
Automated Sign-on for Mainframe - Administrator Guide

14.1.1

Table of contents

Automated Sign-On for Mainframe	4
Before You Begin	5
Before You Begin	5
How Automated Sign-on for Mainframe Works	5
System Requirements and Prerequisites	7
Terms	8
Configuration Workflow	9
Initial Setup	10
Initial Setup	10
1. Install or Upgrade MSS	10
2. Activate Automated Sign-On for Mainframe	11
3. Configure DCAS and RACF on z/OS	12
4. Configure Authentication and Authorization	13
5. Establish Trust between the MSS Administrative Server and the DCAS Server	15
6. Enable Your Emulator for Automated Sign-On	18
7. Create an IBM 3270 Session with an Automated Sign-On	30
Simple Test	31
Simple Test	31
8. Assign Access to One User for Testing	31
9. Run a Test	32
Production	33
Production	33
10. Map Enterprise IDs to Mainframe User Names	33
11. Assign Access to the Automated Sign-on for Mainframe Sessions	36
12. Deploy Automated Sign-on Sessions to Users	37
Administrators' Task List	38
MSS administrator	38
Terminal emulation administrator	39

z/OS administrator	39
Appendix A	40
Appendix A: Configuring DCAS and RACF	40
Overview of DCAS Configuration and the z/OS Security Server	40
Configure RACF so DCAS Can Run as a System Daemon	42
Configure TLS for Use with DCAS	44
Define a PassTicket Profile for Each Application	51
Update the Configuration for the DCAS Server	52
Start the DCAS Server	53

Automated Sign-On for Mainframe

Automated Sign-On for Mainframe enables users to automatically — and securely — log on to host applications on a z/OS mainframe by using a terminal emulation client. Automated Sign-On eliminates the need for eight-character passwords.

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

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

- **Management and Security Server (MSS)** — to secure connections, create and assign automated sign-on sessions
- **the terminal emulation client** — to create a logon macro and configure the client
- **z/OS** — to support PassTickets

Before You Begin

Before You Begin

Become acquainted with how Automated Sign-on works and the setup requirements.

[How Automated Sign-on for Mainframe Works](#)

[System Requirements and Prerequisites](#)

[Terms](#)

Note the [Configuration Workflow](#), and then use this guide to perform the tasks.

Set up the initial configuration.

Run a simple test.

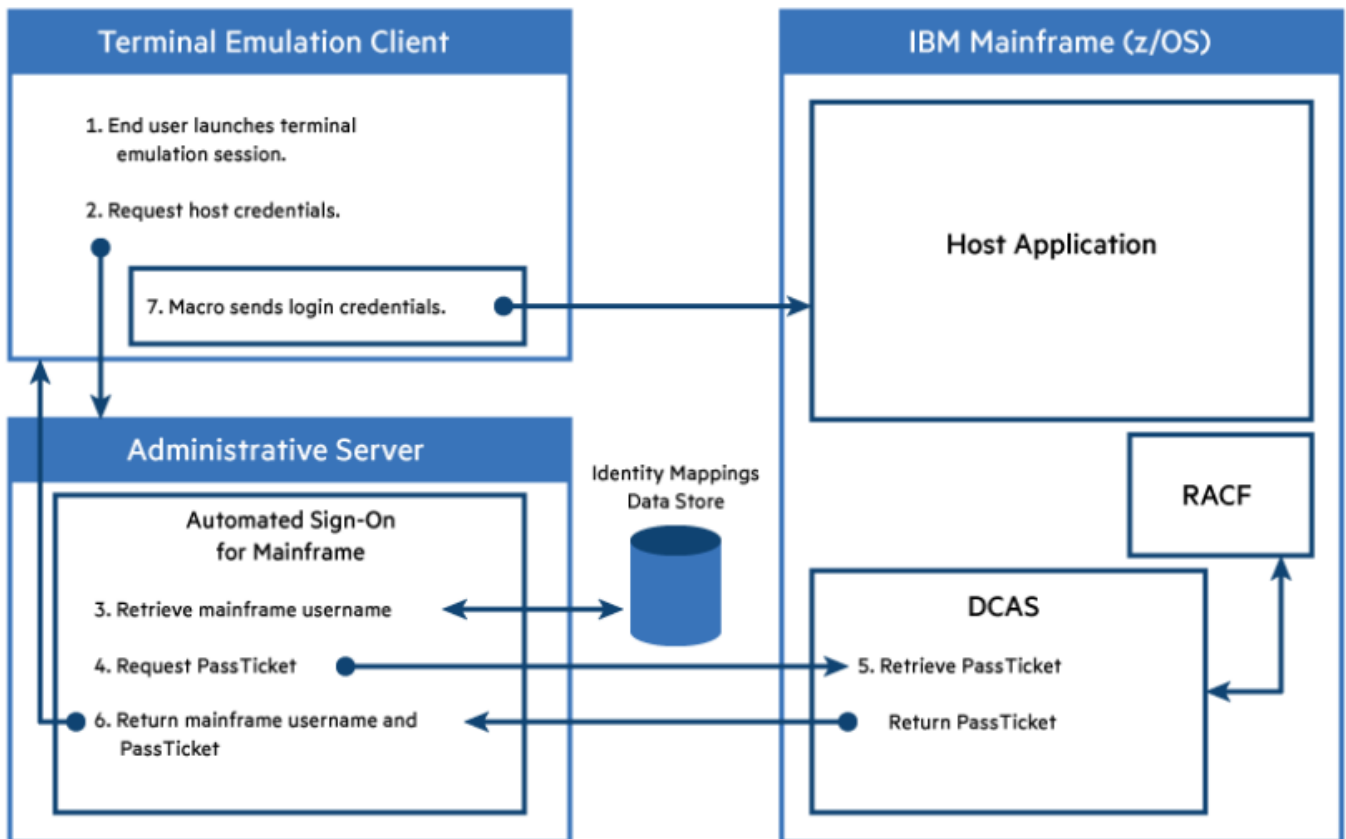
Prepare for production.

