKED value is a hexadecimal string of your choice):

```
RDEFINE PTKTDATA TSORUS
SSIGNON(KEYMASKED(A1A2A3A4A5A6A7A8)
UACCESS(NONE) )
SETR RACLIST(PTKTDATA) REFRESH
```

The APPLID name must be correct. For example, for TSO, the profile is TSO+SID. The SID is the SMF system id that is defined in the SMFPRMxx member in SYS1.PARMLIB. For more information on defining PassTicket profiles, refer to the *z/OS Security Server RACF Security Administrator's Guide* (see [References](#)).

4. Update the Configuration for the DCAS Server

Additional configuration is needed to update the DCAS configuration file and DCAS start procedure. The values must match those used with your deployment.

The DCAS configuration file (*/etc/dcas.conf*) contains the following keywords:

TCPIP <i>tcpstackname</i>	; Server will have affinity to TCP/IP stackname
IPADDR <i>xx.xx.xx.xx</i>	; IP address to which DCAS binds for TLS connection (defaults to <i>inaddr_any</i>)
PORT <i>xxxx</i>	; DCAS listens on this port number (default is 8990)

KEYRING /etc/ssl/xxx.kdb	; HFS file name of Keyring for TLS/SSL negotiation
STASHFILE /etc/TLS/xx.sth	; Stash file containing the Password of Key ring file
SAFKEYRING SERVERKeyring	; Key ring via RACF
CLIENTAUTH xxxxxxx	; Client Authentication level, used with parameters: ; LOCAL1 (TLS does authentication) ; LOCAL2 (default - use RACF to validate the client's certificate)
LDAPSERVER xx.xx.xx.xx	; Fully qualified name or IP address of LDAP Server
LDAPPORT xxxx	; Port# that LDAP Server is listening on
V3CIPHER cipherspec	; Specify a subset of the supported TLS V3 cipher algorithms. The following cipher levels are valid: ; 01=NULL MD5 ; 02=NULL SHA ; 03=RC4 MD5 Export ; 04=RC4 MD5 US ; 05=RC4 SHA US ; 06=RC2 MD5 Export ; 09=DES SHA ; 0A=Triple DES SHA US

5. Start the DCAS server.

The DCAS server can be started as a generic server without stack affinity or as a server with affinity to a specific TCP/IP stack. The DCAS server can be started in different ways (detailed steps follow):

- ♦ automatically when the TCP/IP address space is started.
- ♦ from the z/OS UNIX shell.
- ♦ from an MVS started procedure.

To start the DCAS server automatically when the TCP/IP address space is started, specify DCAS on the AUTOLOG statement in the TCPIP profile dataset. For example:

```
AUTOLOG
DCAS
ENDAUTOLOG
```

The following sample procedure can be used to start DCAS. First, enter the command S DCAS. To pass optional parameters to DCAS, specify them after the final slash (/) on the PARM statement, for example:

```
// PARM=('POSIX(ON) ALL31(ON)')
// 'ENVAR("LIBPATH=/usr/lib")/-d 3 -l SYSLOGD')
```

Sample procedure:

```
//DCAS PROC
/** DEBUGGING AND LOGGING MAY BE REQUIRED TO HELP DETERMINE A PROBLEM
/** THE DCAS.
/**
/** -D OR -D - INDICATES DEBUGGING LEVEL REQUESTED.
/** FORMAT: -D LEVEL
/** LEVEL IS: 1=LOG ERROR AND WARNING MESSAGES
/** 2=LOG ERROR, WARNING, AND INFO
/** 3=LOG ERROR,WARNING, INFORMATI
/**
/**<br>//DCAS EXECPGM=EZADCDMN,REGION=4096K,TIME=NOLIMIT,
/** PARM='POSIX(ON) ALL31(ON) / -d 3 -l SYSLOGD'
/**
//SYDENV DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSERR DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
/**
```

You will find a sample start procedure in EZADCASP in the SEZAINST dataset.

To start the DCAS server from the z/OS UNIX shell, use the following format:

```
dcas <parameter_1> <parameter_2> <parameter_3> &
```

To start the DCAS server from an MVS started procedure, use the following format:

```
PARM=.../<parameter_1> <parameter_2> <parameter_3>
```

Optional Parameters

The following optional parameters can be used from either the z/OS UNIX shell or the MVS started procedure:

-d or -D

Indicates debugging. The following levels apply:

1= Specifies log error and warning messages.

2= Specifies log error, warning, and informational messages.

3= Specifies log error, warning, informational, and debug messages. This is the default.

-l or -L

Indicates logging to SYSLOGD or to a designated log file. If you do not specify this parameter, logging defaults to /tmp/dcas.log. If you specify a debug level, but not logging, then the DCAS server attempts to open the default log file /tmp/dcas.log. If this fails, debugging is turned off. For SYSLOGD, the DCAS server uses the log facility local0.

-c or -C

Indicates the requested configuration file (for example, /u/userx/passtick.conf). If you do not specify this parameter, the DCAS server looks for the configuration file using the following search order:
DCAS_CONFIG_FILE environment variable /etc/dcas.conf tsouserid.DCAS.CONF
TCPIP.DCAS.CONF

If the DCAS server does not find a valid configuration file, it will not start.

When DCAS is started, the process ID (pid) is stored in a Hierarchical File System (HFS) file. The file name under which it is stored depends upon how you configure DCAS:

- ♦ **If the DCAS server is configured with TCP/IP stack affinity**, the pid file is named /tmp/dcas.tcpipname.pid where tcpipname is the name of the TCP/IP stack for which DCAS has affinity.
- ♦ **If the DCAS server is configured without stack affinity**, the pid file is named /tmp/dcas.INET.pid.

You can stop the DCAS server from the UNIX shell or from MVS:

- ♦ **To stop the DCAS server from the UNIX shell**, use the following command: `kill -s SIGTERM pid`
- ♦ **To stop the DCAS server from MVS**, use the following command: `P DCAS`

References

IBM Redbooks (<http://www.redbooks.ibm.com/>)

OS/390 SecureWay® Security Server RACF Command Language Reference (<http://publibz.boulder.ibm.com/epubs/pdf/ich1a421.pdf>)

OS/390 SecureWay® Security Server RACF Security Administrator's Guide (<http://publibz.boulder.ibm.com/epubs/pdf/ich1a721.pdf>)

CA Docs: Examples of Using CA ACF2, CA Top Secret, or IBM RACF to Configure PassTickets (https://support.ca.com/cadocs/0/CA%20Cross-Enterprise%20Application%20Performance%20Management%20Release%209%206-ENU/Bookshelf_Files/HTML/ConfigurePassTicket/index.htm?toc.htm?2225069.html?intcmp=searchresultclick&resultnum=293)