See the [Administrators' Task List](#) to distribute the configuration tasks among administrators.

How Automated Sign-on for Mainframe 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 or InfoConnect Desktop, Host Access for the Cloud, or Rumba+), 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 user name from the data store of mapped mainframe user names.
4. The MSS Administrative Server passes the host application ID and the end user's mainframe user name 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 user name and the PassTicket to the emulation client.
7. The terminal emulation client's login macro sends the user's mainframe user name and PassTicket to the host application. The user is automatically logged on.

System Requirements and Prerequisites

Before installing or configuring Automated Sign-On for Mainframe, the following products and systems must be in place.

System Requirements and Prerequisites for Automated Sign-on for Mainframe

Requirement	Comment
MSS version 12.5 or higher	installed on the designated server See the MSS Deployment Guide for System Requirements
Automated Sign-On for Mainframe Add-On product	activation file installed on the MSS server
LDAP directory	for user authorization
Micro Focus terminal emulation software: - Reflection or InfoConnect Desktop 16 or higher Workspace Automated Sign-on sessions require Reflection or InfoConnect Desktop version 16.2 or higher - Host Access for the Cloud 2.4 or higher - Reflection for the Web 12.3 SP1 or higher The version must be compatible with MSS. See the Reflection for the Web Release Notes . - Rumba+ Desktop 9.4.1 or higher	on the client and administrator workstations Note: The emulator client must have the API functionality that enables Automated Sign-On for Mainframe.
z/OS with DCAS installed	See Appendix A. Configuring DCAS and RACF on z/OS

Requirement	Comment
TLS connection (default is TLSv1.3, TLSv1.2)	from the MSS Administrative Server to DCAS

Terms

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

- **Administrative Console:** the user interface for the MSS Administrative Server, used to manage and configure terminal sessions.
- **Administrative Server (or MSS Administrative Server):** component installed with Host Access Management and Security Server.
- **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.
- **Workspace Automated Sign-on:** a specific session type that can be used with Reflection or InfoConnect Desktop 16.2 or higher.

Configuration Workflow

Automated Sign-on for Mainframe requires configuration to be done in different places, most likely by different people. Some tasks can be done in parallel.

Follow the flow of tasks and note what needs to be configured where. After the Initial Setup is configured, you can run a simple test to prepare for production.

Workflow	Configuration task
Initial setup	<ol style="list-style-type: none"> 1. Install or upgrade MSS on a server that meets the system requirements 2. In MSS: Activate Automated Sign-On for Mainframe 3. On z/OS: Configure DCAS and RACF 4. In MSS: Configure authentication and authorization 5. In MSS: Establish trust between the MSS Administrative Server and the DCAS server 6. In your emulator client: Enable the emulator for automated sign-on. The steps are specific to your emulator and session type. 7. In MSS: Create an IBM 3270 session with an automated sign-on macro
Simple test	<ol style="list-style-type: none"> 8. In MSS: Assign access to one user for testing 9. In MSS: Run a test
Production	<ol style="list-style-type: none"> 10. In the data store: Map enterprise IDs to mainframe user names 11. In MSS: Assign access to the automated sign-on for mainframe sessions 12. Using your typical deployment method, deploy automated sign-on sessions to users

Initial Setup

Initial Setup

The Initial Setup steps ensure that the required products and components are installed. Some basic configuration needs to be done before you configure Automated Sign-on for Mainframe.

The installation requirements are summarized in [System Requirements and Prerequisites](#).

In brief, the administrators will

[Install or upgrade Management and Security Server](#)

[Activate the Automated Sign-On for Mainframe Add-On](#)

[Configure DCAS and RACF on z/OS](#)

[Configure Authentication & Authorization](#)

[Establish trust between the MSS Administrative Server and the DCAS server](#)

[Enable your emulator for automated sign-on](#)

[Create an IBM 3270 session with an automated sign-on macro](#)

See the [Configuration Workflow](#) for an overview of the tasks and where they need to be done.

Use the [Administrators' Task List](#) to distribute the configuration tasks to the appropriate administrator.

1. Install or Upgrade MSS

Install (or upgrade) Host Access Management and Security Server (MSS) on a server that meets the system requirements.

Refer to the [MSS Deployment Guide](#) for details.

2. Activate Automated Sign-On for Mainframe

Automated Sign-On for Mainframe Add-On is provided as an activation file, which must be uploaded and activated in Management and Security Server.

Note

The activation file for your emulator also needs to be uploaded.

1. Check to see which activation files are already installed.
 - a. In the MSS Administrative Console, click About > Activated Products.
 - b. In the list of Currently Installed activation files, look for the Automated Sign-On for Mainframe Add-On your terminal emulator client
 - c. For each activation file listed, check the version. To ensure compatibility with the latest features, be sure the `<major>.<minor>` version of each activation file is the same as the version of MSS.
2. If the activation files for Automated Sign-On for Mainframe and your terminal emulation product are present for version 14.1.1, skip to [Configure DCAS and RACF](#).
If not, you must obtain and activate version 14.1.1. Continue with step 3.
3. Download and install the 14.1.1 activation files for Automated Sign-On for Mainframe Add-On and/or the terminal emulator client.
 - a. On the Downloads site, from the list of product entitlements, locate Host Access Management and Security Server--Automated Sign-On for Mainframe Add-On. Click Download and click the automated sign-on activation file: `activation.automated_signon_for_mainframe-<version>.jaw`
 - b. Accept the terms and license to download both MSS and the activation file for your terminal emulator, such as `activation.mss_for_windows_desktop_emulation-<version>.jaw`. Note the download location.
4. In the MSS Administrative Console, open Configure Settings - Product Activation.
 - a. Click Activate New to open a list of available files and browse to the location where the activation files were downloaded.
 - b. Click the activation file for automated sign-on:
`activation.automated_signon_for_mainframe-<version>.jaw`
The list of Currently Installed products now includes Automated Sign-On for Mainframe Add-On version 14.1.1.
5. Repeat the Activate New steps to activate your terminal emulator client.

6. Restart your browser to ensure that the Administrative Console is fully updated with the new set of activation files. You do not need to restart the MSS Server (service).

Once Automated Sign-On for Mainframe is activated and enabled, it's time to [configure DCAS and RACF](#) on your z/OS system.

3. Configure DCAS and RACF on z/OS

This configuration is required before trust can be established between MSS and the DCAS server.

To enable Automated Sign-on for Mainframe to connect 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 the 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 used 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.

Configure DCAS to communicate with MSS

The z/OS administrator must configure DCAS (and RACF) to communicate with the MSS Administrative Server.

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.

Detailed steps are presented in [Appendix A. Configuring DCAS and RACF on z/OS](#).

In brief, the z/OS administrator will:

- Configure RACF services for DCAS.
- Configure DCAS and TLS on the z/OS mainframe.
- Set up key exchange between the DCAS server and TLS.
- Manage keys and certificates using RACF's Common key ring support.
- Define a PassTicket profile for each application.

- Configure the DCAS server.
- Start the DCAS server.

Note

If you use more than one DCAS server, you can configure each of them for Automated Sign-on. When you assign access to an automated sign-on session, you can choose which DCAS server to use.

When the z/OS setup is complete, return to the MSS Administrative Console to continue configuration.

4. Configure Authentication and Authorization

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

Note

An LDAP directory is required for user authorization.

To configure user authentication and authorization

1. In the MSS Administrative Console, click Configure Settings - Authentication & Authorization.
2. Select an Authentication method (any other than None).

Using smart cards. If users will authenticate to the MSS Administrative Server with smart cards, select X.509. Further configuration may be required before running a simple test, noted in When smart cards are used for authentication.

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

Choose Authentication Method

Authentication method

- None
- LDAP
- Single sign-on through IIS
- Windows Authentication - Kerberos (See help to enable)
- Windows Authentication - NTLMv2 (Not recommended)
- X.509
- SiteMinder (See help to enable)
- Micro Focus Advanced Authentication
- SAML

Choose Authorization Method

Authorization method

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

4. Scroll to LDAP Servers and click +ADD your LDAP server.
5. Enter the required information for your LDAP server. Click Help for assistance.
6. Click Apply. The server is listed under LDAP Servers.

After you configure Authentication and Authorization, proceed to Configure Settings - Automated Sign-on and continue to establish trust between the MSS Administrative Server and the DCAS server.

5. Establish Trust between the MSS Administrative Server and the DCAS Server

This step requires information about the DCAS server and is dependent on configuring DCAS and RACF on z/OS.

Each DCAS server must be configured to accept client connections from the MSS Administrative Server.

Several keystores must be correctly configured for client authentication. (For details, see [Configuring DCAS and RACF](#).)

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

Configure Settings - Automated Sign-On

Before you begin, obtain this information for each DCAS server (from your z/OS host administrator):

DCAS server name

DCAS server port

Note

When [smart cards are used for authentication](#), configure those settings first, and then continue with these steps to configure Automated Sign-On.

See the MSS Help for more information about each setting.

1. In the Administrative Console, click Configure Settings - Automated Sign-on.
2. Check Enable Automated Sign-On for Mainframe (for z/OS systems).
3. Click +ADD and enter the the name and port of the DCAS server.

The default port is 8990; however, the DCAS server can be configured to use any port.

4. Choose which certificate to use for client authentication of the MSS Administrative Server to the DCAS server.

- **Use Management and Security Server certificate.** This option uses the Administrative Server's certificate and private key (configured on the Configure Settings - Certificates panel).
- **Use custom keystore.** This option uses a separate keystore that contains a certificate and private key. Follow these steps:
 - a. Enter the Keystore filename with the correct extension. The keystore can be one of these formats:

Java keystore: `.jks`

PKCS#12 keystore: `.p12` or `.pfx`

Bouncy Castle BCFKS keystore: `.bcfks`

b. Enter the (case-sensitive) Keystore password used to read the keystore. The password for the keystore and the private key must be the same.

c. Click **Upload** to upload the custom keystore to the Management and Security Server.

5. Check **Verify server identity** to verify the hostname entered in the Server name field against the certificate received from the DCAS server when a secure connection is made from the MSS Administrative Server to DCAS.
6. Click TEST CONNECTION to test the connection between the MSS Administrative Server and the DCAS server. Then click OK to return to Configure Settings - Automated Sign-on.

Using a secondary LDAP directory to store mainframe user names

1. If you are using a secondary LDAP directory to use in the Automated Sign-On workflow (Option B in Choose a data store option), check *Enable secondary LDAP server*.
 - Enter the server-specific information for this LDAP server: Server type, Security options, Server name, Server port, User name, and Password.
 - Enter details for the Directory search base. See Help for more information.
 - When TLS/SSL is selected, you need to import the LDAP server's trusted certificate into the default trusted keystore. Click IMPORT CERTIFICATE.
 - TEST CONNECTION verifies the connection between the secondary LDAP server and the MSS Administrative Server. If the connection fails, consult the logs to resolve the issue.
2. Under User Principal Name (UPN), enter the name of the LDAP attribute in the authenticating directory that contains the UPN value.

This value is needed when assigning automated sign-on sessions that derive the mainframe user names from the UPN.
3. If using a secondary LDAP server, enter information for the Search filter. See Help for more information.

Remember your selection. When you Assign Access, you are prompted to select the Method to obtain mainframe user name. Choose from these options:

 - **Not set**. This default is not a viable option for automated sign-on. Choose another method.

- **Derive from UPN.** Select this option to request a passticket from DCAS by deriving the mainframe username from the User Principal Name (UPN) of the user. The UPN is typically available from a smart card or client certificate, and is a standard attribute in Active Directory servers. A UPN is formatted as an Internet-style email address, such as userid@domain.com, and Management and Security Server derives the mainframe username as the short name preceding the '@' symbol.
- **Get LDAP attribute value from authenticating directory.** Select this option to perform a lookup in the LDAP directory (defined in Authentication & Authorization) and return the value of the entered attribute as the mainframe username. All LDAP attributes must meet these criteria:
 - must begin with an alpha character
 - no more than 50 characters
 - any alphanumeric character or a hyphen is permitted
- **Get LDAP attribute value from secondary directory using search filter.** Select this option to use the search filter to find the user object in the secondary LDAP directory; then return the value of the entered attribute as the mainframe username.
- **Literal value.** This option is available for sessions assigned to users, but not groups. Enter a value that meets these criteria:
 - up to eight alphanumeric characters
 - no spaces
 - no other characters

4. Click Apply.

The Initial Setup requirements are met for MSS.

5. Next step: [Enable your emulator for automated sign-on](#)

When smart cards are used for authentication

Configure these 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. Accept or change the default settings.
 4. Click Apply.
 5. Continue with [Configure Settings - Automated Sign-on](#).

6. Enable Your Emulator for Automated Sign-On

Click your emulator (and session type), and then follow the steps for that specific setup.

[Reflection or InfoConnect Desktop Workspace Automated Sign-on](#)

[Reflection or InfoConnect Desktop Managed Sessions](#)

[Host Access for the Cloud](#)

[Reflection for the Web](#)

[Rumba+ Desktop](#)

Reflection or InfoConnect Desktop Workspace Automated Sign-on

This session type in Reflection or InfoConnect Desktop enables the administrator to implement automated sign-on for users who create and save mainframe sessions on their desktops.

The Workspace Automated Sign-on session type requires Reflection or InfoConnect Desktop version 16.2 or higher.

[Enable Reflection or InfoConnect Desktop to use Workspace Automated Sign-on](#)

[Create an IBM 3270 session for Workspace Automated Sign-on](#)

Enable Reflection or InfoConnect Desktop to use Workspace Automated Sign-on

The administrator must:

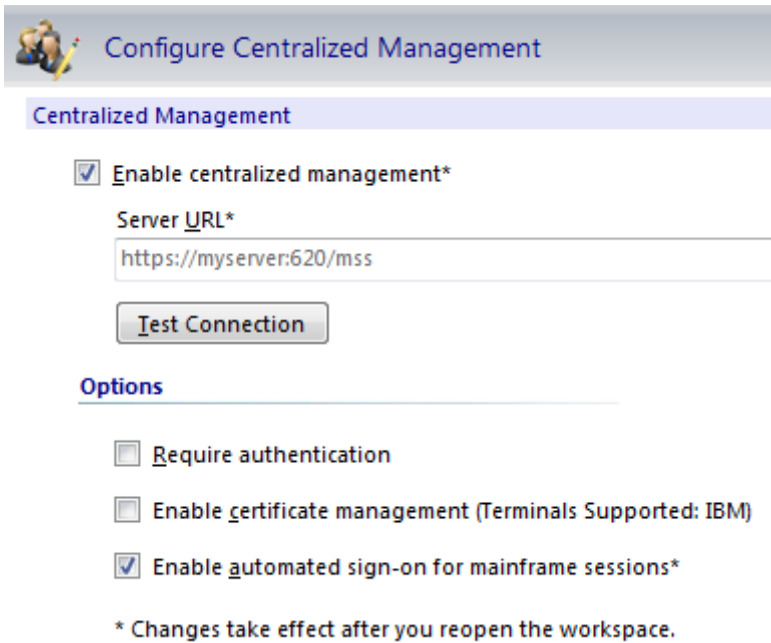
[Configure centralized management](#)

[Create an automated sign-on macro and save the session](#)

Configure centralized management

This global setting establishes a connection between the client and the MSS Administrative Server, which is needed to request and deliver the PassTicket for automated sign-on.

1. In Reflection or InfoConnect Desktop, open the Workspace Settings.
2. Click Configure Centralized Management.
3. Check Enable Centralized Management.
4. Enter the URL for your MSS Administrative Server (Management and Security Server). Click OK.



Configure Centralized Management

Centralized Management

Enable centralized management*

Server URL*

https://myserver:620/mss

Test Connection

Options

Require authentication

Enable certificate management (Terminals Supported: IBM)

Enable automated sign-on for mainframe sessions*

* Changes take effect after you reopen the workspace.

5. Check Enable automated sign-on for mainframe sessions.

This setting is needed to use Automated Sign-on for Mainframe when sessions are created by users and saved on their individual desktops. When enabled, the automated sign-on macro inserts a time-limited PassTicket to log the user on to the mainframe session.

Create an automated sign-on macro and save the session

Create a macro that uses methods and properties on the IbmTerminal object.

To automatically log on a user to a mainframe session, 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 these steps, heeding the required naming conventions.

1. For the macro, gather the application ID and valid logon credentials for the mainframe application.
2. In Reflection or InfoConnect Desktop, create a session and configure it to connect to the host that users will automatically log on to.
3. From the Macros tab, click Record VBA.
4. Log on to the mainframe host application with valid credentials, and then click Stop Recording.
5. In the Recording Complete dialog, save the macro in the current document's project. Name the macro according to these requirements:
 - To apply the macro to all sessions connecting to this mainframe, name it *SignOn*.
 - To apply this macro only to sessions connecting to a specific port on this mainframe, you must append the name with underscore `[_]<port number>`.

For example: *SignOn_623*.

Note

If you want to specify different logon information for different ports on the same host, you can create a macro for each port and save these macros in the same VBA module.

For example, if you connect to a mainframe on port 623 and on port 723 with a different logon, create different macros named *SignOn_623* and *SignOn_723*. When the session connects, the port-specific macro is used.

6. To ensure the session VBA Project component has the required name, save the session document file with the required name: `ASM.rd3x`.
7. In the VBA Editor, open the ASM Project > Modules. Open the Recorded module.
8. Edit the macro code to use PassTickets:
 - Add this line after the variable declarations:


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

 where `APPID` is replaced with the host application ID (noted in step 1).
 - Replace your user name with the `DASOUserID` retrieved by the `GetDASOPassTicket` function:


```
ibmCurrentScreen.SendKeys(ibmCurrentTerminal.DASOUserID)
```

 sending your `DASOUserID` instead of your user name.
 - 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

ibmCurrentScreen.SendKeys(hiddenTextEntry)

```

- Replace that line with one that contains the DASOPassTicket that was retrieved by the GetDASOPassTicket function call, like this:

```
ibmCurrentScreen.SendKeys(ibmCurrentTerminal.DASOPassTicket)
```

9. Save the macro. Close the VBA editor and save the session document.

Hint

To add another macro for a specific port on this mainframe, disconnect this session and connect on that port. Then repeat the steps in this procedure to record another SignOn macro and save it with the port number appended to the SignOn name (for example, SignOn_623).

Create an IBM 3270 session for Workspace Automated Sign-on

To create an IBM 3270 session for Workspace Automated Sign-on, the MSS administrator must [create a workspace sign-on session in MSS](#) and then [upload the session document](#) that contains the automated sign-on macro.

Create a workspace sign-on session in MSS

1. Open the Administrative Console to Manage Sessions, and click +Add.
2. Select Reflection/InfoConnect Desktop as the Product.
3. Select Workspace Automated Sign-on as the Session type.
4. Enter a Session name that exactly matches the name of the host to which the session document files are configured to connect.

For example, if the host name is myHost, then the Session name must be myHost.

If your environment has session documents that are configured to connect to variations of host names (such as fully qualified names or IP Addresses), create a separate Workspace Automated Sign-on session for each name. For examples, see the Manage Sessions Help.

Upload the session document containing the automated sign-on macro

1. Click Browse. Select the Reflection or InfoConnect Workspace session document file (ASM.rd3x) that contains the automated sign-on for mainframe login macro.

The Reflection or InfoConnect Desktop administrator created this session document during Initial Setup.

2. Click Save to upload the settings file and save the session.

The session is added to the Manage Sessions list and is available to be assigned.

Reflection or InfoConnect Desktop Managed Sessions

This session type in Reflection or InfoConnect Desktop uses Management and Security Server to create mainframe sessions and save them on the MSS Administrative Server, where they can be centrally updated and maintained. Managed sessions can be deployed via the Assigned Sessions list.

Automated Sign-on for Mainframe is available with Reflection or InfoConnect Desktop version 16.0 or higher.

[Enable Reflection or InfoConnect Desktop for automated sign-on](#)

[Create an IBM 3270 Workspace session and add an automated sign-on macro](#)

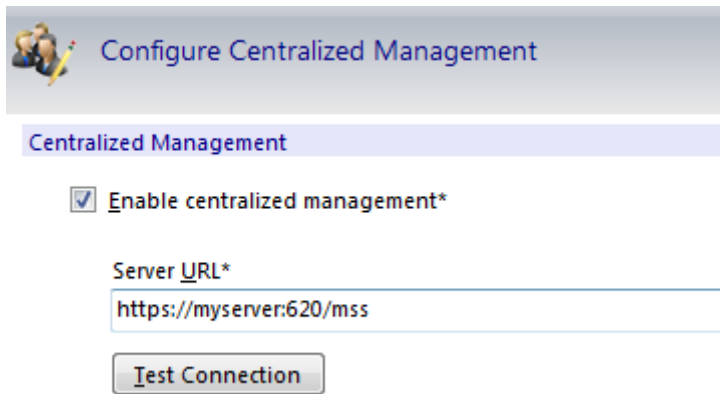
Enable Reflection or InfoConnect Desktop for automated sign-on

In brief, the administrator must enable centralized management in Reflection or InfoConnect Desktop.

Enable centralized management

This global setting establishes a connection between the client and the MSS Administrative Server, which is needed to request and deliver the PassTicket for automated sign-on.

1. In Reflection or InfoConnect Desktop, open Reflection Workspace Settings.
2. Click Configure Centralized Management.
3. Check Enable Centralized Management.
4. Enter the URL for your MSS Administrative Server (Management and Security Server). Click OK.



Configure Centralized Management

Centralized Management

Enable centralized management*

Server URL*

https://myserver:620/mss

Test Connection

Create an IBM 3270 Workspace session and add an automated sign-on macro

In brief, the administrator must

[Create an IBM 3270 Workspace session in MSS](#)

[Record and edit a macro in a Reflection Workspace session](#)

Create an IBM 3270 Workspace session in MSS

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. 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. 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. Continue to record and edit a macro in a Reflection Workspace session.

Record and edit a macro in a Reflection Workspace session

The logon macro is initiated when an authenticated user launches the session to connect or reconnect to the host. Keep these notes in mind when creating the macro.

Note

The automated sign-on 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.
- 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. These settings are needed for testing, and can also be used in production.

1. In the 3270 session you just created, start the macro recorder (Macros > Record VBA).
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 (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.
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 this 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 user name that was retrieved by the `GetDASOPassTicket` function call.

The edited line should look like this: `ibmCurrentScreen.SendKeys(ibmCurrentTerminal.DASOUserID)`

9. 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

ibmCurrentScreen.SendKeys(hiddenTextEntry)
```

Replace that line with one 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. Open Document Settings (File > Settings > Document Settings). Under Host Connection, click 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.

Host Access for the Cloud

Host Access for the Cloud uses MSS for centralized management, so no additional setting is required.

However, be sure the Automated Sign-on for Mainframe activation file is installed on MSS. To verify or install the activation file, [Activate the Automated Sign-On for Mainframe Add-On](#).

Briefly, to enable HACloud for automated sign-on:

- [Create a Host Access for the Cloud session in MSS](#)
- Record and edit a macro in a Host Access for the Cloud

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.

- The logon macro is initiated when an authenticated user launches the session to connect or reconnect to the host.
- In the IBM 3270 session you just created, create a macro to log on to this mainframe session. See [Creating Macros in Host Access for the Cloud User Guide](#).
- Name the macro, for example ASO_logon.
- Edit the macro to contain the AutoSignon object that provides the methods needed to create a Host Access for the Cloud login to use with Automated Sign-on.

See the example in Automatic Sign-On Macro for Mainframe in the [Host Access for the Cloud User Guide](#).
- Save the macro and send the settings to the MSS Administrative Server.
- Save the session and send the settings to the MSS Administrative Server.

Reflection for the Web

Reflection for the Web uses MSS for centralized management, so this requirement is met.

However, be sure these prerequisites are met:

- **Compatible versions.** The version of Reflection for the Web must be compatible with Management and Security Server. See the [Reflection for the Web Release Notes](#).
- **Activation file.** The Automated Sign-On for Mainframe activation file must be installed on MSS.

To verify or install the activation file, [Activate the Automated Sign-On for Mainframe Add-On](#).

Create a Reflection for the Web 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 or edit the default settings, 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. Keep the session open and continue with 7B. Record and edit a macro in a Reflection for the Web session.

Record and edit a macro in a Reflection for the Web session

The logon macro is initiated when an authenticated user launches the session to connect or reconnect to the host. Keep these notes in mind when creating the macro

Note

The automated sign-on 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.

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

1. In the open IBM 3270 session you created, start the macro recorder (Macro > Start recording).
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 replace specific user information with values that support logon by any authenticated user.

3. Click Macro > Stop Recording...
4. Enter the macro name, such as TSO Logon.
5. 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.
6. Click Macro > Macros. Select the macro you just created and click Edit. The macro opens for editing in a Macro Editor window.

7. 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.

8. 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 );
```

9. 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 );
```

10. Save the macro and close the Macro Editor window.
11. To configure the macro to run on session startup, or to run on each connection, click Macros > Macros...
12. 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.
13. Close the Macros dialog box, and then Save and Exit the session to send the settings to the MSS Administrative Server.

Rumba Desktop

Automated Sign-on for Mainframe is available with Rumba+ Desktop. Automated sign-on configuration is also documented in the Rumba+ Desktop System Administrator Guide. Search for “MSS Automated Sign-On” for the prerequisites and configuration steps.

Enable Rumba+ Desktop for automated sign-on

In brief, the Rumba administrator must:

- Ensure the prerequisites are met
 - Management and Security Server is installed.
 - The Automated Sign-On for Mainframe Add-On activation file is installed.
 - Centralized Management is enabled in Rumba Options on each user’s local machine.
- Create an automated sign-on on connection macro

See “Creating a connection macro” in the [Rumba+ System Administrator Guide](#).

Note

The automated sign-on 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.

- Create a session profile that contains the macro

See “Creating a session profile” in the [Rumba+ System Administrator Guide](#). Note where the session profile is saved.

Create a Rumba Desktop session with the automated sign-on macro

The MSS administrator must:

- Create a Rumba+ session in MSS

Open the Administrative Console to Manage Sessions, and click +Add.

Select Rumba+ Desktop as the Product.

Enter a Session name.

- Upload a Rumba+ Session Profile
 - On the Add New Session panel, click Browse. Locate and select the Rumba+ session profile (created in step 6C) that contains the automated sign-on connection macro.
 - Click Save to upload the profile and save the session.

The session is added to the Manage Sessions list and is available to be assigned.

7. Create an IBM 3270 Session with an Automated Sign-On

After the emulator is enabled (step 6), the administrator can create an IBM 3270 session in MSS that includes an automated sign-on macro. The steps differ depending on your emulator and session type.

Follow the steps for your emulator type:

- [Reflection or InfoConnect Desktop - Workspace Automated Sign-on](#)
- [Reflection or InfoConnect Desktop - Managed Sessions](#)
- [Host Access for the Cloud](#)
- [Reflection for the Web](#)
- [Rumba+ Desktop](#)

Simple Test

Simple Test

When the Initial Setup is complete, you are ready to run a simple test.

In brief, the MSS administrator will:

- [Assign access to one user for testing](#)
- [Run a test](#)

8. Assign Access to One User for Testing

To test the macro, choose one user who has a “literal value” mapped to a mainframe user name.

1. In Administrative Console, click Assign Access.
2. For testing, Search for the user with the mapped user name. (You can use * in the Search box.)
3. For the selected user, check the session that you created with the logon macro.
4. Next to the selected session, click Edit.

In the Source of user name dialog, identify the method to use to derive the mainframe user name to automatically log on that user to that session.

- For testing, choose Literal value, which is available only for individual users, not groups.
(Literal value is not used in production.)
 - Enter a Literal value, such as the user’s mainframe user name, that conforms to these criteria:
 - up to eight alphanumeric characters
 - no spaces
 - no other characters
 - If you configured more than one DCAS server, select or verify the one to use for this assignment.
 - Click OK.
5. Click Currently Assigned (on the left navigation) to confirm your entries.
 6. [Run a test](#).

For production, access is assigned based on your data store of identity mappings. (Map enterprise IDs to mainframe user names.)

9. Run a Test

Run a test for your emulator type.

For Reflection/ InfoConnect Desktop - Workspace Automated Sign-on

Create a session document that connects to the host name that you configured in the Automated Sign-on session.

Expected result The test is successful when the user is signed on automatically to the IBM 3270 session. That is, the automated sign-on macro retrieved a PassTicket and logged on the user with their enterprise ID,

For other emulator types

Log on to the MSS Administrative Server using one of the test user's credentials (or smart card). Or, if presented with a list of assigned sessions, click the session that has the automated sign-on for mainframe macro.

Note

If the user is assigned to more than one automated sign-on session, a list of assigned sessions is presented. Each link will automatically log the user on to that session.

Expected result The IBM 3270 session opens without needing to log on to the mainframe. That is, the automated sign-on macro retrieved a PassTicket and logged on the user with their enterprise ID.

Production

Production

When the Simple Test is successful for one user, you are ready to prepare for deployment.

To deploy the automated sign-on session to multiple users, the users' enterprise IDs must be mapped to their mainframe user names. Then, user authorization can be set by assigning access for all of your users and groups to their entitled sessions.

In brief, the MSS administrator will:

- [Map enterprise IDs to mainframe user names](#)

- [Assign access to the automated sign-on for mainframe sessions](#)

- [Deploy automated sign-on sessions to user](#)

10. Map Enterprise IDs to Mainframe User Names

Your users' enterprise user names (used for authenticating) must be mapped to their mainframe user names (used for authorization) so that the automated sign-on macro can log on directly to the mainframe application.

Mainframe user names 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, the administrator will:

- [Choose a data store option](#)

- [Implement identity mappings and data storage](#)

Choose a data store option

There are two options:

[An authenticating directory with primary user objects](#)

[An authenticating directory plus a secondary directory](#)

To help you decide, read through the conditions and scenarios described for each option.

An authenticating directory with primary user objects

Conditions

- Mainframe user names 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 user names.

Implementation scenarios

1. Add an attribute to an object.

Advantages:

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

Disadvantage: Requires a change in schema.

2. Re-purpose an unused attribute.

Advantage: No change in schema is required.

An authenticating directory plus a secondary directory

Conditions

An LDAP directory is used to authenticate users.

Mainframe user names are stored on a separate LDAP directory that is not used for authentication.

Implementation scenario

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 user name.

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.

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 user name 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 User names (RACF IDs), recognized by RACF.

For example, a user might have the following identities.

Enterprise ID - `CN=Joe User,OU=Users,DC=my-org,DC=com`

Mainframe User name (RACF ID) - `TS0S2W3`

Note

A user can have multiple mainframe user names, 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.

When the identity mappings are in place, continue with [assigning access](#) to the automated sign-on for mainframe sessions.

11. Assign Access to the Automated Sign-on for Mainframe Sessions

Once the identity mappings are stored, you can authorize users to access the automated sign-on sessions. Access can be assigned to individuals or groups.

1. In the MSS Administrative Console, click Assign Access.
2. Verify the Domain. Enter the name of a user or a group (or enter *) in the Search box. Click Search.
3. Select a user or group, and check the automated sign-on sessions that they are entitled to access.
4. Next to the selected session, click Edit. Select the source of the mainframe user name.

Note

If you are using a secondary LDAP server, select the method you set for the Search filter.

Derive from UPN

Get LDAP attribute value from authenticating directory

Get LDAP attribute value from secondary directory using search filter

Do not use Literal value for production. See Help for more information.

5. If you configured more than one DCAS server, select or verify the one to use for this assignment.
6. Click OK.

7. To confirm your entries, click **Currently Assigned** in the navigation panel.

Now, you are ready for the final step, deploying automated sign-on sessions to users.

12. Deploy Automated Sign-on Sessions to Users

After the authorized users are assigned access to the automated sign-on sessions, use your typical method to deploy sessions for production.

For instance, you might provide the URL for the automated sign-on session to the authorized users. Once authenticated to the MSS Administrative Server, they will be automatically logged on to their mainframe session.

Product documentation for deployment-related tasks

Check your product documentation for deployment information specific to your product.

- [Reflection Desktop Deployment Guide](#)
- [InfoConnect Desktop Installation and Deployment Guide](#)
- [Host Access for the Cloud User Guide](#)
- [Rumba+ Desktop System Administrator Guide](#)

Administrators' Task List

The configuration of Automated Sign-on for Mainframe typically requires assistance from more than one administrator.

This section takes the vantage point of each system administrator and the tasks that need to be done. Of course, one person may serve the role of more than one administrator.

The tasks are presented as checklists to ensure the Initial Setup is complete before you run a Simple Test and then deploy to Production.

MSS administrator

Perform these tasks in Management and Security Server, noting the dependencies. Refer to the [Configuration Workflow](#) for the full picture of what needs to be configured.

Note

The Initial Setup requires a task to be done by the z/OS administrator.

Configuration steps	Task
Initial setup	<ol style="list-style-type: none"> 1. Install or upgrade MSS 2. Activate the Automated Sign-On for Mainframe Add-On 3. After the z/OS administrator has configured DCAS and RACF on z/OS, configure authentication and authorization. 4. Establish trust between the MSS Administrator Server and the DCAS server 5. Create an IBM 3270 session with an automated sign-on macro
Simple test	<ol style="list-style-type: none"> 1. Assign access to one user for testing 2. Run a test

Configuration steps	Task
Production	<ol style="list-style-type: none">1. Map enterprise IDs to mainframe user names2. Assign access to the automated sign-on for mainframe sessions3. Deploy automated sign-on sessions to users

Terminal emulation administrator

For each terminal emulation product, complete these tasks:

Enable the product to use automated sign-on.

For Reflection/InfoConnect Desktop, configure centralized management and then create an automated sign-on macro.

z/OS administrator

Configure DCAS and RACF on z/OS. See the detailed steps in [Appendix A](#).

Appendix A

Appendix A: Configuring DCAS and RACF

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

Note

These instructions are intended ONLY for quickly configuring DCAS and the z/OS Security Server on a demo or test system using System SSL. We strongly suggest using AT-TLS on a production system.

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.

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

[Configure RACF so that DCAS can run as a system daemon](#)

[Configure TLS for use with DCAS](#)

[Define a PassTicket profile for each application](#)

[Update the Configuration for the DCAS server](#)

[Start the DCAS server](#)

Overview of DCAS Configuration and the z/OS Security Server

Automated Sign-on for Mainframe works with DCAS, a component of the z/OS Communications Server. Automated Sign-on 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:

RACF must be configured so that DCAS can run as a system daemon.

- 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.

- A PassTicket profile must be defined for each host application that will support automated sign-on.

The DCAS server configuration must be updated with values that match those used with your deployment.

The DCAS server must be started.

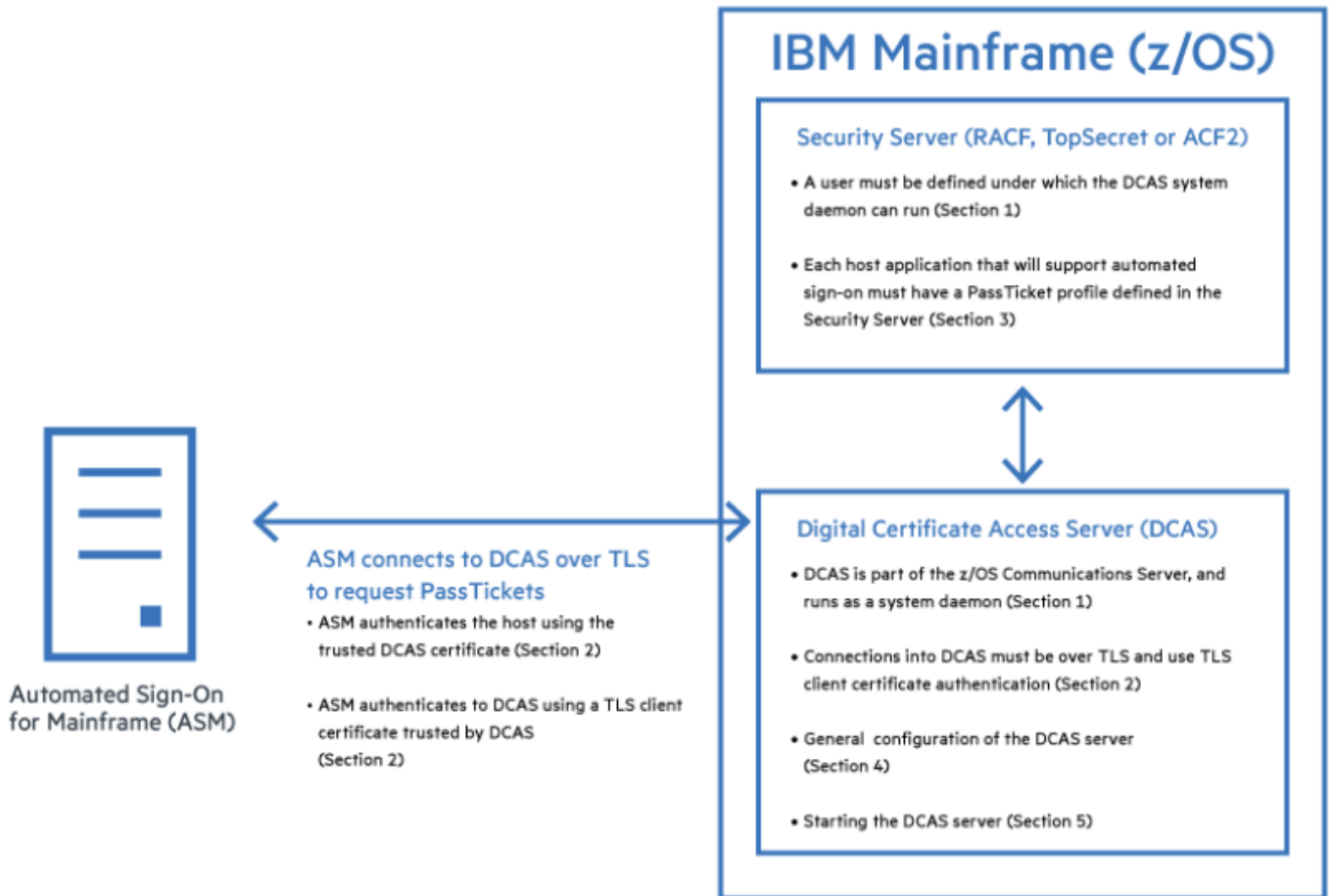
Detailed steps are provided in the sections that follow.

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 Automated Sign-on, 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.



Configure RACF so DCAS Can Run as a System Daemon

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](#).

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.

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
```

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.cvtssystemname profile with the user ID associated with the client certificate to access the profile.

More information

[IBM Redbooks](#)

Configure TLS for Use with DCAS

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

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.

Note

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

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 both the DCAS server and client certificates.

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.

Use RACF's common key ring support to manage keys and certificates

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.

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.

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).

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](#).

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

Note

In these examples, `dcasid` is the name of the user ID.

1. Generate the DCAS server self-signed certificate on the host and transfer to the DCAS client.

- Create a self-signed certificate using RACDCERT GENCERT:


```
RACDCERT GENCERT ID(dcasid)
SUBJECTSDN(CN('DCASCERT')
OU('TEST')
C('US'))
TRUST
SIZE(2048)
WITHLABEL('DCASCERT')
```

- 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)
```

- 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)
```

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

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.

Note

In these examples, `dcasid` is the name of the user ID.

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

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

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)
```

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')
```

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)
```

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)
```

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).

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 tcpstackname ;Server will have affinity to TCP/IP stackname

IPADDR xx.xx.xx.xx ;IP address to which DCAS binds for TLS connection (defaults to inaddr_any)

PORT xxxx ;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 xxxxxx ;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

LDAPPORNT 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

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